14-8-2025

EXP-4

MNIST Digit Classification Using PyTorch

Aim :-

To implement a simple feedforward neural network using PyTorch to classify handwritten digits from the MNIST dataset

Objectives

1. Load and preprocess the MNIST dataset using torchvision
2. Define a simple feedforward neural network architecture with one hidden layer.
3. Train the model using stochastic Gradient Descent(SGD) and CrossEntropyLoss.
4. Evaluate the model on the test dataset to compute accuracy
5. Understand the workflow of building and training a basic neural network in PyTorch

Pseudo Code

```
Begin
    Import torch, torch.nn, torch.optim,
        torchvision.dataset, torchvision.tranforms.

Define NeuralNetwork class

    INIT:
        Layer1 = Liner(28*28 => 128
```

Layer 2 = Linear(128 → 64)

Outputlayer = Linear(64→10)

Activation = ReLU

Forward pass:

Flatten input image

Pass through Layer 1, apply Relu

Pass through Layer 2, apply ReLU

Pass through Outputlayer

Return output (logits)

LOAD MNIST dataset with transforms:

- Convert to tensor
- Normalise pixel values

Create Data Loaders for training and testing

INITIALIZE model, loss function (CrossEntropyLoss), optimizer (SGD)

For each epoch in range (num_epochs):

SET model to training mode

For each batch in traing data:

Zero gradients

Foward pass → get predictions

Update weights using optimizer

print average loss for the epoch

Set model to evaluation mode

For each batch in test data:

Forward pass

Compare predictions with Labels

Count correct predictions

Compute accuracy = (correct / total) * 100

print accuracy

END

Observation

Accuracy: 74.73%.

Result:

The code has been executed successfully.