UI CS:4420 Spring 25 Course Project: Deep Learning on FashionMNIST

Objective: The goal of this project is to develop hands-on experience with convolutional neural networks (CNNs) by training and evaluating deep learning architectures on the FashionMNIST dataset. You will explore custom-built deep learning models and assess performance using appropriate evaluation metrics.

Instruction:

- The used dataset is FashionMNIST (10-class clothing image classification dataset). Please refer to the tutorial for loading the dataset.
- Please use the same data preprocessing for the input data used in the tutorial.
- Please accomplish the following tasks for this project and compose the report based on your findings.
- Please combine your code and written analysis as a Jupyter Notebook (an ipynb file).
- The submission is due by 11:59 PM on 4/19/25.

Task 1: Implementing CNN from scratch [30 Points]

Please implement the following CNN architecture manually by using PyTorch:

Layer Type	Configuration
Input	28×28 Gray-Scale Image
Conv1	32 filters, 3×3, stride 1, padding 1
MaxPool	2×2, stride 2
Conv2	64 filters, 3×3, stride 1, padding 2
MaxPool	2×2, stride 2
Conv3	64 filters, 3×3, stride 1, padding 1
MaxPool	2×2, stride 2
Flatten	—
FC1	Fully Connected (Output Size: 256)
FC2	Fully Connected (Output Size: 128)
FC3	Fully Connected (Output Size: 10)
Softmax	—

In your report, please print and show the configuration of your implemented CNN for checking whether this model has been correctly implemented.

Task 2: Training and Testing CNN: Train/Val/Test Set Split [40 Points]

Given the train/test data in FashionMNIST, first, please train the model with no validation set and test the trained model in test data. Second, please train the model with a

validation set accounting for 10%, 20%. 30%, and 40% training data. During the training, please use the SGD algorithm as the training algorithm with a learning rate of 0.1 and cross entropy loss as the training loss. Please feel free to choose a relatively large training epoch number (at least 10) and a relatively large batch size (at least 32) and keep them the same for the rest of the project.

During testing, please report the performance with different number of validation data by using Accuracy as the evaluation metric. What are your findings in this experiment? Please write them down in your report with analysis.

Task 3: Training and Testing CNN: Learning Rate [40 Points]

Next, please explore the influence of the learning rate for the model performance. Based on the best train/val split ratio you find in the previous task, please change the learning rate in SGD from small to large (0.001, 0.01, 0.1, 1, 10) and test the model on the trained model with the same metrics used in Task 2.

Similar to Task 2, please share your findings in this experiment in your report based on the obtained results with analysis.

Task 4: Training and Testing CNN: Training Algorithm [20 Points]

Based on the best learning rate you find in the previous experiment, please change the training algorithm and use the Adam algorithm for training with the same learning rate and training setting. Will the change of training algorithm improve the performance? Please discuss your findings with detailed results in your report.

Task 5: Testing CNN: Evaluation Metric [20 Points]

Based on the best model you get from previous training, please consider the following one-class setting for testing and use a new metric, AUC (area under the receiver operating characteristic curve), for evaluating the performance of the trained model. Suppose we treat the images with label "2" as the positive images and all other images with labels different from "2" as the negative images. Please modify your codes for testing and compute the corresponding AUC for the performance of the model.