MLOPs Assignment-1: Comparative Analysis of Deep Learning and Classical Models on MNIST and FashionMNIST

Name: Gautam Kumar kushwaha
Roll No: M25CSA037Course:MLOPS
Assignment: 1

Introduction:
This assignment explores the performance of deep learning and classical machine learning models on MNIST and FashionMNIST datasets. The objective is to analyze the impact of architectural depth, optimization strategies, and compute environments (CPU vs GPU) on classification accuracy, training time, and computational complexity.

Datasets

1. MNIST, FashionMNIST
2. Image size: 28×28 grayscale
3. Split: **70% train – 10% val – 20% test**

Models

1. ResNet-18 (pretrained = False)
2. ResNet-50 (pretrained = False)
3. SVM (poly, rbf kernels)

Training Details

1. Framework: PyTorch
2. Optimizers: SGD, Adam
3. Learning rates: 0.001, 0.0001
4. Batch sizes: 16, 32
5. Epochs: mention two values you tried
6. USE_AMP = True
7. pin_memory: True/False

Q1(a): Deep Learning Results

Tables: see in github readme.md file
https://github.com/gautamkushwaha/MLOps-Gautam_m25csa037/tree/Assignment_1

For MNIST, ResNet-18 consistently outperformed ResNet-50 across most hyperparameter settings, achieving a peak accuracy of 99.30%. The marginal improvement of ResNet-50 does not justify its increased computational cost. On FashionMNIST, accuracy drops across all configurations due to higher visual complexity, yet ResNet-18 remains more efficient and stable.

Q1(b): SVM Experiments
While SVMs train significantly faster and require fewer resources, they lag behind deep learning models in classification accuracy. This gap is more pronounced in MNIST, where CNN-based feature learning provides a strong advantage.

Q2: CPU vs GPU Analysis:
GPU acceleration significantly benefits deeper architectures such as ResNet-50, reducing training time by more than 50%. However, for lightweight models like ResNet-18, CPU training remains competitive, suggesting that model complexity should guide hardware selection

Best Model Discussion:
Across all experiments, ResNet-18 with Adam optimizer and a learning rate of 0.001 emerged as the best-performing configuration. It consistently achieved high accuracy while maintaining low training time and computational cost, making it the most practical choice for small-scale image classification tasks.

Learning Outcomes:

1. Learned how to structure reproducible ML experiments
2. Understood trade-offs between model depth and compute cost
3. Gained hands-on experience with GitHub branching and GitHub Pages
4. Learned to document experiments for reproducibility and evaluation