Assignment 1 Report

Datasets/Goals:

UCI Adult Income -

This Dataset is built from a 1994 U.S. Census, in this case it is used to train a model to predict whether or not an individual earns more than $50k/year based on personal attributes like age, sex, and occupation.

CIFAR100 -

A more complex version of CIFAR10, a set containing 60,000 images with 10 classes consisting of both animals and vehicles with a simple goal of identifying each image as one belonging to a single one of these classes. CIFAR100 has 20 superclasses, each with 5 subclasses, for 100 total classes. The content is still mostly animals and vehicles, but now includes much more specific classes as well as a few inanimate objects.

PatchCamelyon -

Another large image classification dataset but a binary goal. PatchCamelyon contains images of histopathologic scans of lymph node sections and our model's goal is to determine if there is a presence of metastatic tissue or not.

Building Models:

With help from Google's Gemini, I was able to build the complete models comprised of:

config.py - Configuration file, where the dataset, model, epochs, batch size, and learning rate can all be changed easily

Data loaders cifar100_py, adult_data.py and patchcamelyon_data.py - where each dataset is loaded and any preprocessors would happen. Split into separate files to help with unnecessary loading impacting runtime

models.py - Where the MLP and CNN are defined

train_eval.py - Where the functions that train and validate the model are defined

main.py - The file that calls upon all the previous defined functions and initializes the select parameters

Experiments:

The files are run in the same order they were introduced, apart from the dataloader in which only one of the three is run based on the selected dataset. As previously mentioned, selecting the model and dataset must be manually edited in the config.py file. To keep this experiment consistent for a fair comparison between models, I will start with keeping it consistent at batch size=100, epochs=10, and learning rate=0.001.

Results:

All results are at batch size=32, epochs=10, and learning rate=0.001 unless otherwise stated

Dataset	Architecture	Training Accuracy	Final Accuracy	Training Time
UCI Adult Income	MLP	86.19%	85.69%	>1 minute
CIFAR100	MLP	20.09%	19.2%	2-3 minutes
CIFAR100	MLP (100 epochs, 1,000 batch size)	26.57%	23.09%	13 minutes
PatchCamelyon	MLP	100%	52.55%	>1 minute
UCI Adult Income	CNN	85.54%	86.26%	>1 minute
CIFAR100	CNN	37.98%	40.30%	3 minutes
CIFAR100	CNN (50 epochs, 1,000 batch size)	47.41%	46.12%	11 minutes
PatchCamelyon	CNN	51.30%	48.5%	>1 minute

Issues/Adjustments:

I had the most trouble with the data loaders in general. The adult data required some work with making it compatible with models, the big issue being that the adult data was strictly categorical and the other two sets were image sets. The cifar dataset had minimal issues, mostly just messing with data paths until I found a format that worked. The patchcamelyon was the most difficult, trying to download it off any github and have it be formatted correctly was something I struggled with significantly. The data loader was the most complex of the three, pinpointing any of the errors.

I decided to experiment with a larger task for the CIFAR100, running a 50 epoch 1,000 batch size configuration to compare the accuracies and rate of accuracy improvement when training time increases.

I was satisfied with the results for the UCI and CIFAR datasets however the PatchCamelyon results are strange. I tried messing with the epochs run (10-100) and batch size (10-10000) and the final accuracy never deviated from the range of 45-55%.. Finally I tried a 600 epoch configuration at batch sizes 100, 1000, 10000, and 100000 and it still never got higher

than 55%. I used the CNN model for all of these variations because of the better performance in the CIFAR100 dataset then used the MLP for a 600 epoch 1000 batch size configuration and when getting the same results I was convinced that the runtime was not the issue.

Takeaways:

For the UCI data set we can see little to no difference. Both models took extremely short amounts of time to complete and accuracy is close enough to say that there is no real advantage to one over the other. I don't find this surprising given how simple the UCI data set is.

For the CIFAR100 dataset, the MLP did significantly worse than the CNN despite both models taking roughly the same amount of time. This shows that a CNN model should be preferred when taking on a project with image recognition. Given that there are 100 classes, and thus a 1% of randomly guessing correctly, and an accuracy of ~20% and ~40% does show that the models are being trained correctly. The configuration with larger epochs and batch sizes show a better accuracy but looking at the accuracy over time we can see that the MLP plateaus at roughly 25% while the CNN reaches about 50%. The rate of improvement diminished much earlier and faster in the MLP than it did in the CNN, thus from the significantly higher accuracy and efficiency in larger configurations at no cost in time it is safe to say the CNN does significantly better than the MLP in this case of image processing.

For the PatchCamelyon dataset we have some anomalies. First the MLP had a 100% accuracy in training yet a final accuracy 52.5%. Going from perfection to a coin flip is a very big issue in image recognition, I believe that this is an example of overfitting on the training data. The CNN had an accuracy of about 50% in both train and test, which doesn't immediately point to the overfitting the MLP likely went through, but is still concerning given that this is a binary classification and the benchmark of a random guess is 50%. When I was doing the epoch adjustments for CNN there appeared to be the same issue of 100% train accuracy with no substantial validation accuracy improvement. I assume it ran into the same issue of overfitting and thus I concluded that my models are ineffective for this dataset.