Architecture Matters: Investigating the Influence of Differential Privacy on Neural Network Design

Felix Morsbach
Karlsruhe Institute of Technology
Karlsruhe, Germany
felix.morsbach@kit.edu

Tobias Dehling
Karlsruhe Institute of Technology
Karlsruhe, Germany
dehling@kit.edu

Ali Sunyaev
Karlsruhe Institute of Technology
Karlsruhe, Germany
sunyaev@kit.edu

Abstract

We explore the relationship between neural network architectures and model accuracy under differential privacy constraints. Our findings show that architectures that perform well without differential privacy, do not necessarily do so with differential privacy. This shows that extant knowledge on neural network architecture design cannot be seamlessly translated into the differential privacy context. Moreover, as neural architecture search consumes privacy budget, future research is required to better understand the relationship between neural network architectures and model accuracy to enable better architecture design choices under differential privacy constraints.

1 Introduction

Differential privacy has become the de facto standard for achieving data confidentiality in machine learning settings. The most prominent way to train differentially private neural networks is by clipping gradients in order to limit the impact of each data point and by adding noise to the gradient updates [2]. While gradient clipping is also used in non-private training to avoid overfitting [18], adding noise inherently reduces the accuracy of the machine learning model [10]. This is considered as a trade-off between utility and privacy and is usually measured as accuracy loss, describing the difference between the accuracy with and without differential privacy constraints. The accuracy loss, even on comparably simple tasks, such as the CIFAR-10 classification task, can be significant. This hinders the adoption of differentially private neural networks.

It is well known, that the architecture of a neural network can have a significant influence on the accuracy of the model. For example, expanding the size of a network in depth rather than width is usually said to increase accuracy [7], but also novel architectural features such as residual or dense connections can vastly improve model accuracy [8, 9]. However, early works on differentially private neural networks do not seem to account for this in great detail. They either find that the effect is not significant [2] or do not include different architectures in their benchmarks [10]. On the contrary, other research indicates that expanding the overall size of the network in a differentially private setting exhibits an inflection point [14]. Increasing the network size beyond this inflection point reduces model accuracy, which is not the case in the plain setting. Further, the choice of an activation function plays an important role. Bounded activation functions (e. g., tanh) consistently outperform unbounded ones (e. g., ReLU) due to the phenomena of exploding activations during differentially private training [15].

Preprint. Under review.

Nevertheless, even with these findings taken into account, there still remains a significant gap between the accuracy of differentially private and non-private neural networks, even on simple tasks. We hypothesize that model accuracy can be improved by carefully tuning a neural network architecture for differential privacy instead of simply copying architectures and hyperparameters that work well without differential privacy constraints [17]. Extant research has already shown that the choice of an appropriate activation function differs between the non-private and differentially private setting [15]. However, there are a multitude of other architectural features whose impact on model accuracy under differential privacy constraints has not been really understood. Consequently, we ask: *How does the architecture of a neural network affect the accuracy loss incurred by differential privacy constraints?*

In this paper we present initial findings from our research: We designed and carried out an experiment to test how the architecture affects accuracy loss under differential privacy constraints. Our findings show, not only, that the network architecture has an influence on the accuracy loss incurred by differential privacy constraints, but also, that the suitability of architecture choices is sensitive to variations in the targeted level of differential privacy. We elaborate the implications of our findings and argue how future research is necessary to improve the applicability of privacy-preserving machine learning.

2 Background

Differentially private stochastic gradient descent (DP-SGD) is an algorithm for training neural networks with (ϵ, δ)-differential privacy guarantees [2, 5]. DP-SGD extends classic stochastic gradient descent in two ways. First, gradients are clipped on a per-example basis to a l_2 norm, which is set through a clipping threshold C. Second, random noise is added to the gradient update calibrated via the standard deviation σ, also called noise multiplier. Thus, DP-SGD adds two additional hyperparameters to the training algorithm.

The privacy level (ϵ, δ) of a model trained with DP-SGD is only dependent on the batch size, the noise multiplier, the number of epochs, and the number of training examples. Hence, the privacy level for a given set of these hyperparameters can be calculated upfront without the need to actually train the model. This also makes the privacy level independent of the architecture of a given model and enables the comparison of different architectures at a fixed privacy level.

3 Experiments

To test our hypothesis, we conducted an experiment on the standard CIFAR-10 image classification task which contains 60000 color images in 10 classes [11]. We used Tensorflow [1] and the Tensorflow-Privacy extension [3] as machine learning libraries for the implementation. For our experiments we chose 9 convolutional neural network architectures, including prominent examples from the literature such as the LeNet-5 [12], but also architectures from other well-cited works [4, 13, 16] and own creations. The architectures differ mostly in the size and number of convolution, pooling, and fully-connected layers. We trained each model architecture with the differentially private stochastic gradient descent (DP-SGD) algorithm [2] and 18 different combinations of hyperparameters for 100 epochs each. The combinations result from a varying batch size of 250, 1000, and 2000; a noise multiplier of 0.01, 0.1, and 0.3; number of micro batches of 1 and 5; and a fixed learning rate of 0.1 as well as a fixed clipping threshold of 1.0.

4 Results

This setup resulted in 162 differentially private models. Analysis of the results revealed two distinct clusters, one where the models learn something useful and one where they do not. In this context, learning means that the validation accuracy improves with additional training epochs (until convergence). We excluded models that did not learn anything useful. Accordingly, we removed models where the ratio of maximum and initial validation accuracy is below 1.5 and the maximum validation accuracy does not reach 0.4. This filtering step reduces the number of models to 80.

In order to answer our research question, we need to show that the architecture of a neural network does have an effect on the model's accuracy loss under differential privacy constraints. We hypothesize that given a set of neural architectures for a given machine learning task, the architecture which

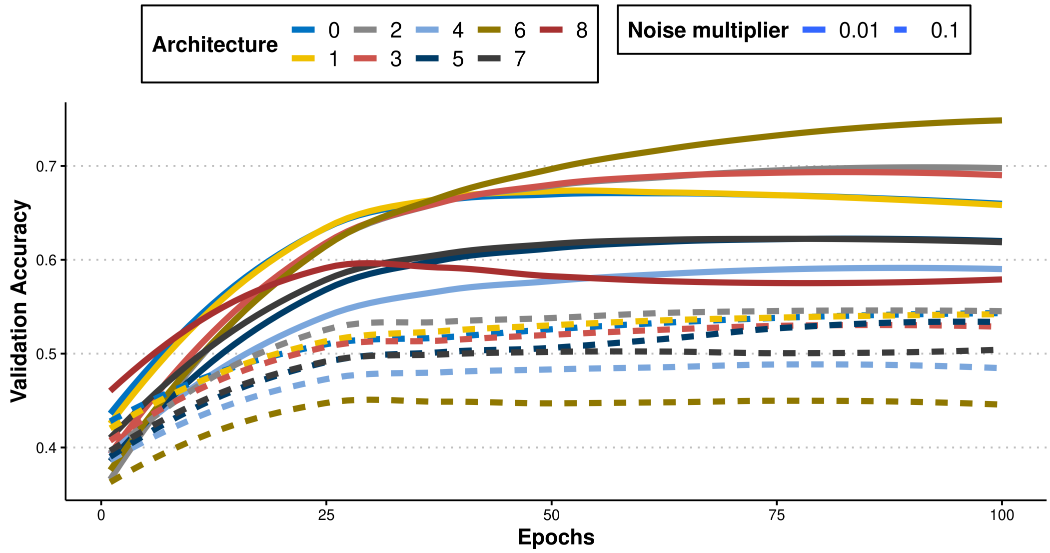

Figure 1: Experiment results for batch size 250 and 5 micro batches.

performs best without differential privacy constraints does not necessarily also performs best under differential privacy constraints. Or formulated differently, let there be two neural network architectures A_1 and A_2 and let $U(A)$ denote the accuracy of architecture A without differential privacy and $U_d(A)$ the accuracy with differential privacy. We assume the case exists that $U(A_1) > U(A_2)$ and $U_d(A_1) < U_d(A_2)$.

Figure 1 shows a subset of the results for training the 9 different architectures on the CIFAR-10 machine learning task, run in identically settings with two different noise multipliers. We can clearly see that with a low noise multiplier model 6 [4] performs best, but with a high noise multiplier model 6 comes in last while model 2 performs best. As the noise multiplier is one of the hyperparameters that determines the privacy level (besides epochs, batch size, and training set size, which are identical for all models in this subset), this means that at different privacy levels, different architectures perform best. This, in turn, shows that, in a given set of architectures, it is not always the case that a single architecture will perform best for all differential privacy settings. Therefore, it is also not guaranteed that an architecture that performs best without differential privacy necessarily performs best in the differential privacy setting.

5 Conclusion

In this paper, we investigated how the architecture of a neural network can influence the accuracy loss incurred through differential privacy constraints. Our findings show that neural network architectures that perform well in the non-private setting will not necessarily perform well in the differentially private setting. Further, we found that even between various levels of differential privacy, there are different architectures that perform best.

These findings imply that best practices, experiences, or architectures from the non-private setting cannot be easily transferred to the design of differentially private neural networks. But that they rather have to be specifically adjusted and tuned for the differential privacy setting. This also holds for different privacy levels.

The design of neural networks is a challenging task and either takes a lot of resources to try many different architectures through neural architecture search [6], or requires a lot of experience from the modeler. Doing a neural architecture search in the plain setting requires time for computations, but searching longer will not hurt the final accuracy of the model. However, searching for an architecture or optimizing hyperparameters consumes privacy budget in the differential private setting [17]. Therefore, spending more time on architecture search or hyperparameter optimization will decrease the privacy budget available for the actual training; hence, it will probably reduce the accuracy of the final model. Thus, a good understanding of how to design neural network architectures is, especially important in the differential privacy setting in order to maximize the available privacy budget for the actual training. However, as shown by our findings, the experience from the design of neural network architectures without differential privacy constraints can only be transferred to the design of differentially private neural network architectures to a limited degree. Therefore, in order to improve the applicability of privacy-preserving machine learning by reducing the accuracy loss incurred by differential privacy constraints, we need to derive new best practices for the design of architectures tailored specifically for differentially private neural networks.

In our future work, we will derive best practices for the design of neural architectures under differential privacy constraints. We will set up additional experiments across multiple classic machine learning benchmarks. Subsequently, we will analyse the results, derive a set of candidate best practices and test those on different benchmarks. We aim to also incorporate qualitative data from expert interviews to even further advance the best practices for designing neural network architectures that perform well under differential privacy constraints.

References

[1] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, Savannah, GA, November 2016. USENIX Association.

[2] Martin 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*, CCS '16, pages 308–318, New York, NY, USA, October 2016. Association for Computing Machinery.

[3] Galen Andrew, Steve Chien, and Nicolas Papernot. TensorFlow Privacy. https://github.com/tensorflow/privacy, 2018.

[4] Pathum Chamikara Mahawaga Arachchige, Peter Bertok, Ibrahim Khalil, Dongxi Liu, Seyit Camtepe, and Mohammed Atiquzzaman. Local differential privacy for deep learning. *IEEE Internet of Things Journal*, 7(7):5827–5842, 2019.

[5] Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, August 2014.

[6] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *The Journal of Machine Learning Research*, 20(1):1997–2017, January 2019.

[7] Ian J. Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks. In *2nd International Conference on Learning Representations (ICLR)*, 2014.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[9] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, July 2017.

[10] Bargav Jayaraman and David Evans. Evaluating Differentially Private Machine Learning in Practice. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1895–1912, 2019.

[11] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, 2009.

[12] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

[13] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282, Fort Lauderdale, FL, USA, April 2017. PMLR.

[14] Nicolas Papernot, Steve Chien, Shuang Song, Abhradeep Thakurta, and Ulfar Erlingsson. Making the Shoe Fit: Architectures, Initializations, and Tuning for Learning with Privacy, September 2019.

[15] Nicolas Papernot, Abhradeep Thakurta, Shuang Song, Steve Chien, and Úlfar Erlingsson. Tempered Sigmoid Activations for Deep Learning with Differential Privacy. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, 2021.

[16] NhatHai Phan, Xintao Wu, Han Hu, and Dejing Dou. Adaptive laplace mechanism: Differential privacy preservation in deep learning. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 385–394. IEEE, 2017.

[17] Koen Lennart van der Veen, Ruben Seggers, Peter Bloem, and Giorgio Patrini. Three Tools for Practical Differential Privacy. *arXiv:1812.02890 [cs, stat]*, December 2018.

[18] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why Gradient Clipping Accelerates Training: A Theoretical Justification for Adaptivity. In *Eighth International Conference on Learning Representations*, April 2020.