ECSE440L Deep Learning Lab 7:
Programming Assignment 5: Make your own Convolutional Neural Network

Dataset

Use the following line of code to load dataset containing four files:
from keras.datasets import fashion_mnist

1. train_images
◦ 60,000 samples of 28 x 28 grayscale image. The data is of size 60000 x 784. Each pixel has a single intensity value which is an integer between 0 and 255.
2. train_labels
◦ 60,000 samples of 10 classes for the images in the given train_images. Class details are mentioned below.
3. test_images
◦ 10,000 samples of 28 x 28 grayscale image. The data is of size 10000 x 784. Each pixel has a single intensity value which is an integer between 0 and 255.
4. test_labels
◦ 10,000 samples from 10 classes for the images in the given test_images.

Class details are listed below.

Class labels
Each training and test samples are assigned to one of the following labels:

Class	**Type of dress**
0	T-shirt / top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

Your Task: Building Convolution Neural Network to classify clothes

1. PLEASE PREPARE A REPORT ACCOMPANYING YOUR SUBMITTED CODES.
2. Build a Convolution Neural network with the following specification:
   - ✓ conv1: Convolution layer having 2feature detectors, with kernel size 3 x 3, and **sigmoid** as
   - ✓ the activation function, with stride 1 and no-padding.
   - ✓ Pool1: A max-pooling layer with pool size 2x2.
   - ✓ conv2: Convolution layer having 2 feature detectors, with kernel size 3 x 3, and **rectified**
   - ✓ **linear unit** as the activation function, with stride 1 and no-padding.
   - ✓ Pool2: A max-pooling layer with pool size 2x2.
   - ✓ FC1: Fully connected layer with 50 neurons, and **hyperbolic tangent** as the activation

- ✓ function.
- ✓ Output: Output layer containing 10 neurons, **softmax** as activation function.

3. Print the model architecture (both summary and plot) in your report.
4. Split the given training dataset into 80% training & 20% test (a.k.a., validation set).
5. Build your CNN training algorithm in such a way that it saves the weights (i.e., feature detectors) in every single epoch so that you can retrieve the best set of feature detectors when the experiment is over.
6. Fit the model above with the training dataset, with 20 epochs, minibatch size of 200. Then, sit back and relax while the experiment is running.
7. Using the fit history data, print Epoch-loss, epoch-accuracy plot for training and validation. (Note: validation data is not test data). And, with the same data, print an Epoch-accuracy plot for training and validation.
8. From the plot/history of every epoch, determine the best model parameters (i.e., the weights (kernels, filters, feature detectors). Evaluate that model on the given test dataset, and
    - ✓ print classification (base) error and accuracy in your report.
    - ✓ Print the classification report.
    - ✓ Print the confusion matrix, and also plot the confusion matrix which kind of look like
    - ✓ heatmap.