Homework 7 Report

Name: Gilberto Feliu
Student ID: 801257813
Homework Number: 7
GitHub Repository: https://github.com/gfeliu-rgb/Intro-to-ML-ECGR-4105

Problem 1A:
For this part, I built a convolutional neural network (CNN) to classify all 10 categories in CIFAR-10. The architecture used two convolutional layers, each followed by a Tanh activation and a MaxPool2d layer, and a fully connected section with one hidden layer containing 64 neurons. I trained the model for 300 epochs using SGD with a learning rate of 0.001, using normalized CIFAR-10 for both training and validation.

After training, the model produced:

Training time: 1710.6581 seconds

Final training loss: 0.5593

Evaluation accuracy: 0.7074

Compared to the fully connected network from Homework 6, which had a training time of 627 seconds, a loss of 0.497, and an accuracy of 0.4871, this CNN took longer but achieved far better accuracy. Although the loss is slightly higher, the accuracy jump shows that the CNN actually learned meaningful features rather than memorizing. Convolution layers extract spatial patterns that fully connected layers miss. As before, preprocessing was handled outside the training loop and all training was done on the GPU for speed. Overall, this CNN significantly improved performance compared to the earlier fully connected model.

Problem 1B:
Here, I extended the CNN from Problem 1A by adding a third convolutional layer, again paired with Tanh and MaxPool2d. Everything else in the setup, including training configuration and preprocessing, remained the same. The model was trained for 300 epochs using SGD at a 0.001 learning rate on normalized CIFAR-10.

The results were:

Training time: 1944.0797 seconds

Final training loss: 0.5673

Evaluation accuracy: 0.7333

This deeper network performed slightly better than the original CNN from 1A, improving accuracy from 0.7074 → 0.7333, meaning the added layer helped capture more detailed spatial information. When compared to Homework 6-3B's fully connected network (training time 1167.7 sec, loss 0.0151, accuracy 0.4552), the extended CNN clearly outperforms it, even though training takes longer. The higher loss paired with higher accuracy again suggests good generalization. As before, the model ran on the GPU and preprocessing happened outside the loop. Overall, increasing network depth boosted accuracy beyond both the fully connected network and the original 2-layer CNN.

Problem 2A:
For this problem, I implemented a ResNet-10 style CNN with skip connections to classify CIFAR-10. The design included three convolutional layers with a residual link and two fully connected layers. Training used normalized CIFAR-10, 300 epochs, SGD at 0.001, and the GPU.

Performance metrics were:

Training time: 2188.1524 seconds

Final training loss: 0.6781

Evaluation accuracy: 0.6853

Compared to the deeper CNN from Problem 1B (accuracy 0.7333), the ResNet-10 model trained slower and reached lower accuracy. Although it introduced more complexity through residual connections, in this case it did not outperform the extended CNN.

Problem 2B:
In this part, I trained three modified versions of ResNet-10 to test different regularization strategies: weight decay (λ = 0.001), dropout (p = 0.3), and batch normalization. Each variant was trained for 300 epochs using SGD at 0.001 on normalized CIFAR-10. The training time, loss, and evaluation accuracy were recorded.

Here are the results:

ResNet-10 + Weight Decay

Training time: 2230.95 seconds

Final loss: 0.7093

Accuracy: 0.7079

ResNet-10 + Dropout (p = 0.3)

Training time: not explicitly stated (same setup)

Final loss: 1.0689

Accuracy: 0.6555
This was the worst performer—dropout reduced network capacity too aggressively.

ResNet-10 + Batch Normalization

Training time: 3162.85 seconds

Final loss: 0.5698

Accuracy: 0.6912

Relative to the baseline CNN from Problem 1A, all ResNet variations trained longer because of the extra architectural complexity. Only the weight-decay version exceeded the 1A CNN accuracy, while dropout harmed accuracy. Batch normalization stabilized training, gave the best loss, and produced accuracy that was close to both the weight-decay model and the Problem-1A CNN.

Overall, weight decay provided the most reliable improvement, dropout was too restrictive, and batch normalization improved stability but not accuracy.