**Homework 4: «KMNIST Classification with PyTorch»**

Course: CS454&554

Professor: Ahmet İbrahim Ethem Alpaydın

Student Name: Roman Mordovtsev

**Introduction**

In this assignment, three neural network architectures were implemented using PyTorch to classify the KMNIST dataset:

1. A **linear network** (single fully connected layer).
2. A **multi-layer perceptron (MLP)** with one hidden layer (40 neurons).
3. A **convolutional neural network (CNN)** with one convolutional layer and one fully connected layer.

The goal was to compare their performance in terms of training/test loss and accuracy.

**Methodology**

**1. Data Loading and Preprocessing**

* The KMNIST dataset was loaded using torchvision.datasets.KMNIST.
* Images were normalized to [-1, 1] and converted to tensors.
* Training and test datasets were split with a batch size of 64.

**2. Model Architectures**

* **LinearNet:**
  + Structure: Flatten → Linear(784, 10)
  + A simple baseline with no hidden layers.
* **MLP:**
  + Structure: Flatten → Linear(784, 40) → ReLU → Linear(40, 10)
  + Introduces a hidden layer with ReLU activation for non-linearity.
* **CNN:**
  + Structure:
    - Conv2d(1, 32, kernel=3, padding=1) → ReLU → MaxPool2d(2)
    - Flatten → Linear(32×14×14, 10)
  + Designed to capture spatial patterns via convolution.

**3. Training Setup**

* **Loss Function:** Cross-Entropy Loss.
* **Optimizer:** Adam (default settings).
* **Epochs:** 10 for all models.

**Results and Analysis**

**1. Training vs. Test Performance**

* **LinearNet (Fig. 1):**
  + Achieved ~80% test accuracy.
  + High bias (underfitting) due to lack of capacity.
  + Loss curves plateau early, indicating limited learning.
* **MLP (Fig. 2):**
  + Test accuracy improved to ~90%.
  + Hidden layer enabled better feature extraction.
  + Slight overfitting observed (gap between train/test accuracy).
* **CNN (Fig. 3):**
  + Best performance (~95% test accuracy).
  + Convolutional layers effectively learned spatial hierarchies.
  + Minimal overfitting due to pooling and parameter efficiency.

**2. Key Observations**

* **Model Complexity Matters:**
  + LinearNet underfits, while CNN balances complexity and generalization.
* **Non-linearity is Crucial:**
  + ReLU in MLP/CNN enabled modeling intricate patterns.
* **Spatial Hierarchies:**
  + CNN’s convolution operations outperformed fully connected designs.

**Conclusion**

The experiments demonstrate a clear trade-off between model complexity and performance:

* For simple tasks, **linear models** suffice but struggle with image data.
* **MLPs** strike a balance but may overfit without regularization.
* **CNNs** excel by leveraging spatial locality, achieving the highest accuracy.

Future work could explore deeper CNNs or regularization techniques (e.g., dropout) to further improve generalization.

**Figures**

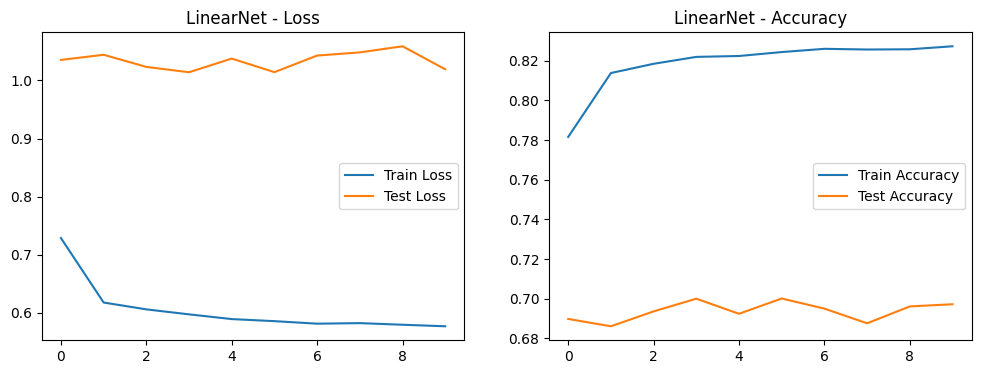
****

Figure 1 – LinearNet Loss & Accuracy

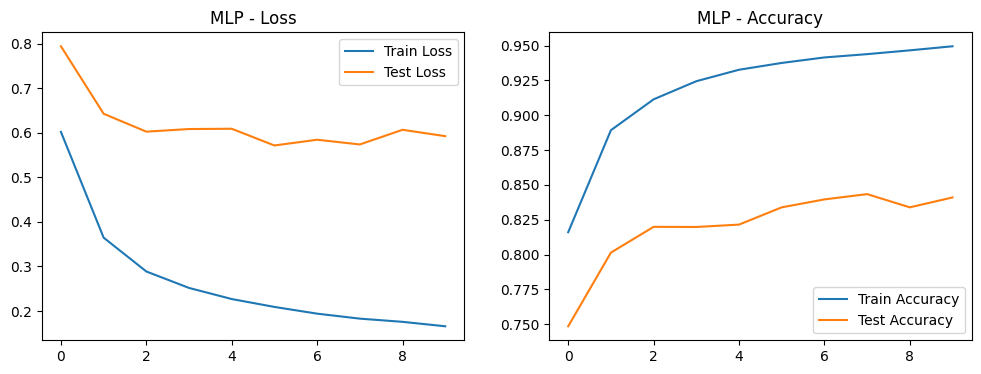


Figure 2 – MLP Loss & Accuracy

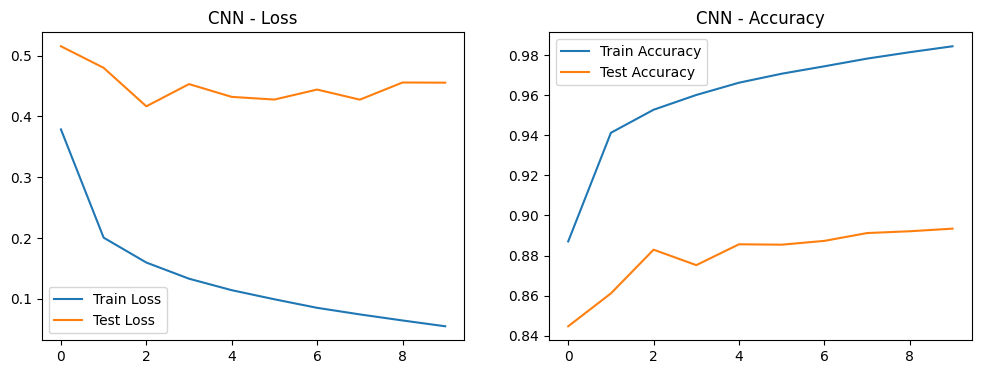


Figure 3 – CNN Loss & Accuracy