Emmanuel Taveras

Task 1

In Task 1 the project was set up so that each file handled one part of the work. The train.sh script acted like a manager, looping through the six experiments and calling train.py to run each one. Inside train.py the models were built using the separate files for LeNet, ResNet18, and the Vision Transformer, while dataset.py prepared the MNIST and CIFAR-10 images by resizing and normalizing them. During training the script used CrossEntropyLoss and the Adam optimizer with a learning rate of 0.001. Every five epochs it saved checkpoints, recorded results, and plotted training curves. All outputs were stored in the reports/task1 folder, with results in a CSV file, plots of the curves, and model weights in a checkpoints folder. The results showed clear differences between the models. LeNet was almost perfect on MNIST, reaching 99.8 percent training accuracy and 98.9 percent test accuracy. On CIFAR-10 it overfit badly, with training accuracy at 86.9 percent while test accuracy dropped to 57.5 percent. ResNet18 performed very strongly on both datasets, hitting 99.9 percent train and 99.4 percent test accuracy on MNIST, and 98.9 percent train and 76.8 percent test accuracy on CIFAR-10. It generalized much better than LeNet, although some overfitting was still present. The Vision Transformer struggled across the board. With its very large number of parameters, small datasets like MNIST and CIFAR-10 were not enough to train it. Its performance hovered around 22 percent accuracy, with no real learning happening. This was made worse by normalization and optimizer choices that did not match standard ViT training practices. The experiments showed that LeNet is best suited for simpler datasets like MNIST, ResNet18 performed strongly and handled CIFAR-10 better, while the Vision Transformer struggled because of its training configuration.

Task 2

In Task 2 we reused the same modular structure from Task 1. The train_task2.py script controlled training, logging, and plotting, while dataset.py was extended to include augmentation options like rotation and horizontal flip. Models were imported directly from their files, and results were automatically saved to CSVs, plots, and checkpoints through the train2.sh script. For LeNet on MNIST, rotation gave a small boost in accuracy (98.94 percent vs 98.76 percent) and a higher learning rate sped up convergence while still finishing strong at 98.47 percent. For ResNet18 on CIFAR-10, augmentation had little to no effect, and rotation slightly reduced accuracy. Optimizer choice mattered more, with Adam training more reliably than SGD, while batch size mainly affected the smoothness of curves without changing final accuracy. Overall, LeNet stayed near perfect on MNIST under all conditions, while ResNet18 consistently reached about 76 to 77 percent accuracy on CIFAR-10, showing that the dataset and model architecture had more influence than the training tweaks.