Analyzing CNN Architectures for Secure and Private Image Classification with Homomorphic Encryption and Differential Privacy

Robin Baumann[a], David Böhm[b], Raoul Saipt[b] and Astrid Laubenheimer[a]

[a]*Intelligent Systems Research Group, Karlsruhe University of Applied Sciences*
[b]*Karlsruhe University of Applied Sciences*

Abstract

Homomorphic encryption (HE) enables privacy-preserving deep learning, but it comes with significant performance overheads. In this study, we evaluate the impact of model architectures on the utility and efficiency of deep learning models under differential privacy (DP) and HE settings. Our experiments reveal that dedicated model architectures are crucial for maintaining model utility when using DP. Moreover, we observe that aligning complex model architectures like ResNets for HE by replacing ReLU with square activation, max pooling with average pooling, and group norm with batch norm strongly deteriorates model utility and results in architectures with sharp minima that fail to generalize. Training such models with DP, however, yields a regularizing effect that improves model utility. Our study contributes to the understanding of the role of model architecture on the applicability of DP and HE.

Keywords

Homomorphic Encryption, Differential Privacy, Deep Learning

1. Introduction

With the increasing importance of data privacy, there is a growing need for secure and private machine learning techniques that can protect sensitive data while still allowing for useful insights to be generated. Homomorphic encryption and differential privacy are two such techniques that have emerged as powerful tools for enabling secure and private machine learning [4, 6].

Homomorphic encryption (HE), first introduced in [30], allows for data to be encrypted in a way that preserves its mathematical structure, enabling computations to be performed on the encrypted data without first decrypting it. Differential privacy (DP) [10], on the other hand, provides a rigorous framework for protecting the privacy of individual data points, by adding random noise to the output of an algorithm in a way that preserves overall statistical properties of the data. Both techniques fulfill different aspects in privacy-preserving deep learning. HE works towards protecting the sensitive data, as well as the model weights from being exposed to adversaries during inference while DP obfuscates the training data in order to protect individual data owners from being exposed through the means of specifically designed attacks, like model inversion or membership inference. Following [4], we denote the former as input and model secrecy, respectively, and the latter as data privacy.

We consider a system framework in which both aspects are explicit requirements. More specifically, our framework comprises a trusted environment where we can train our AI models on sensitive data in plaintext with the consent of data owners. However, since we deal with sensitive personal data, we

CERC 2023: Collaborative European Research Conference, June 09–10, 2023, Barcelona, Spain

robin.baumann@h-ka.de (R. Baumann); astrid.laubenheimer@h-ka.de (A. Laubenheimer)
0009-0004-0961-4655 (R. Baumann); 0000-0001-7955-4521 (A. Laubenheimer)

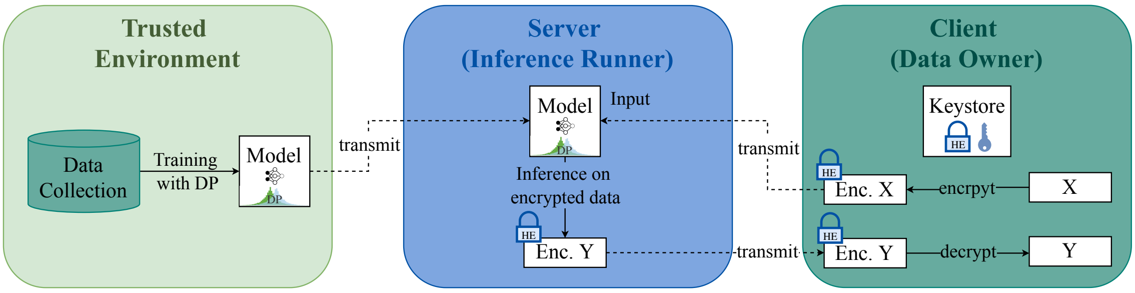


Figure 1: Overview of the investigated application scenario. We assume that all models are trained in a trusted environment with differential privacy (DP) in order to protect data owners. At inference time, we encrypt the data and evaluate the model on the ciphertext rather than on plaintext. The model can be encrypted additionally to prevent the weights from being exposed to adversaries.

still want to protect individuals and thus apply differential privacy during model training to ensure plausible deniability. The trained models can then be used for inference on a centralized server. In order to preserve the privacy of users during model inference, we apply homomorphic encryption. An overview of this setting is depicted in Figure 1. Using this design, the model can be deployed on a server that can be considered as an honest-but-curious adversary but still provide input and output privacy, as well as model secrecy.

Designing privacy-preserving AI solutions comprises several trade-offs between utility (i.e. prediction quality) and privacy, privacy and efficiency, and utility and efficiency, where efficiency includes both, speed and hardware utilisation. These trade-offs are already well-studied [4, 32] and several optimizations have been proposed in order to tighten the gap between plaintext and homomorphically encrypted image classification models [3, 5, 12, 33, 36] or designing more accurate model architectures for DP [8, 16, 20, 29, 31]. Little work has been published on the combined setting, i.e. optimizing deep learning architectures for accurate predictions under DP and fast inference using HE.

In this work, we compare several convolutional neural network (CNN) architectures regarding their utility on two image classification datasets with different data complexity, both with and without DP applied. We then exchange several building blocks of those networks in order to make them compatible with HE and repeat the experiments. We observe significant deterioration in model utility when aligning model architectures for HE that were originally designed for the application of DP. However, the application of DP seems to have regularizing effects on the model training, thus improving the results on the HE-aligned model architectures. Overall, this paper contributes to the growing body of work on secure and private machine learning, and provides insights into the practical considerations involved in designing and implementing these systems.

The remainder of this paper is structured as follows. Section 2 reviews related work in the areas of homomorphic encryption, differential privacy and analysis of deep learning model architectures. Section 3 describes the investigated model architectures and training procedure. In Section 4, we report our results and interpret and discuss them in Section 5 while also pointing out the limitations of this work. Finally, Section 6 concludes this work.

2. Related Work

In the following, we review literature published on homomorphic encryption and differential privacy with a specific focus on image classification applications.

2.1. Homomorphic Encryption

First introduced in [30], HE enables evaluating functions over encrypted data. Dowlin et al. [12] introduced CryptoNets, the first deep learning architecture specifically designed for HE. The architecture uses squared activation functions and scaled max pooling, but only consists of two convolutional layers. Chabanne et al. [5] extended CryptoNets to six layers by using a polynomial approximation of the ReLU activation function preceded by a batch normalization layer [15]. Although normalization in theory incorporates the computation of a square root, the approximation of this can be mitigated by reparameterizing the layer weights and biases prior to the normalization layer [14]. Badawi et al. [3] introduced the first homomorphically encrypted CNN that can run on a GPU. Nandakumar et al. [26] proposed a fully homomorphically encrypted training algorithm for deep neural networks. While all of the previously mentioned works evaluated their approaches only on MNIST or CIFAR-10, Wingarz et al. [33] investigated the scalability of HE for datasets with higher input and output dimensionality by leveraging parameter quantization and pre-processing for faster encryption. In general, HE induces an enormous computational overhead, especially on high resolution image data [33]. Therefore, all of the network architectures introduced in the publications mentioned above are significantly smaller than state of the art architectures in visual computing.

2.2. Differential Privacy

Differential privacy [10] is arguably the most popular data privacy mechanism, providing a mathematically rigorous privacy guarantee in the form of a privacy budget (ϵ, δ). This budget depends on the dataset size and number of training epochs. In deep learning, DP is usually applied to the gradients during optimization with the DP-SGD optimization algorithm [1], which obfuscates the exact gradients with noise sampled from a normal distribution. Due to this obfuscation, the application of differential privacy results in neural networks with a lower prediction accuracy opposed to non-private counterparts. Several model architectures have been proposed to reduce the loss in accuracy under DP. Proposed optimizations to the training process of established network architectures in order to enable DP on large-scale vision datasets include the application of weight standardization [28], replacing batch normalization with group normalization [34], increasing the batch size [24], applying parameter averaging [27] and pre-training on a non-private dataset [8, 20]. Remerscheid et al. proposed SmoothNets [29], a model family found by a neural architecture search subjected to various observations about the implications of specific hyperparameter choices on the model utility under DP. Especially the width-depth ratio when scaling neural networks seems to be of importance when using DP [8, 29].

2.3. Analyzing Deep Learning Architectures

Since deep learning optimization is highly non-convex, many local optima, as well as saddle points and plateaus exist in the loss landscape [7]. Li et al. [22] proposed a method to visualize the loss landscape of neural networks and investigate the implications of several architectural decisions, like the inclusion of skip connections, on the structure of the loss landscape. Dinh et al. [9] showed that flatness of minimizers is not necessarily correlated to the generalization ability of the network. Keskar et al. [17] reported a tendency of large batch optimization to converge to sharp minimizers, whereas small batch sizes seem to have a regularizing effect and converge to flat minimizers. Besides the topological structure of the loss landscape, other theories of generalization focus on norms expressed over the weight space [21], model compression under the PAC-Bayes Framework [23] and formulation of data-dependend error bounds [11].

3. Methodology

From the literature review we identified several orthogonal neural network architecture design decisions when optimizing either for DP or HE. While wide architectures, like wide ResNets [37], seem to outperform other architectures for DP [8], HE architectures are usually restricted in their width and depth in order to limit the number of multiplications in a forward pass. In addition, the application of the ReLU non-linearity requires a polynomial approximation in HE architectures, with square activation being the approximation with the lowest degree. Furthermore, max pooling is not applicable in HE and thus is usually replaced with (scaled) average pooling. Batch normalization layers can be reparameterized and are therefore applicable in both scenarios. However, several DP-optimized architectures include group normalization [34] instead of batch norm. Unfortunately, the reparameterization trick [14] does not generalize from batch normalization to group normalization, since the latter depends on on-the-fly computed statistics over channel groups, involving the computation of a square root. Using group normalization in HE therefore requires the approximation of a square root. In our experiments, we replaced group norm with batch norm when training models for HE.

3.1. Baseline Architectures

We implemented two baseline architectures, one optimized for DP and the other optimized for HE. In order to compare the effects of orthogonal model architecture designs on the opposing privacy regime, we adjusted the respective baseline architectures to meet the best practices from the other regime. Specifically, we removed restrictions on activation functions and pooling layers from the HE baseline and added those to the DP baseline. Furthermore, we implemented two variations of these baseline architectures as described below.

Our DP baseline model is inspired by [8] and is a Wide Residual Network (WRN) [37] with width scaling $k = 2$. Like [8], we replaced batch normalization with group normalization. The number of groups of the group normalization layers in the WRN Blocks is $\frac{C_{out}}{4}$, where C_{out} denotes the number of output channels of the preceding convolutional layer. See Table 1 for the concrete architecture in the plaintext and HE settings. Our variation of this model includes a thinner model with a width multiplicator of $k = \frac{1}{2}$ while also halving the number of filters in the first convolution layer (which in original Wide ResNet does not depend on k) in order to reduce the number of multiplications and make the application of HE feasible. We refer to these models as Wide ResNet and Tiny ResNet, respectively.

For our HE baseline, we adopted two distinct versions of CryptoNet [12]: the original version as described in [12], and an adapted variation that incorporates findings from the DP deep learning literature suggesting that increased width is a beneficial factor for DP training. Specifically, our adapted version features an increase in width by a factor of 3 for all convolutional layers of the baseline CryptoNet. We refer to the former model as CryptoNet and the latter as CryptoNet-L for the remainder of this paper.

3.2. Model training

We built and trained all models with TensorFlow and used the HeLayers Library for homomorphic encryption [2]. We performed our experiments on the Fashion-MNIST [35] and CIFAR-10 [19] datasets. Table 2 summizes the characteristics of both datasets. We trained all networks with batch sizes of 128 and learning rates of 10^{-3}. For non-DP experiments, we used the Adam Optimizer with default parameters [18], and for DP experiments, we used the differentially private counterpart of Adam im-

Table 1
DP-optimized Baseline architecture (left) and our alignment in order to support HE. Numbers in brackets denote number of filters in convolutional layers. Like [37], we use filter sizes of 3 × 3 in all Conv layers. We trained models for $k = 2$ and $k = \frac{1}{2}$. Our fully connected network (FCN) classifier consists of three layers with ReLU/Square activations and 64, 16, and 10 units respectively.

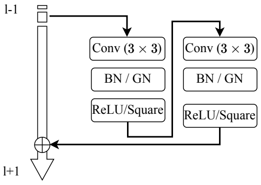
Layer	DP-optimized Architecure	HE-aligned Architecture	(HE-)WRN Block
1	Conv (16)	Conv (8)	 l-1 Conv (3 × 3) BN / GN ReLU/Square Conv (3 × 3) BN / GN ReLU/Square l+1
2	GroupNorm(4)	BatchNorm()	
3	ReLU()	Square()	
4	WRN Block 1 (16 × k)	HE-WRN Block 1 (16 × k)	
5	WRN Block 2 (32 × k)	HE-WRN Block 2 (32 × k)	
6	WRN Block 3 (64 × k)	HE-WRN Block 3 (64 × k)	
7	Average Pooling	Average Pooling	
8	Classifier FCN	Classifier FCN	

Table 2
Characteristics of the dataset under study.

Dataset	Contents	Image Dimensions	# Images (Train/Test)	# Classes
Fashion-MNIST	Clothing	28 × 28 × 1	60.000/10.000	10
CIFAR-10	Animals and Vehicles	32 × 32 × 3	50.000/10.000	10

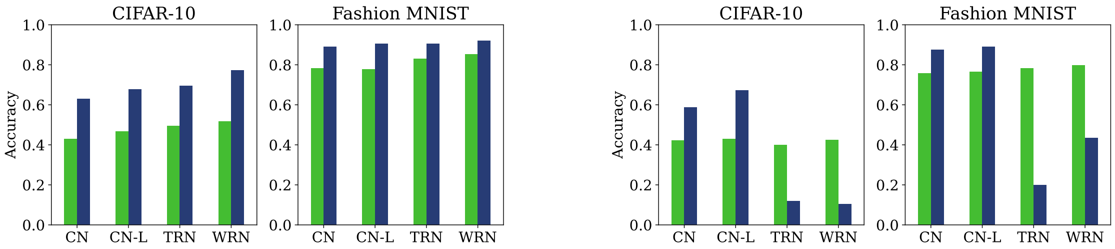
plemented in TensorFlow Privacy [13]. We trained all DP Models to suffice $(8, 10^{-5})$-DP on all datasets in the Rényi-DP setting [25].

4. Results

We conducted three experiments for which we report results in the following subsection. First, we trained all DP-optimized models, i.e. the baseline Wide ResNet, Tiny ResNet and both DP-aligned CryptoNets, with and without DP. After that, we repeated the process for our HE-optimized CryptoNets and the HE-aligned Wide ResNets. We also encrypted those models and evaluated their performance and relative inference speed drop compared to the plaintext models. Finally, we analyzed the loss landscape of the local minima identified for all models.

4.1. DP-optimized Architectures

Figure 2a summarizes the results on the DP-optimized Wide ResNets and the DP-aligned CryptoNets evaluated over the test sets of Fashion MNIST and CIFAR-10. We observe no significant differences in the performance of these models on the Fashion MNIST dataset when trained without DP. As expected, the performance gap is smaller for Wide ResNet and Tiny ResNet since those models are optimized for DP. Wide ResNet is performing best. On CIFAR-10, this finding replicates, but the gap between non-DP and DP models is larger across all models, indicating a correlation between dataset complexity and model performance under the application of DP.



(a) DP-optimized

(b) HE-aligned

Figure 2: Accuracies of all trained Models on the Fashion MNIST and CIFAR-10 datasets. 🟩 w/ DP, 🟦 w/o DP. CN(-L): CryptoNet(-L), WRN: Wide ResNet, TRN: Tiny ResNet

Table 3
RAM usage and ratio of inference time for inference on encrypted data. Both, model and data are encrypted.

Model	Fashion MNIST		CIFAR-10	
	RAM used	Inf. time scaling factor	RAM used	Inf. time scaling factor
CryptoNet	14.7 GB	158.0	12.7 GB	41.5
CryptoNet-L	31.3 GB	141.0	31.5 GB	68.7

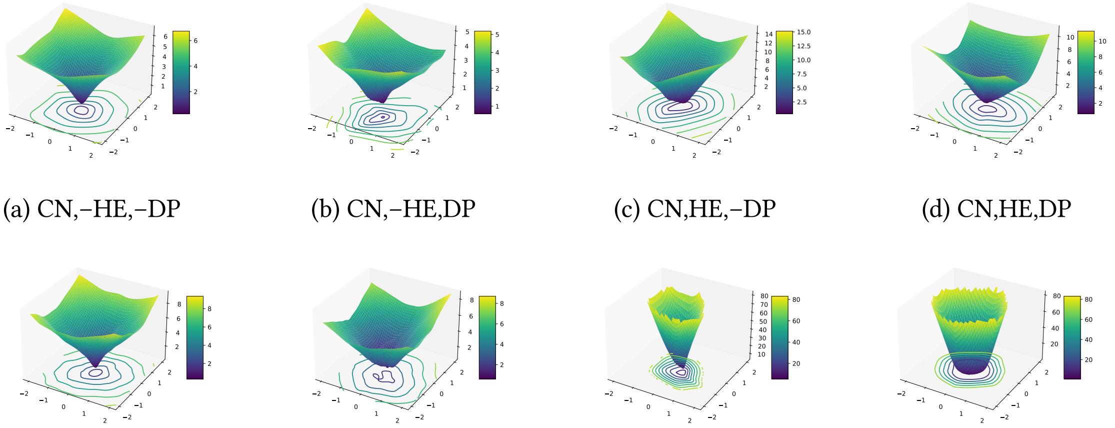
4.2. Homomorphic CNNs

Figure 2b shows the results for the HE-optimized CryptoNets and the HE-aligned Wide ResNet and Tiny ResNet. Notably, the performance of both CryptoNets does not change drastically when using HE building blocks instead of DP counterparts. In contrast, both ResNet architectures perform terrible in the HE-aligned setting. For CIFAR-10, the non-DP models completely fail to make accurate predictions on the test set. The predictions for Fashion MNIST are also poor, as can be seen by the large drop in prediction accuracy. Interestingly, the DP models perform better on both datasets, indicating a regularizing effect of DP on these model architectures. In Table 3 we report the relative difference in inference speed for all homomorphically encrypted networks compared to their plaintext counterpart, as well as used RAM for the prediction. We were unable to compute results for both ResNet architectures, since our workstation[1] ran out of memory for those models. We only report the average relative increase in inference time for a single data sample, since wall clock time depends strongly on the hardware used for evaluation.

4.3. Visualizations of local minima

We applied the visualization method proposed by Li et al. [22] to all models and evaluated the implications of DP/HE-alignment on the structure of the loss landscapes. The results are depicted in Figure 3. We observe different effects for HE-aligned models, as well as for models trained with DP and without DP. For CryptoNet and Tiny ResNet, the application of DP introduces visible distortions to the loss-landscape in the model architectures without HE-alignment (subfigures 3b and 3f) compared to the non-DP models (subfigures 3a and 3e). This can be particularly observed in the contour lines that are projected onto the xy-planes. When applying both, HE and DP, the loss surface is less distorted (subfigures 3d and 3h). Across all models, the application of HE yields steeper valleys and thus, sharper minimizers (subfigures 3c, 3d, 3g and 3h), while the application of DP in this setting

[1] Intel® Xeon® Silver 4114 CPU 2.20 GHz and 64 GB of DDR4 RAM



(a) CN,−HE,−DP

(b) CN,−HE,DP

(c) CN,HE,−DP

(d) CN,HE,DP

(e) TRN,−HE,−DP

(f) TRN,−HE,DP

(g) TRN,HE,−DP

(h) TRN,HE,DP

Figure 3: Loss-Landscapes for CryptoNets (a-d) and Tiny ResNet (e-h). Best viewed digitally and zoomed in. All z-values are on log-scale. Abbreviations in subfigure captions are: CN: CryptoNet, TRN: Tiny ResNet, HE: Architecture for HE, DP: model trained with DP. −HE and −DP means without HE or DP, respectively. Models are trained and evaluated on CIFAR-10. Loss values are computed using categorical cross-entropy.

seems to attenuate this effect and thus regularize the optimization in the HE setting (subfigures 3d and 3h), especially for Tiny ResNet.

5. Discussion

Our results confirm the importance of dedicated model architectures for the application of DP in order to close the performance deterioration induced by DP. Wide ResNet and Tiny ResNet did perform better in the non-HE setting than both CryptoNets in our experiments. We note at this point that our Tiny ResNet comprises ~29k parameters, whereas CryptoNet comprises ~28k parameters. Both models have significantly less parameters than CryptoNet-L (~119k) and Wide ResNet (~336k). Since Tiny ResNet outperforms CryptoNet-L with and without DP, the parameter count of a model does not seem to be the most important architectural property for model utility. Our results indicate that the network topology has a direct impact on utility when applying DP. Especially on Fashion MNIST, the utility gap between the DP and non-DP models is smaller compared to the CryptoNets. However, on CIFAR-10 this result is not as straightforward. While both ResNets outperform CryptoNets, the gap between DP and non-DP models is approximately similar for all models, indicating that regarding more complex datasets other tools are required to close this gap. Towards this, [8, 20] have leveraged data augmentation, learning rate schedules, and other tweaks to the training procedure in order to obtain tighter utility gaps. Hence, architecture alone can help to close this gap, but does not suffice on its own.

Aligning more complex model architectures like ResNets for homomorphic encryption strongly deteriorates model utility. We have observed that replacing ReLU with square activation, max pooling with average pooling, and group norm with batch norm results in model architectures with sharp minima which in our experiments completely failed to generalize to the test set. Training these models with DP, however, seems to yield a regularizing effect that increases model utility to a level that is competitive with the results of DP-trained CryptoNets. However, encrypted evaluation of both ResNet architectures failed using the HELayers library [2] because of memory restrictions. Presumably, this is due to the presence of skip connections. Without these, each layer including its corresponding inputs can be loaded separately into the memory. Concerning skip connections, the intermediary results of multiple layers need to be kept in memory, thus resulting in higher RAM allocations that exceeded our 64GB main memory even for the Tiny ResNet. We note that one could evade this problem through engineering effort, i.e., providing more efficient implementations for the inference of skip connections. However, we have not conducted any further investigation and leave it to future work.

Overall, we note that the increase in inference time and RAM usage is significant for the application of HE on deep learning, even for small models as CryptoNets on small toy datasets such as Fashion-MNIST and CIFAR-10. While related work such as [33] have investigated the scalability of HE-enabled deep learning on datasets with higher complexity and dimensionality, they do not consider complex network architectures as our Tiny ResNet, which yield better results in the plaintext setting while having an approximately similar amount of parameters. In the computer vision literature, residual networks and other complex architectures with millions of parameters yield state of the art results. Given the computational complexity and resource requirements for our small networks, applying HE to state of the art models is intractable in real world use cases.

5.1. Limitations

We have limited our study to minimal requirements for homomorphic encryption. Existing work on HE-enabled deep learning comprises model architectures with polynomial approximations of activation functions of higher degree. Using these approximations could close the observed gap between the DP-optimized architectures and their HE-aligned counterpart. On the other hand, it may lead to further increase in computational complexity of the inference with homomorphically encrypted data. Future work should investigate upon that. In addition to that, the loss landscape visualization that we have used only provides a high level idea about the generalizability of the model architectures. Other methods have been proposed in the literature that may give more insight. However, this is still a very active area of research.

6. Conclusion

In conclusion, our research findings underline the importance of dedicated model architectures for the application of differential privacy (DP) in order to close the performance deterioration induced by DP. Specifically, we have observed that Wide ResNet and Tiny ResNet perform better in the non-HE setting than CryptoNets. Moreover, the network topology seems to have a great impact on utility when applying DP. While Tiny ResNet outperforms CryptoNet-L with and without DP, our results indicate that the number of model parameters is not the most significant architectural property regarding model utility. Furthermore, we have observed that aligning more complex model architectures like ResNets for homomorphic encryption strongly deteriorates model utility. While related work has investigated the scalability of HE-enabled deep learning on datasets with higher complexity and dimensionality, they do not consider complex network architectures like our Tiny ResNet, which yield better results in the plaintext setting while having an approximately similar amount of parameters. However, evaluating ResNets in an encrypted setting drastically increases the resource requirements compared to CryptoNets, thus rendering homomorphic encryption impractical for these model architectures.

References

[1] Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. p. 308–318. CCS '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2976749.2978318

[2] Aharoni, E., Adir, A., Baruch, M., Drucker, N., Ezov, G., Farkash, A., Greenberg, L., Masalha, R., Moshkowich, G., Murik, D., Shaul, H., Soceanu, O.: HeLayers: A Tile Tensors Framework for Large Neural Networks on Encrypted Data. Privacy Enhancing Technology Symposium (PETs) 2023 (2023), https://petsymposium.org/2023/paperlist.php

[3] Al Badawi, A., Jin, C., Lin, J., Mun, C.F., Jie, S.J., Tan, B.H.M., Nan, X., Aung, K.M.M., Chandrasekhar, V.R.: Towards the alexnet moment for homomorphic encryption: Hcnn, the first homomorphic cnn on encrypted data with gpus. IEEE Transactions on Emerging Topics in Computing **9**(3), 1330–1343 (2021). https://doi.org/10.1109/TETC.2020.3014636

[4] Cabrero-Holgueras, J., Pastrana, S.: SoK: Privacy-Preserving Computation Techniques for Deep Learning. Proceedings on Privacy Enhancing Technologies (2021), https://petsymposium.org/popets/2021/popets-2021-0064.php

[5] Chabanne, H., de Wargny, A., Milgram, J., Morel, C., Prouff, E.: Privacy-preserving classification on deep neural network. Cryptology ePrint Archive, Paper 2017/035 (2017), https://eprint.iacr.org/2017/035

[6] Chen, H., Hussain, S.U., Boemer, F., Stapf, E., Sadeghi, A.R., Koushanfar, F., Cammarota, R.: Developing Privacy-preserving AI Systems: The Lessons learned. In: 2020 57th ACM/IEEE Design Automation Conference (DAC). pp. 1–4 (Jul 2020). https://doi.org/10.1109/DAC18072.2020.9218662, iSSN: 0738-100X

[7] Dauphin, Y.N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., Bengio, Y.: Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In: Proceedings of the 27th International Conference on Neural Information Processing Systems. vol. 2, p. 2933–2941. MIT Press, Cambridge, MA, USA (2014)

[8] De, S., Berrada, L., Hayes, J., Smith, S.L., Balle, B.: Unlocking High-Accuracy Differentially Private Image Classification through Scale (Jun 2022). https://doi.org/10.48550/arXiv.2204.13650

[9] Dinh, L., Pascanu, R., Bengio, S., Bengio, Y.: Sharp minima can generalize for deep nets. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 1019–1028. PMLR (06–11 Aug 2017), https://proceedings.mlr.press/v70/dinh17b.html

[10] Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) Automata, Languages and Programming. pp. 1–12. Springer Berlin Heidelberg (2006)

[11] Dziugaite, G.K., Roy, D.M.: Data-dependent pac-bayes priors via differential privacy. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018), https://proceedings.neurips.cc/paper/2018/file/9a0ee0a9e7a42d2d69b8f86b3a0756b1-Paper.pdf

[12] Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., Wernsing, J.: Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In: Balcan, M.F., Weinberger, K.Q. (eds.) Proceedings of The 33rd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 48, pp. 201–210. PMLR, New York, New York, USA (20–22 Jun 2016), https://proceedings.mlr.press/v48/gilad-bachrach16.html

[13] Google: Tensorflow privacy. GitHub Repository: https://github.com/tensorflow/privacy (2018)

[14] Ibarrondo, A., Önen, M.: Fhe-compatible batch normalization for privacy preserving deep learn-

ing. In: Garcia-Alfaro, J., Herrera-Joancomartí, J., Livraga, G., Rios, R. (eds.) Data Privacy Management, Cryptocurrencies and Blockchain Technology. pp. 389–404. Springer International Publishing (2018)

[15] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Bach, F., Blei, D. (eds.) Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 448–456. PMLR, Lille, France (07–09 Jul 2015), https://proceedings.mlr.press/v37/ioffe15.html

[16] Kaissis, G., Ziller, A., Passerat-Palmbach, J., Ryffel, T., Usynin, D., Trask, A., Lima, I., Mancuso, J., Jungmann, F., Steinborn, M.M., Saleh, A., Makowski, M., Rueckert, D., Braren, R.: End-to-end privacy preserving deep learning on multi-institutional medical imaging. Nature Machine Intelligence **3**(6), 473–484 (Jun 2021). https://doi.org/10.1038/s42256-021-00337-8

[17] Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P.T.P.: On large-batch training for deep learning: Generalization gap and sharp minima. In: International Conference on Learning Representations (2017)

[18] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations (2015)

[19] Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep. (2009)

[20] Kurakin, A., Song, S., Chien, S., Geambasu, R., Terzis, A., Thakurta, A.: Toward Training at ImageNet Scale with Differential Privacy (Feb 2022). https://doi.org/10.48550/arXiv.2201.12328

[21] Ledent, A., Mustafa, W., Lei, Y., Kloft, M.: Norm-based generalisation bounds for deep multi-class convolutional neural networks. Proceedings of the AAAI Conference on Artificial Intelligence **35**(9), 8279–8287 (May 2021). https://doi.org/10.1609/aaai.v35i9.17007

[22] Li, H., Xu, Z., Taylor, G., Studer, C., Goldstein, T.: Visualizing the loss landscape of neural nets. In: Neural Information Processing Systems (2018)

[23] Lotfi, S., Finzi, M.A., Kapoor, S., Potapczynski, A., Goldblum, M., Wilson, A.G.: PAC-bayes compression bounds so tight that they can explain generalization. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) Advances in Neural Information Processing Systems (2022)

[24] McMahan, H.B., Ramage, D., Talwar, K., Zhang, L.: Learning differentially private recurrent language models. In: International Conference on Learning Representations (2017)

[25] Mironov, I.: Rényi differential privacy. In: 2017 IEEE 30th Computer Security Foundations Symposium (CSF). pp. 263–275 (2017). https://doi.org/10.1109/CSF.2017.11

[26] Nandakumar, K., Ratha, N., Pankanti, S., Halevi, S.: Towards deep neural network training on encrypted data. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 40–48 (2019). https://doi.org/10.1109/CVPRW.2019.00011

[27] Polyak, B.T., Juditsky, A.B.: Acceleration of stochastic approximation by averaging. SIAM Journal on Control and Optimization **30**(4), 838–855 (1992). https://doi.org/10.1137/0330046

[28] Qiao, S., Wang, H., Liu, C., Shen, W., Yuille, A.: Micro-batch training with batch-channel normalization and weight standardization (2019). https://doi.org/10.48550/ARXIV.1903.10520

[29] Remerscheid, N.W., Ziller, A., Rueckert, D., Kaissis, G.: Smoothnets: Optimizing cnn architecture design for differentially private deep learning. arXiv preprint arXiv:2205.04095 (2022)

[30] Rivest, R.L., Adleman, L., Dertouzos, M.L.: On data banks and privacy homomorphisms. Foundations of Secure Computation, Academia Press pp. 169–179 (1978)

[31] Sander, T., Stock, P., Sablayrolles, A.: Tan without a burn: Scaling laws of dp-sgd. arXiv preprint arXiv:2210.03403 (2022)

[32] Torkzadehmahani, R., Nasirigerdeh, R., Blumenthal, D.B., Kacprowski, T., List, M., Matschinske, J., Spaeth, J., Wenke, N.K., Baumbach, J.: Privacy-Preserving Artificial Intelligence Techniques in Biomedicine. Methods of Information in Medicine **61**, e12–e27 (Jan 2022).

https://doi.org/10.1055/s-0041-1740630

[33] Wingarz, T., Gomez-Barrero, M., Busch, C., Fischer, M.: Privacy-Preserving Convolutional Neural Networks Using Homomorphic Encryption. In: 2022 International Workshop on Biometrics and Forensics (IWBF). pp. 1–6 (Apr 2022). https://doi.org/10.1109/IWBF55382.2022.9794535

[34] Wu, Y., He, K.: Group normalization. International Journal of Computer Vision **128**, 742–755 (2018)

[35] Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. CoRR **abs/1708.07747** (2017), http://arxiv.org/abs/1708.07747

[36] Xiong, A., Nguyen, M., So, A., Chen, T.: Privacy Preserving Inference with Convolutional Neural Network Ensemble. In: 2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC). pp. 1–6 (Nov 2020). https://doi.org/10.1109/IPCCC50635.2020.9391544, iSSN: 2374-9628

[37] Zagoruyko, S., Komodakis, N.: Wide residual networks. In: British Machine Vision Conference (2016)