MSML603(Machine Learning)

PROJECT 2 REPORT

-Ayush Sinha LNU

-121334060, PCS06

1.) Feedforward Neural Network: -

I. Results

The Feedforward Neural Network (FFNN) was trained and evaluated on the MNIST dataset over five runs, with training across 10 epochs per run. Below are the key results:

- Testing Accuracies Across Runs:
  - Run 1: 98.1%
  - Run 2: 98.11%
  - Run 3: 97.87%
  - Run 4: 98.04%
  - Run 5: 98.04%
- Average Testing Accuracy: 98.03%

II. Lessons Learned

- Hidden Layers
  - Including three hidden layers with neuron counts of 256, 128 and 64 provided sufficient capacity for learning while maintaining efficiency.
  - The choice of ReLu activation facilitated fast training and avoided vanishing gradients.
- Regularizations
  - Adding dropout layers between dense layers significantly improved generalization and prevented overfitting.
- Input Layer

- Flattening the input images (28x28) into a 784-dimensional vector allowed for seamless integration into feedforward structure.
- Learning Rate
  - Using the Adam optimizer with a learning rate of 0.001 resulted in stable and consistent convergence across all runs.
- Epochs
  - Training the network for 10 epochs consistently led to testing accuracy above 95%.
- Impact of Regularization
  - Dropout layers with rates of 0.3 and 0.5 were critical in preventing overfitting, especially with 10 epochs of training.

2.) <u>Convolutional Neural Network</u>: -

I. Results

The Convolutional Neural Network (CNN) was trained and evaluated on the MNIST dataset over five runs, with training across 10 epochs per run. Below are the results:

- Testing Accuracies Across Runs:
  - Run 1: 99.1%
  - Run 2: 99.23%
  - Run 3: 99.2%
  - Run 4: 99.1%
  - Run 5: 99.26%
- Average Test Accuracy: 99.2%

II. Lessons Learned

- Convolutional Layers
  - The use of two convolutional layers allowed the network to effectively extract both low-level (edges and textures) and high-level (shapes and patterns) features.
  - Filters of size 3x3 with increasing numbers of filters (32 and 64) worked well to capture hierarchical features from MNIST dataset.

- Pooling Layers
    - Max pooling layers reduced spatial dimensions, making the computation more efficient while preserving key features.
    - Using a pooling size of 2x2 was sufficient for dimensionality reduction.
- Fully Connected Layers
    - A dense layer with 128 neurons enabled the network to integrate extracted features and performance accurate classification.
    - Dropout after the dense layer helped prevent overfitting.
- Regularization
    - Dropout rates of 0.25 and 0.5 proved highly effective in preventing during training over 10 epochs.
- Training Stability
    - The choice of optimizer and learning rate resulted in stable training across all runs, with minimal variance in testing accuracy.