**Task 1 a)**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | DP | No DP |
| **CIFAR-10** | Acc | 58.06 | 72.08 |
| ε | 49.99 | inf |
| Time | 28201.90 s | 385.38 s |
| **PathMNIST** | Acc |  |  |
| ε |  |  |
| Time |  |  |
| **ChestMNIST** | Acc |  |  |
| ε |  |  |
| Time |  |  |
| **DermaMNIST** | Acc |  |  |
| ε |  |  |
| Time |  |  |
| **RetinaMNIST** | Acc |  |  |
| ε |  |  |
| Time |  |  |
| **BreastMNIST** | Acc |  |  |
| ε |  |  |
| Time |  |  |

The code provided implements a robust training pipeline for a neural network model using PyTorch, with the added ability to apply Differential Privacy (DP) for improved privacy preserving mechanisms. The process starts with the definition of data transformations, such as tensor conversion and normalization. Key parameters related to differential privacy, including `MAX\_GRAD\_NORM`, `EPSILON` and `DELTA`, are set to control the level of privacy during training.

The `make\_private` function introduces differential privacy into the model, optimizer and data loader. This function uses a privacy engine (`privacy\_engine`) to ensure that the training process complies with privacy constraints.

The training function (`train`) iterates through the specified number of epochs, updates the model parameters using backpropagation and monitors the training progress, including the average loss, accuracy and privacy budget (ε). The training time is also tracked.

A test function (`test`) evaluates the model on a separate test data set and provides insights into the generalization performance.

The entire training pipeline is orchestrated by the "pipeline" function, which initializes the model and optimizer, corrects the model with ModuleValidator and trains two models (one with and one without DP).

The CIFAR-10 dataset is used for training and testing, with data loading configured via PyTorch's DataLoader. The ResNet18 model architecture is used for image classification with 10 output classes, and the RMSprop optimizer with a given learning rate (LR) is used for updating the model parameters during training.

Table : Accuracies, Privacy Budgets and Training Times for Task 1a) and b)

**Task 1 b)**

Which learning tasks resulted in better performance than the rest?