Poster: Efficient Hyperparameter Optimization for Differentially Private Deep Learning

Aman Priyanshu
Manipal Institute of Technology
aman.priyanshu@learner.manipal.edu

Rakshit Naidu
Carnegie Mellon University
rnemakal@andrew.cmu.edu

Fatemehsadat Mireshghallah
University of California, San Diego
fmireshg@eng.ucsd.edu

Mohammad Malekzadeh
Imperial College London
m.malekzadeh@imperial.ac.uk

***Abstract*—Tuning the *hyperparameters* in the differentially private stochastic gradient descent (DPSGD) is a fundamental challenge. Unlike the typical SGD, private datasets cannot be used many times for hyperparameter search in DPSGD, such as Grid Search. Therefore, there is an essential need for algorithms that, within a given search space, can find near-optimal hyperparameters for the best achievable privacy-utility tradeoffs efficiently. We formulate this problem into a general optimization framework for establishing a desirable privacy-utility tradeoff, and systematically study three cost-effective algorithms for being used in the proposed framework: *evolutionary*, *Bayesian*, and *reinforcement learning*. Our experiments, for hyperparameter tuning in DPSGD conducted on MNIST and CIFAR-10 datasets, show that these three algorithms significantly outperform the widely used *grid search* baseline. As this abstract offers a first-of-a-kind framework for hyperparameter tuning in DPSGD, we discuss existing challenges and open directions for future studies. As we believe our work has implications to be utilized in the pipeline of private deep learning, we open-source our code at https://github.com/AmanPriyanshu/DP-HyperparamTuning.**

***Index Terms*—Differential Privacy, Privacy Preserving Machine Learning, Hyperparameter Optimization, Bayesian Optimization, Reinforcement Learning**

I. Introduction

Deep neural networks (DNNs) [1] can learn very useful patterns from large multi-dimensional datasets, enabling motivational applications; eg: in health [2]. However, large amounts of training data are required for not only learning the near-optimal DNN parameters for the underlying task, but also for finding the right set of hyperparameters that enable appropriate learning. For a task defined on public datasets, the same data can be reused as many times as we wish. But, as every reuse of the available data comes at a price of some privacy loss, *hyperparameter tuning* has been a fundamental challenge for tasks defined on private datasets.

Differential Privacy (DP) [3] provides strong guarantees for the individuals participating in private datasets. DP restricts the maximum contribution of each sample on the result of a computation on the private dataset. Differentially-private stochastic gradient descent (DPSGD) [4] is a widely accepted algorithm for training DNNs on private datasets, where zero-mean Gaussian noise, with a predefined variance, is added to the clipped gradients computed for each sample in the training dataset at each iteration. Noisy gradients often result in a degraded accuracy for the trained DNN.

Previous works look at two variants: (1) optimizing privacy parameters of a private model for achieving comparable performance to a non-private model and (2) providing privacy guarantees to reach moderate performance [5]. However, in practice, both hyperparameters and privacy parameters need to be optimized within the user-specified privacy budget. Thus, in this abstract, we propose a systematic study for learning hyperparameters faster (constrained by a privacy budget) and with less privacy cost through four different optimization algorithms.

II. Methodology

Although there is a wide range of hyperparameters that one can choose from in DPSGD (eg: noise multiplier, clipping factor, batch size, learning rate, etc.), in this abstract, we specifically focus on two important hyperparameters: *noise multiplier* σ (the standard deviation of the Gaussian noise) and *learning rate* η. We optimize for these two parameters specifically as the epsilon (ϵ i.e. privacy leakage) and validation loss (minimizing the validation loss) are highly dependent on the chosen values for σ and η, respectively. To this end, we study three cost-effective algorithms: *evolutionary*, *Bayesian*, and *reinforcement learning* and compare the results with the *grid search* baseline.

We consider the problem of training a DNN with a fixed architecture (i.e., the number, type, and size of each layer) using DPSGD. Let D_{train} denotes the training set and D_{valid} denote the validation set. Let $H = \{h_1, \ldots, h_N\}$ denotes the set of N *hyperparameters* that are used during training on D_{train} and have impact on both *validation loss* (val_loss) and *privacy loss* in DP (ϵ), on D_{valid}. To provide a general but customizable framework, we define *reward* as a weighted linear combination of val_loss and ϵ:

$$reward = (\alpha_U \cdot e^{-val\_loss}) + (\alpha_P \cdot e^{-\epsilon}) \quad (1)$$

We use regularizers α_U and $\alpha_P \in [0, 1]$ to control the importance of utility and privacy, respectively (to control the

TABLE I
COMPARISON OF DIFFERENT METHODS BASED ON THE BEST-ACHIEVED REWARD AND THE AVERAGE TIME REQUIRED TO ATTAIN THIS REWARD FOR THE SEARCH SPACE ON (A) CIFAR-10 AND (B) MNIST.

Method	Time (in hours)	Best Reward (in %)	Accuracy (in %)	Epsilon (ϵ)
(A) CIFAR-10				
Grid Search	150.020	51.406	**44.936**	0.600
Evolutionary	**11.064**	52.044	37.999	0.599
Bayesian	49.636	51.846	43.864	**0.581**
Reinforcement	52.971	**52.398**	44.884	0.590
(B) MNIST				
Grid Search	43.712	72.260	**89.133**	0.683
Evolutionary	5.250	72.615	73.745	**0.175**
Bayesian	**2.853**	73.385	81.562	0.349
Reinforcement	31.165	**74.906**	75.022	0.240

privacy-utility trade-off). In our proposed framework, we first set these α regularizers, and then start searching for the optimal hyperparameters in H using the algorithms explained in the following section. In this abstract, we consider $H = \{\sigma, \eta\}$, where σ denotes the noise multiplier and η denotes the learning rate in DPSGD. Our aim for the following experiments remains to optimize the reward denoted by Equation (1). Notice that, in practice, the value of α_U and α_P depends on the requirements of the underlying task.

A. Optimization Techniques

- *Grid Search Method*: Grid search is utilized to provide a sufficient exploration within a restricted search space.
- *Evolutionary Optimization*: Evolutionary optimization algorithms provide an opportunity to utilize adaptive search optimization algorithms [6].
- *Bayesian Optimization*: Bayesian optimization combines prior experience with sample information to approximate the function distribution using the Bayesian formula [8].
- *Reinforcement Learning*: In our application of this method, we use a regression network capable of estimating the reward output of training on a particular set of hyperparameters. Subsequently, in the following episodes, we select a certain percent of experiments based on the reward estimate of the regression network. This methodology allows us to estimate the hyperparameter-reward function and verify the proximal search space of high-performing hyperparameters, giving us generalized results.

III. RESULTS

To assess and analyze the effectiveness of optimization algorithms across both CIFAR-10 and MNIST datasets, we use Grid-Search on a similar search complexity as the other methods. We display the computational time taken, best reward achieved, and its respective accuracy and epsilon value in Table I. Grid search displays a poor understanding of the epsilon-accuracy as it is not adaptive in nature. It achieves a reward of 72.2% and 51.4% on the MNIST and CIFAR-10 datasets, respectively.

Here, we observe that although Reinforcement Learning provides the highest performance, it comes at the expense of computational time. On the other hand, Evolutionary Algorithms and Bayesian Optimization provide consistent results with respect to computational time and performance.

IV. CONCLUSION & FUTURE WORK

In this abstract, we discussed different methodologies for hyperparameter tuning for the private training of deep neural networks using DPSGD algorithm. We proposed a novel, customizable reward function that allows users to define a single objective function for establishing their desired privacy-utility tradeoff. We quantified, compared, and analyzed the methods of grid search (as the baseline), Bayesian optimization, evolutionary optimization, and reinforcement learning, across two datasets, CIFAR-10, and MNIST. We observed that Bayesian and evolutionary optimization behave similarly in terms of the privacy-utility trade-off point they provide, and how efficiently they find it. Reinforcement learning, however, provides a more desirable trade-off but with varying efficiencies across datasets. All three methods perform much better than the baseline grid search algorithm. We believe that our work serves as a valuable resource for privacy-preserving ML practitioners, developers, and researchers for hyperparameter tuning.

For future work, one can use our proposed method alongside that of [7], where a portion of the privacy budget is allocated to finding the appropriate learning rate on the private dataset. Another direction is to extend our proposed method to tune other hyperparameters in DPSGD, and even the network architecture and non-linear activation functions that are used.

REFERENCES

[1] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

[2] Ziller, A., Usynin, D., Remerscheid, N., Knolle, M., Makowski, M., Braren, R., ... Kaissis, G. (2021). Differentially private federated deep learning for multi-site medical image segmentation. arXiv preprint arXiv:2107. 02586.

[3] Dwork, C., McSherry, F., Nissim, K., & Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. Theory of cryptography conference, 265–284. Springer.

[4] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, 308–318.

[5] van der Veen, K. L., Seggers, R., Bloem, P., & Patrini, G. (2018). Three tools for practical differential privacy. arXiv preprint arXiv:1812. 02890.

[6] Bochinski, E., Senst, T., & Sikora, T. (2017). Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms. 2017 IEEE International Conference on Image Processing (ICIP), 3924–3928. doi:10.1109/ICIP.2017.8297018 BIBTEXWU201926 Wu, J., Chen, X.-Y., Zhang, H., Xiong, L.-D., Lei, H., & Deng, S.-H. (2019). Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimizationb. Journal of Electronic Science and Technology, 17(1), 26–40. doi:10.11989/JEST.1674-862X.80904120

[7] Chen, C., & Lee, J. (2020). Stochastic adaptive line search for differentially private optimization. 2020 IEEE International Conference on Big Data (Big Data), 1011–1020. IEEE.

[8] Wu, J., Chen, X.-Y., Zhang, H., Xiong, L.-D., Lei, H., & Deng, S.-H. (2019). Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimizationb. Journal of Electronic Science and Technology, 17(1), 26–40. doi:10.11989/JEST.1674-862X.80904120