EX.NO:04
14.08.25

BUILD A SIMPLE FEED FORWARD NEURAL NETWORK TO RECOGNIZE HANDWRITTEN CHARACTER

Aim:

To build a simple feed forward neural network to recognize Handwritten character.

Objective:

1. To load and preprocess the MNIST dataset for neural network input
2. To build feed forward neural network with hidden layers
3. To train the model using stochastic gradient descent optimizer and sparse categorical cross-entropy loss.
4. Evaluate the trained model on test data and measure its accuracy.
5. To predict the class of given handwritten image.

Pseudocode

START

Load MNIST dataset (training and testing data)

Flatten each image from 28 x 28 to 784 features

Normalize pixel values to range [0,1]

Create a sequential neural network:

Layer 1 : Dense (128 neurons, Relu activation)

Layer 2 : Dense (64 neurons, Relu activation)

Output layer : Dense (10 neurons, softmax

Eg Training

Epoch	Accuracy	Loss
1	0.9929	0.0232
2	0.9968	0.0128
3	0.9976	0.0099
4	0.9976	0.0088
5	0.9987	0.0058

Training

Epoch	Loss
1	1.0594
2	0.3860
3	0.3290
4	0.2974
5	0.2726

Test Accuracy : 92.82 %

activation).

Compile model

optimizer = stochastic gradient descent

loss = sparse categorical crossentropy

Metric = accuracy

Train model on training data for 5 epochs

Evaluate model on testing data

print test accuracy

END

observation

The loss decreases with each epoch, showing that the model is learning.

Accuracy improves steadily during training

Testing

overall accuracy of the model on the entire dataset = ~~0.9828~~ 92.82%

Results

successfully build a simple feed forward neural network to recognize handwriten character.

L4.2.ipynb (4) - JupyterLab | IndentationError fix | What is PyTorch? | Data Science | difference between tensorflow

Not secure 10.1.38.19/user/ra2311047010011/lab/tree/DEEP%20LEARNING/L4.2.ipynb

File Edit View Run Kernel Tabs Settings Help

Filter files by name

/ DEEP LEARNING /

Name	Last Modified
data	7 days ago
Breast_canc...	14 days ago
Exp2.ipynb	14 days ago
installpytor...	28 minutes ago
L_3.ipynb	7 days ago
L1.1.ipynb	14 days ago
L1.ipynb	14 days ago
L2.ipynb	7 days ago
L3.ipynb	7 days ago
L4.2.ipynb	47 seconds ago
L4.ipynb	28 minutes ago
Untitled.ipy...	14 days ago

L4.ipynb | installpytorch.ipynb | L4.2.ipynb

Code | Notebook | Python 3 (ipykernel)

[37]:
```
import torch
import torch.nn as nn
import torch.optim as optim
import torchvision
from torchvision import transforms
from torch.utils.data import DataLoader
```

[38]:
```
transform = transforms.Compose([
    transforms.ToTensor(),
    transforms.Normalize((0.5,), (0.5,))
])
```

[39]:
```
train_dataset = torchvision.datasets.MNIST(
    root='./data', train=True, transform=transform, download=True
)
test_dataset = torchvision.datasets.MNIST(
    root='./data', train=False, transform=transform, download=True
)
```

[40]:
```
train_loader = DataLoader(train_dataset, batch_size=64, shuffle=True)
test_loader  = DataLoader(test_dataset, batch_size=64, shuffle=False)
```

[41]:
```
import torch.nn as nn
class SimpleFFN(nn.Module):
    def __init__(self):
        super(SimpleFFN, self).__init__()
        self.flatten = nn.Flatten()
        self.fc1 = nn.Linear(28*28, 128)
        self.fc2 = nn.Linear(128, 64)
        self.fc3 = nn.Linear(64, 10)
        self.relu = nn.ReLU()
```

Simple 0 3 Python 3 (ipykernel) | Busy Mem: 1.31 GB Mode: Command Ln 1, Col 1 L4.2.ipynb 0

ENG INTL 09:14 22-08-2025

L4.2.ipynb (4) - JupyterLab | IndentationError fix | What is PyTorch? | Data Science | difference between tensorflow

Not secure 10.1.38.19/user/ra2311047010011/lab/tree/DEEP%20LEARNING/L4.2.ipynb

File Edit View Run Kernel Tabs Settings Help

Filter files by name

/ DEEP LEARNING /

Name	Last Modified
data	7 days ago
Breast_canc...	14 days ago
Exp2.ipynb	14 days ago
installpytor...	30 minutes ago
L_3.ipynb	7 days ago
L1.1.ipynb	14 days ago
L1.ipynb	14 days ago
L2.ipynb	7 days ago
L3.ipynb	7 days ago
L4.2.ipynb	2 minutes ago
L4.ipynb	30 minutes ago
Untitled.ipy...	14 days ago

L4.ipynb | installpytorch.ipynb | L4.2.ipynb

Code | Notebook | Python 3 (ipykernel)

```
        self.fc3 = nn.Linear(64, 10)
        self.relu = nn.ReLU()

    def forward(self, x):
        x = self.flatten(x)
        x = self.relu(self.fc1(x))
        x = self.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

[42]:
```
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = SimpleFFN().to(device)
```

[43]:
```
criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD(model.parameters(), lr=0.01)
epochs = 5
```

[44]:
```
for epoch in range(epochs):
    running_loss = 0.0
    for images, labels in train_loader:
        images, labels = images.to(device), labels.to(device)

        outputs = model(images)
        loss = criterion(outputs, labels)

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        running_loss += loss.item()
```

Simple 0 3 Python 3 (ipykernel) | Idle Mem: 1.31 GB Mode: Command Ln 1, Col 1 L4.2.ipynb 0

ENG INTL 09:16 22-08-2025

L4.2.ipynb (4) - JupyterLab | IndentationError fix | What is PyTorch? | Data Scienc | difference between tensorflow

Not secure 10.1.38.19/user/ra2311047010011/lab/tree/DEEP%20LEARNING/L4.2.ipynb

File Edit View Run Kernel Tabs Settings Help

L4.ipynb | installpytorch.ipynb | L4.2.ipynb

Code | Notebook | Python 3 (ipykernel)

Filter files by name

/ DEEP LEARNING /

Name	Last Modified
data	7 days ago
Breast_canc...	14 days ago
Exp2.ipynb	14 days ago
installpytor...	30 minutes ago
L_3.ipynb	7 days ago
L1.1.ipynb	14 days ago
L1.ipynb	14 days ago
L2.ipynb	7 days ago
L3.ipynb	7 days ago
L4.2.ipynb	49 seconds ago
L4.ipynb	30 minutes ago
Untitled.ipy...	14 days ago

```
        running_loss += loss.item()

    print(f"Epoch [{epoch+1}/{epochs}], Loss: {running_loss/len(train_loader):.4f}")
```

```
Epoch [1/5], Loss: 1.0231
Epoch [2/5], Loss: 0.3820
Epoch [3/5], Loss: 0.3246
Epoch [4/5], Loss: 0.2925
Epoch [5/5], Loss: 0.2678
```

[45]:
```
print("Training Finished")
```

```
Training Finished
```

[46]:
```
correct, total = 0, 0
with torch.no_grad():
    for images, labels in test_loader:
        images, labels = images.to(device), labels.to(device)
        outputs = model(images)
        _, predicted = torch.max(outputs, 1)
        total += labels.size(0)
        correct += (predicted == labels).sum().item()

print(f"Test Accuracy: {100 * correct / total:.2f}%")
```

```
Test Accuracy: 92.41%
```

[ ]:

[ ]:

[ ]:

Simple 0 3 Python 3 (ipykernel) | Idle Mem: 1.31 GB Mode: Command Ln 1, Col 1 L4.2.ipynb 0

ENG INTL 09:16 22-08-2025