Build a Simple NN on Fashion-MNIST

1. Dataset Overview

The Fashion-MNIST dataset is a popular image dataset used for benchmarking machine learning algorithms. It contains grayscale images of clothing items belonging to 10 different categories such as T-shirt, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, and Ankle boot.

2. How to Access the Dataset

The dataset is available directly in PyTorch's torchvision library. Students should use the torchvision.datasets module to download and load for both training and testing. Each image is 28×28 pixels, single-channel (grayscale), and each label corresponds to one of the 10 clothing categories.

3. Task Description

Your task is to build a simple fully connected neural network (Multilayer Perceptron) to classify images from the Fashion-MNIST dataset. The goal is to correctly predict the type of clothing item shown in each image and achieve a test **accuracy higher than 85%.**

4. Model Architecture Requirements

The neural network should include the following layers:

- Input layer: 28×28 flattened into 784 features.
- Hidden Layer 1: 256 neurons, activation function ReLU.
- Hidden Layer 2: 128 neurons, activation function ReLU.
- Output Layer: 10 neurons (one for each class).

Use CrossEntropyLoss as the loss function and Adam optimizer with a learning rate of 0.001.

5. Training Setup

Train the network using:

- Batch size: 64
- Epochs: between 5 and 10
- Input normalization: scale pixel values between 0 and 1

Use the training set for model learning and the test set for evaluation.

6. Evaluation and Submission

After training your model, report the following results:

- Final training and test accuracy.
- A confusion matrix for the test set.
- Example images showing model predictions (correct and incorrect).

Submit your results as a GitHub repo containing Jupyter Notebook (.ipynb) through the given form including:

1. Code for loading data, building the model, training, and evaluation.
2. Visualizations of loss and accuracy per epoch.
3. A short conclusion paragraph summarizing model performance.