AI in Medicine I
WiSe 2022/23

Technical University Munich
Department of Informatics

I31 - AI in Medicine and Healthcare
I32 - Computational Imaging in AI and Medicine

Practical exercise 11 **11. Jan. 2024**

Trustworthy AI: Federated and privacy-preserving ML

Submission deadline: 25. Jan. 2024, 11 p.m.

Please submit your solutions (via Moodle).
The corresponding tutorial session is

11. Jan. 2024, 4-6 p.m. in lecture hall 5901.EG.051

For questions regarding this exercise sheet, please contact: `du216@ic.ac.uk`
For general questions, please contact: `course.aim-lab@med.tum.de`

1. (70% points) **Differentially private model training**

   Differentially private algorithm is the one which is approximately invariant to an inclusion or an exclusion of a single individual. In simple terms: the output of the algorithm (in our case model) does not change if you include/exclude a single data point from the training pool. This is achieved through an addition of carefully calibrated noise to the output of the algorithm. For many ML applications, the DP algorithm of choice is the differentially private stochastic gradient descent (DP-SGD). Here the noise is added to the gradients of the model after the backward propagation is performed. Since in deep learning, it is often impossible to calculate the magnitude of DP noise **exactly**, a technique called *clipping* is introduced. The gradient values exceeding a specific clipping threshold are *clipped* i.e. they are capped at the value of this threshold.

   (a) (30% points) **Differentially private image classification**

   In this task you are expected to implement a simple DP image classification pipeline. Particularly, you should take a simple off-the-shelf model (e.g. ResNet-18) and train it on CIFAR-10 dataset (or any other benchmark dataset of your choosing).

   We suggest you use one of the pre-existing frameworks for DP training such as Opacus (most active, very straightforward to use) or deepee (less up-to-date, but has a neat tutorial attached) to simplify your implementation. You are free to choose hyperparameters for your training setting, but we recommend choosing the same ones as those proposed in the tutorial.

   After you implement a simple DP pipeline and train the model, you need to repeat the process without DP (i.e. train the model normally) and compare the results.

   The only deliverables expected for this part are a short (up-to half a page) description of how you train the model and a table of results including (for both settings):

   - Accuracy of the model
   - Value of ε for that accuracy (or inf for a non-private case)
   - Time to train the model

Important: make sure your code is well-written and well-documented, as you will use it for the next two tasks.

(b) (30% points) **Applying DP-SGD to medical data**

After you have completed your simple pipeline, you need to repeat your experiments on medical data. The expectation is that you analyse at least 3 separate training tasks (e.g. chest X-ray, dermatology datasets etc.)

You are not restricted to classification only and a diversity of training settings can score extra marks.

Do not change your setup from task one: use the same model and parameters (**exception**: if you need to adapt the last layers for a new task, this is acceptable, but reflect this in the table).

We recommend you stick to datasets from the MedMNIST collection, but you are free to choose your own dataset provided a) it is of clinical relevance and b) you clearly describe it (with relevant images/samples).

The deliverables for this section are: table identical to task one; short (up-to half a page) report on which learning tasks resulted in better performance than the rest.

(c) (40% points) **Analysing the parameters of private learning**

So far you have used multiple sets of identical hyperparameters. However, ε does not exist in a vacuum: magnitude of noise matters and can result in stronger or weaker privacy guarantees.

In this section, you need to perform a comparative study of the parameters which contribute to DP training process. You are expected to conduct a number of experiments on the datasets you have previously worked with and this time you need to alter the parameters you use.

You are free to experiment with different architectures, batch sizes, etc. The main requirement in this section is that you perform an **extensive** evaluation across multiple privacy settings. While you are free to choose your own "privacy levels", the recommended defaults (for image classification tasks) are 1, 4 and 8 (corresponding to "high", "medium" and "low" privacy levels).

The deliverables for this section include: figures showing how different parameters can affect the utility of the model (and how they change the privacy level if at all); a table supporting these results (similar to tasks one and two) and a written report (up-to a page) discussing which parameters were most important and why.

There is no minimal (or maximal) number of experiments for this section, provided your results can clearly show which factors affect the privacy-utility trade-off in your training settings.

2. (30% points) **Threats of federated learning**

Federated learning is a design paradigm, which allows a federation of clients to train a model collaboratively while preserving governance over their data. In simple terms: they train the model locally and only share the resulting model gradients, preventing other parties from being able to see the training data directly.

(a) (70% points) **Implementing federated learning Chest X-ray classification**

In this task you are required to design and implement a simple federated learning setting for an image classification task.

We recommend one of the two options with regards to the dataset:

1. MedMNIST chest X-ray classification OR
2. Paediatric pneumonia prediction

Your system should adhere to the following structure:

- Have between 3 and 10 participants (3 is still enough for full marks)
- Be synchronous (i.e. each successful update gets integrated each round)
- The data should be IID distributed (i.e. the data should be balanced across all participants)

You are free to utilise any open-source framework for FL. Some recommendations from our side include Flower (easiest to run with the best community support) and FLSim (can be configured in-depth, but much more involved). N.B. If you are using a machine with Apple silicon (M1 or M2) and want to use Flower, check out the following thread on potential grcp problems and troubleshooting.

The expected deliverables for this section are a) a short (up-to a page in total) report on how metrics (such as accuracy, loss, etc.) change at each individual client compared to the global model and b) how adjustment of individual components (e.g. replacing LeNet with ResNet-18, adding more clients etc.) affects the global performance.

You are assessed on the quality and the depth of your report (e.g. how well have you considered individual components of the system and how well have you evaluated them). You are **not** assessed on: having the perfect accuracy, quality of your engineering etc. Full marks can be scored if you use a single dataset and alter other learning parameters e.g. batch size, optimizers, architectures etc.

(b) (30% points) **Adversarial samples in federated learning**

Now imagine that one of the participants is an adversary. Their main goal is to subvert the training process and destroy the utility of your model.

One simple way of achieving this is model poisoning attacks, which perturb the training data in such a way that the model stops learning from them correctly. Poisoning can be performed using a large variety of techniques ranging from incorrect labelling to adding meaningless pixels to the image.

Your task is to replace the data (either partially or all of it) of a single client from the previous section and observe how the model performs under these circumstances.

For poisoning, you are free to pick any approach (e.g. even mislabelled data would suffice), but here is a link to the repository with more sophisticated (and often undetectable) methods torchattacks.

The deliverables for this section are a) a short (half a page) report on your findings, particularly emphasising what happens to performance of the global model over time and b) a detailed description (ideally in a table) of how you perform poisoning and which parameters you choose. By parameters we mean e.g. percentage of samples poisoned, perturbation technique used etc.

Your grade is determined by the quality of your report and not by how sophisticated the attack you chose is. Full marks can be scored even if you extensively evaluate single-pixel perturbations or mislabelling alone.