Project 3 Report:

Table 1: MNIST Results

Architecture	Learning Rate	Batch Size	Optimizer	Dropout	Accuracy (%)	Runtime (Min)
MLP - Shallow	0.001	32	Adam	0	97.61	1.3
MLP - Medium	0.001	128	Adam	0.2	97.87	1.03
MLP - Deep	0.001	128	Adam	0.5	97.87	1.05
CNN - Baseline	0.001	32	Adam	0.2	98.65	1.44
CNN - Enhanced	0.001	32	Adam	0.2	98.78	1.57
CNN - Deep	0.001	32	Adam	0.5	99.19	1.79

Test Accuracy on Best MNIST Model: 99.31%

Table 2: CIFAR-10 Results

Architecture	Learning Rate	Batch Size	Optimizer	Dropout	Accuracy (%)	Runtime (Min)
MLP - Shallow	0.001	64	Adam	0	46.34	1.16
MLP - Medium	0.001	128	Adam	0	49.12	1.09
MLP - Deep	0.001	64	Adam	0	48.14	1.22
CNN - Baseline	0.001	64	Adam	0	67.9	1.2
CNN - Enhanced	0.001	32	Adam	0.2	69.92	1.54
CNN - Deep	0.001	32	Adam	0.5	80	1.79

Test Accuracy on Best CIFAR-10 Model: 79.34%

Analysis

Epochs:

In my opinion, the number of epochs is just as crucial to the accuracy of the model as any of the hyperparameters we were instructed to tune. In my early experiments, the validation set accuracy follows a logistic-like curve when mapped to the number of epochs. Intuitively, it makes sense that increasing the number of times the training data goes through the model will increase the effect each datapoint has on it, and thus, the more familiar the model becomes with the dataset. Even with a poorly structured model, there can still be more accurate predictions if the number of epochs is increased. This does of course, cost time, and there is also a point of diminishing returns. In my experiments I limited the number of epochs in my to 10. However, I believe that we would get a slightly more accurate model if the number of epochs were set to 11. And a slightly more accurate model that would have 12 epochs, and so on. The question when implementing a neural network in practice is: how accurate is accurate enough?

Adam vs SGD

In my early experiments, even before an exhaustive search of the hyperparameter combinations, it was clear that SGD as an optimization method was inferior to Adam. In every single experiment, even when just testing to verify if the code actually functioned, Adam came out as the more accurate optimization method. This could be for a number of reasons. The most likely culprit is the fact that the number of epochs in my experiments was limited to 10. Perhaps if the number of epochs is high enough, then SGD would make for a better optimizer. Whatever the case may be, the results of my experiments suggests that Adam converges faster to the optimal solution than SGD. For my hyperparameter tuning, I elected to skip testing SGD as an optimizer entirely.

Learning Rate

I also picked one learning rate for every experiment as well. In my experiments, it turned out that 0.001 as a learning rate was optimal in all but a handful of trial runs. This was the middle option, as initially, I also tested 0.01 and 0.0001. My theory is that the learning rate 0.001 is in the Goldilocks zone as far as learning rates are concerned. Increasing the learning rate too much causes divergence, and I found that when the learning rate was 0.01, there was a tendency to overshoot the target, and it actually took longer for the higher learning rate to actually learn. Conversely, when the learning rate was too low, the model couldn’t learn fast enough. It was clear to me early on that 0.001 was the best learning rate, and so I didn’t bother with the other models.

Regarding the MNIST and CIFAR Accuracy

The MNIST dataset is a much more simple dataset than CIFAR-10. CIFAR has 32 * 32 *3 = 3072 datapoints per image to keep track of, as compared to 28 * 28 = 784 total datapoints for each image in MNIST. That’s almost four times as many input variables per image to take into account and measure the impact of. For this reason, its obvious why every model can have a higher accuracy on MNIST than CIFAR.

The convolutional neural networks are very powerful because of their ability to capture local features within an image at different levels of granularity (determined by pooling). This makes the CNN’s more powerful, and thus more apt to dealing with the complexities of the CIFAR dataset.

In the case of MNIST, the difference in accuracy between the MLP and the CNN architectures are quite small. This goes back to my earlier point about epochs. How accurate is accurate enough? For a simple dataset like MNIST, it may only take a simple MLP to get accurate enough predictions for your use. However, for more complex data, a convolutional neural network is the way to go.