AI in Medicine I

Practical Exercise 5

Trustworthy AI: Federated and Privacy-preserving ML

Submitted to the

Department of Informatics
I31 - AI in Medicine and Healthcare
I32 - Computational Imaging in AI and Medicine

at TUM

by

Argudo, Mateo (03717216)
Bolaños, Daniela (03765336)
Frey, Daniel (03632203)
Otto, Julia (03698083)

Sadly, one member of our group dropped out shortly before handing in the exercise. For this reason, we were not able to complete the last exercise. In the short time we had, we tried our best to achieve as much as we can. Please take the unfortunate circumstances into consideration when grading.

January 26, 2024

1 Differentially Private Model Training

The impact of ε-δ-differential privacy (DP) on training and inference of neural networks is analyzed based on the `opacus` codebase. For context, DP is defined as follows:

$$\Pr\left[\mathcal{A}(D_1) \in \mathcal{S}\right] \leq \mathrm{e}^{\varepsilon} \cdot \Pr\left[\mathcal{A}(D_2) \in \mathcal{S}\right] + \delta. \tag{1}$$

Here, ε defines an upper boundary for the ratio of probabilities Pr processed by an algorithm $\mathcal{A}$. For all subsets S of the image im $\mathcal{A}$, the randomness introduced to two datasets D_1 and D_2, which only differ by one element, enables privatization of vulnerable data. The parameter δ allows deviations from that strict boundary.

1.1 Differentially Private Image Classification

A pretrained ResNet-18 model (after deactivating all BatchNorm layers and adapting the fully connected layer to ten output features) was trained on the CIFAR-10 dataset (split 70%–15%–15% for training–validation–test). For both non-DP and DP training, conventional cross-entropy loss function and Adam optimizer were

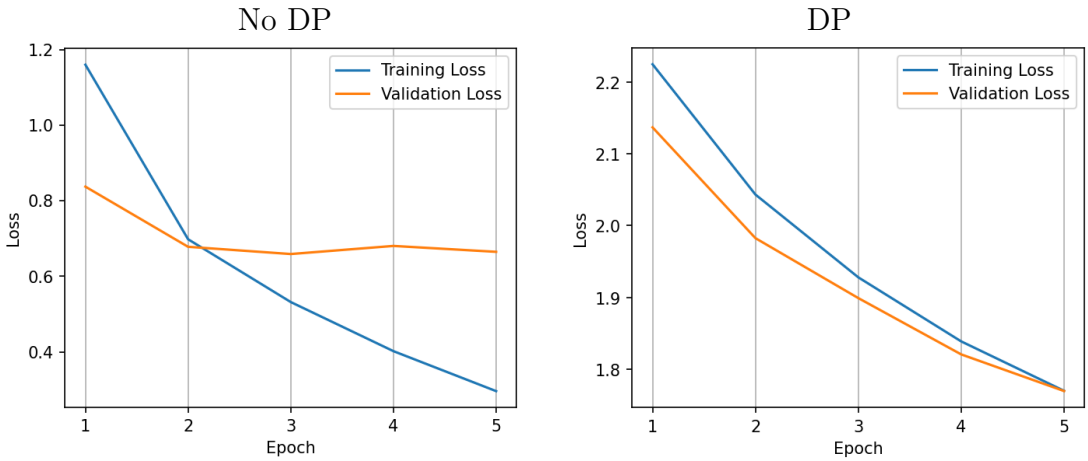

Figure 1: ResNet-18 loss curves for CIFAR-10 without and with DP.

Model	Dataset	Time	Accuracy	`epsilon`	`delta`
ResNet-18	CIFAR-10	308 s	37.76 %	2.0	1×10^{-3}
ResNet-18	CIFAR-10	104 s	70.22 %	inf	0

Table 1: ResNet-18 training time and performance with and without DP.

applied, and hyperparameters (`batch_size=64`, `learning_rate=1e-4`, `epochs=5`) were kept the same. For DP, the method `make_private_with_epsilon` was used, which allowed to define a target for both ε and δ alongside a maximum gradient cap of `max_grad_norm=10.0`. Training results are depicted in Figure 1. DP adds computational effort to the training, as reflected by the relative increase of training time (almost 3 times as long) concurrent with a significant loss of performance (almost half of its counterpart) during inference. Judging from the on-going downward trend of the loss curves, DP seems to demand prolonged training times at otherwise identical conditions in order to preserve performance. An overview of the results is provided in Table 1.

1.2 Applying DP-SGD to Medical Data

We wanted to test how DP affects neural-network performance on medical datasets. In this case, we used the same hyperparameters, loss function and optimizer as in 1.1. We considered the datasets PathMNIST (9 classes), DermaMNIST (7 classes) and BreastMNIST (2 classes). Splits were applied as provided by default. For the model, the number of output classes needed to be adapted to each dataset individually to match the number of classes. The loss curves are depicted in Figure 2, 3 and 4, and accuracy and training time results are listed in Table 2. For all the training curves, the ones with DP delivered higher losses for the same batches. Moreover, in the cases of Figure 3 and 4, the loss even goes up. We also overfit quickly ($\sim$ 5 epochs).

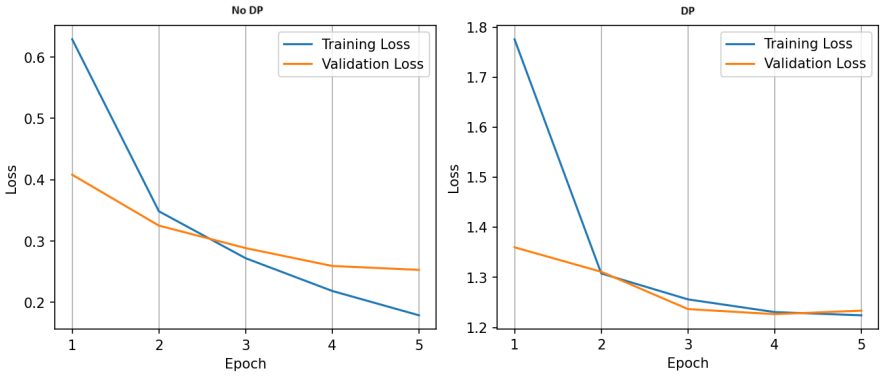

Figure 2: ResNet-18 loss curves for PathMNIST without and with DP.

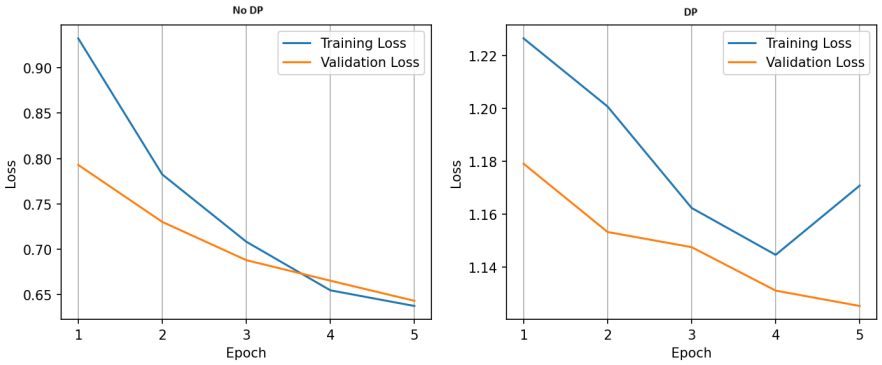

Figure 3: ResNet-18 loss curves for DermaMNIST without and with DP.

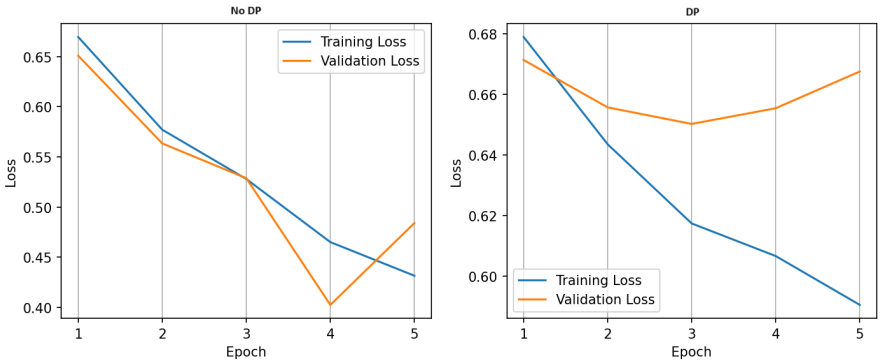

Figure 4: ResNet-18 loss curves for BreastMNIST without and with DP.

Model	Dataset	Time	Accuracy	`epsilon`	`delta`
ResNet-18	PathMNIST	763 s	66.35 %	2.0	1×10^{-3}
ResNet-18	PathMNIST	239 s	81.96 %	inf	0
ResNet-18	DermaMNIST	65 s	66.78 %	2.0	1×10^{-3}
ResNet-18	DermaMNIST	22 s	75.68 %	inf	0
ResNet-18	BreastMNIST	12 s	73.44 %	2.0	1×10^{-3}
ResNet-18	BreastMNIST	4 s	71.73 %	inf	0

Table 2: ResNet-18 training time and performance with and without DP for different Medical Datasets.

The time and accuracy also seems affected when using DP. Training on each dataset takes about three times longer with DP. For PathMNIST and DermaMNIST, we see a loss of accuracy of around 15% and 9%. In the case of BreastMNIST, the accuracy improved slightly by around 2%. This is rather unexpected and might have to do with the binary task as opposed to multi-label classification.

1.3 Analyzing the Parameters of Private Learning

In this subsection, we wanted to analyzed different parameters that were not different before. In order to experiment with different parameters and see how those can affects the utility of the model, we decided to take different values of epsilon, in different levels, e.g., in the BreastMNIST Dataset shown in the Table 3. However, the change of parameters only could be until an epsilon of 4.0, due to constraints of the library (Opacus: Discrete Mean differs from Coninous Mean Error). It was also noted that lower deltas ($\delta = 1 \times 10^{-6}$) enabled us to explore higher epsilons, but we decided not to test $\epsilon = 8$ since it required $\delta << 1 \times 10^{-12}$ for the privacy accountant of Opacus not to crash, which is too small.

1.3.1 BreastMNIST Dataset

First, the evaluation of the epsilon parameters were chose according to the privacy levels (low, medium, high) from 0.5 to 4.0, highest and lowest respectively; and a delta value of 1×10^{-6}, which was chose according with the capacity of the running to get a good performance until the highest epsilon level (see Table 3). Then, we observed, according to the Figure 5, that the test accuracy increased from around 66% to around 73% , but it kept constant in the medium and high privacy level, which means the privacy level ϵ affects the metrics of the model, because we expected lower values with the decrease of epsilon, in order to get more privacy. Nevertheless, when we observed the Figure 6, we see that while epsilon is increasing, also the training time; which is a result we do not expect to perform according to the parameter, because with higher privacy there should be more computations needed.

We also wanted to experiment not only regarding epsilon parameters, but also regarding the noise multiplier parameters. To do that, we used a different parameters having into account the privacy levels (see Table 4). We can see in the

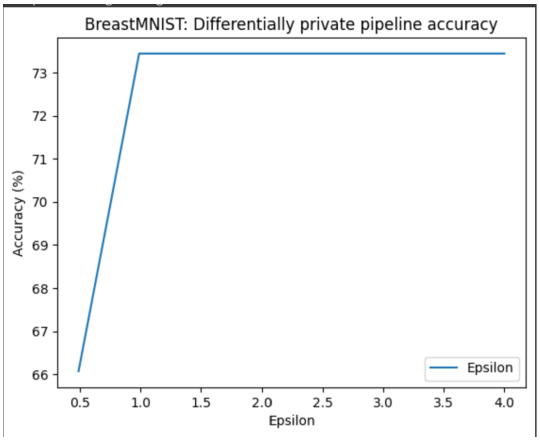

Figure 5: Differencial Private accuracy vs privacy level curve for BreastMNIST Dataset

Privacy parameter	Time	Accuracy	`epsilon`	`delta`
Epsilon	10 s	66.07 %	0.49	1×10^{-6}
Epsilon	9 s	73.44 %	0.99	1×10^{-6}
Epsilon	9 s	73.44 %	1.49	1×10^{-6}
Epsilon	10 s	73.44 %	2.0	1×10^{-6}
Epsilon	12 s	73.44 %	4.0	1×10^{-6}

Table 3: Privace learning settings with epsilon testing for BreastMNIST Dataset

Privacy parameter	Noise	Time	Accuracy	`epsilon`	`delta`
Noise Multiplier	1.0	7 s	73.44 %	0.31	1×10^{-6}
Noise Multiplier	1.5	7 s	73.44 %	3.13	1×10^{-6}
Noise Multiplier	2.5	7 s	67.41 %	1.52	1×10^{-6}
Noise Multiplier	3.5	7 s	73.44 %	1.00	1×10^{-6}
Noise Multiplier	4.5	7 s	67.41 %	1.52	1×10^{-6}

Table 4: Privace learning settings with Noise Multiplier testing for BreastMNIST Dataset

Figure 7, while the noise multiplier parameter is increasing, the epsilon decreases; which allows to conclude there will be more privacy.

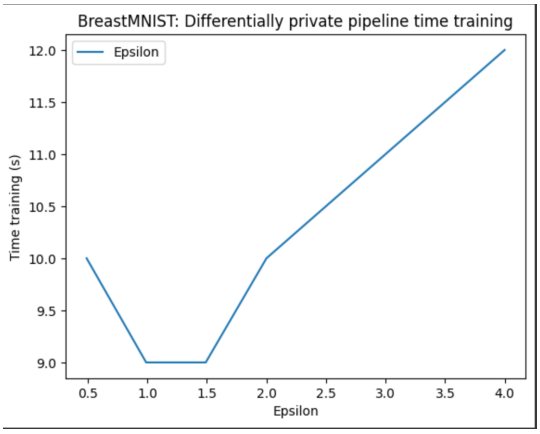

Figure 6: Differencial Private training time vs privacy levelcurve for BreastMNIST Dataset

1.3.2 PathMNIST Dataset

In this big dataset, we observed in the Figure 9 a different behaviour respect to the training time when we increase the epsilon parameter, which means, this result is what we expect in terms of privacy, since a higher privacy epsilon level, we could observe a higher time consuming.

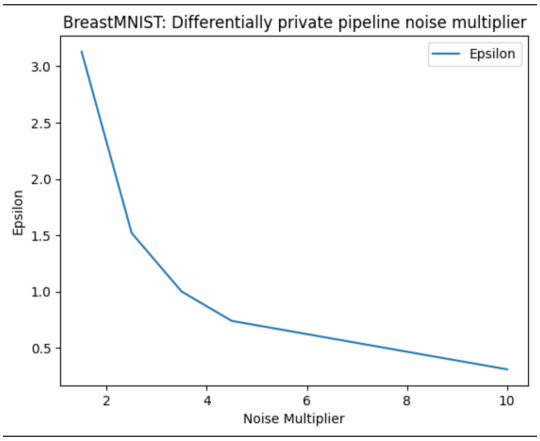

Figure 7: Differencial Private noise multiplier vs privacy level curve for BreastMNIST Dataset

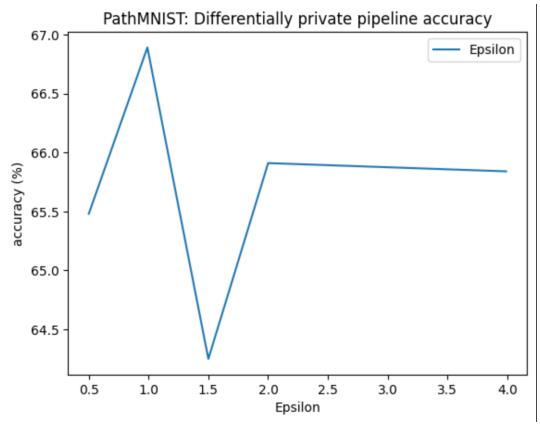

Figure 8: Differencial Private accuracy vs privacy level curve for PathMNIST Dataset

Privacy parameter	Time	Accuracy	`epsilon`	`delta`
Epsilon	15.53 min	65.48 %	0.50	1×10^{-6}
Epsilon	13.05 min	66.89 %	0.99	1×10^{-6}
Epsilon	12.50 min	64.25 %	1.50	1×10^{-6}
Epsilon	13.03 min	65.91 %	2.0	1×10^{-6}
Epsilon	12.56 min	65.84 %	3.99	1×10^{-6}

Table 5: Privace learning settings with epsilon testing for PathMNIST Dataset

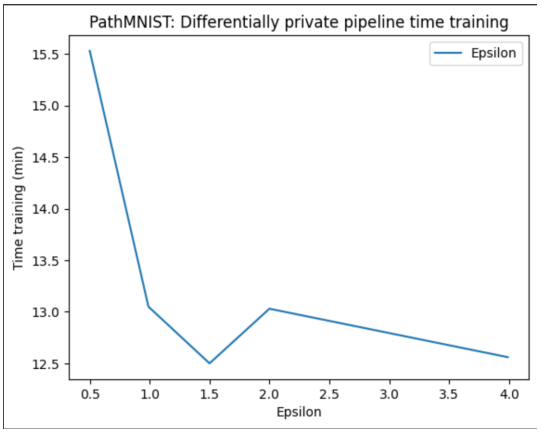

Figure 9: Differencial Private training time vs privacy level curve for PathMNIST Dataset

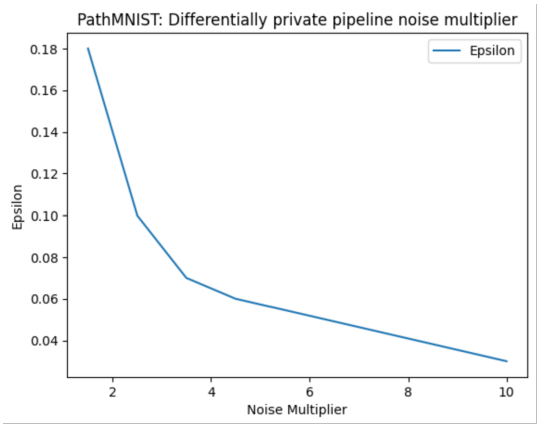

Figure 10: Differencial Private noise multiplier vs privacy level curve for PathMNIST Dataset

Privacy parameter	Noise	Time	Accuracy	`epsilon`	`delta`
Noise Multiplier	1.0	12.44 min	39.74 %	0.03	1×10^{-6}
Noise Multiplier	1.5	12.48 min	62.32 %	0.18	1×10^{-6}
Noise Multiplier	2.5	12.53 min	57.31 %	0.10	1×10^{-6}
Noise Multiplier	3.5	7 s	50.02 %	0.07	1×10^{-6}
Noise Multiplier	4.5	7 s	50.40 %	0.06	1×10^{-6}

Table 6: Privace learning settings with Noise Multiplier testing for PathMNIST Dataset

1.3.3 DermaMNIST Dataset

In this dataset, we did not see any difference in terms of the accuracy of the model within different privacy epsilon levels. We observed a similar result as the other datasets, which we do not expect.

In general, we could observed that epsilon and Noise Multiplier parameters are very important to determine certain privacy levels of the model, and to get a better idea about the accuracy of the model.

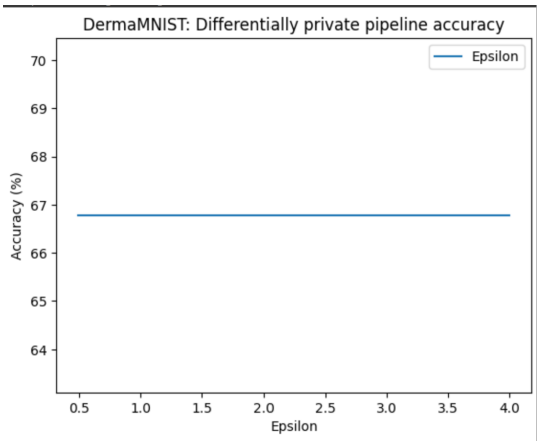

Figure 11: Differencial Private accuracy vs privacy level curve for DermaMNIST Dataset

Privacy parameter	Time	Accuracy	`epsilon`	`delta`
Epsilon	10 s	66.78 %	0.49	1×10^{-6}
Epsilon	9 s	66.78 %	1.00	1×10^{-6}
Epsilon	9 s	66.78 %	1.50	1×10^{-6}
Epsilon	10 s	66.78 %	2.0	1×10^{-6}
Epsilon	12 s	66.78 %	4.0	1×10^{-6}

Table 7: Privace learning settings with epsilon testing for DermaMNIST Dataset

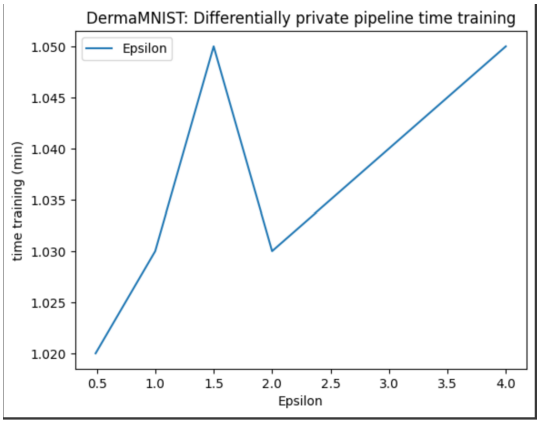

Figure 12: Differencial Private training time vs privacy level curve for DermaMNIST Dataset

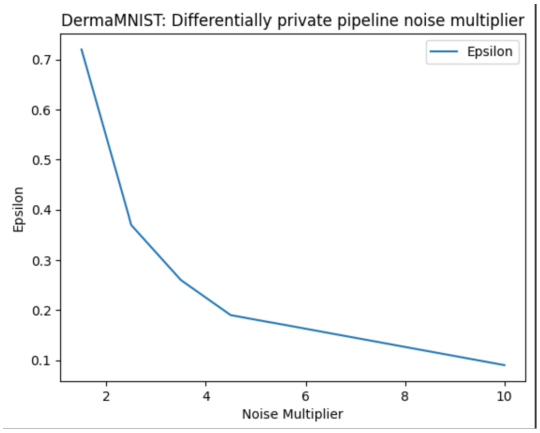

Figure 13: Differencial Private noise multiplier vs privacy level curve for DermaMNIST Dataset

Privacy parameter	Noise	Time	Accuracy	`epsilon`	`delta`
Noise Multiplier	1.0	7 s	73.44 %	0.31	1×10^{-3}
Noise Multiplier	1.5	7 s	73.44 %	3.13	1×10^{-3}
Noise Multiplier	2.5	7 s	67.41 %	1.52	1×10^{-3}
Noise Multiplier	3.5	7 s	73.44 %	1.00	1×10^{-3}
Noise Multiplier	4.5	7 s	67.41 %	1.52	1×10^{-3}

Table 8: Privace learning settings with Noise Multiplier testing for DermaMNIST Dataset

2 Threats of Federated Learning

2.1 Implementing Federated Learning Chest X-ray Classification

The idea of this exercise is to simulate federated learning. This is especially useful in order to decentralize the handling of sensible data, as this allows to keep and work with the data at the place the data is collected.

In this exercise a CNN model is trained to make a binary prediction on the ChestMNIST dataset. In the first part we search for the best hyperparameters (learning rate, batch size, optimizer and number of epochs) based on three clients. The influence of each parameter is also evaluated. In the second part we change the number of clients incrementally from three to ten and compare the individual and global models performance.

The hyperparameter tuning shows that a smaller learning rate allows a higher accuracy. The behavior of the clients does not change strongly, when tuning the learning rate. But a different batch size allows for a more constant behavior across all clients and smaller changes across the different rounds. Two optimizers have been tested: Adam and SGD. Adam performs better than SGD, improving an accuracy of 73% to 84%. The number of epochs can be optimized to 20, again this

leads to the clients performance converging closer together. The number of rounds reaches its best performance on eight rounds. The best accuracy after tuning all hyperparameter combinations is 89%.

When training on double the size of the clients, it must be noted that the split dataset is smaller for each client. For this reason it makes sense, that the accuracy of some individual clients is smaller. The global accuracy stays about the same, but the accuracy reaches its peak and then stays relatively constant at an earlier round, round six. Thus we can conclude that a higher number of clients, with less data each, show a more varying accuracy in this case, but the model converges faster to the highest accuracy achieved with the chosen optimized parameters.

2.2 Adversarial Samples in Federated Learning

We had no time left to solve this exercise because one member left the group shortly before submission.