Experiment 13

Problem Statement:

Write complete code Using N'NIST dataset, Improve autoencoder's performance using convolutional layers.

GitHub & Google Colab Link:

GitHub Link: https://github.com/piyush-gambhir/ncu-lab-manual-and-end-semester-projects/blob/main/NCU-CSL312%20-%20DL%20-%20Lab%20Manual/Experiment%2013/Experiment%2013.ipynb

Google Colab Link:

Installing Dependencies:

In [ ]:
```
! pip install tabulate numpy pandas matplotlib seaborn
```

```
Requirement already satisfied: tabulate in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packa
ges (0.9.0)
Requirement already satisfied: numpy in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packages
(1.26.4)
Requirement already satisfied: pandas in c:\users\mainp\appdata\local\programs\python\python311\lib\site-package
s (2.2.2)
Requirement already satisfied: matplotlib in c:\users\mainp\appdata\local\programs\python\python311\lib\site-pac
kages (3.8.4)
Requirement already satisfied: seaborn in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packag
es (0.13.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\mainp\appdata\local\programs\python\python311\
lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\mainp\appdata\local\programs\python\python311\lib\site-p
ackages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\mainp\appdata\local\programs\python\python311\lib\site
-packages (from pandas) (2024.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\mainp\appdata\local\programs\python\python311\lib\si
te-packages (from matplotlib) (1.2.1)
Requirement already satisfied: cycler>=0.10 in c:\users\mainp\appdata\local\programs\python\python311\lib\site-p
ackages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\mainp\appdata\local\programs\python\python311\lib\s
ite-packages (from matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\mainp\appdata\local\programs\python\python311\lib\s
ite-packages (from matplotlib) (1.4.5)
Requirement already satisfied: packaging>=20.0 in c:\users\mainp\appdata\local\programs\python\python311\lib\sit
e-packages (from matplotlib) (24.0)
Requirement already satisfied: pillow>=8 in c:\users\mainp\appdata\local\programs\python\python311\lib\site-pack
ages (from matplotlib) (10.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\mainp\appdata\local\programs\python\python311\lib\si
te-packages (from matplotlib) (3.1.2)
Requirement already satisfied: six>=1.5 in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packa
ges (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

Code

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras import layers, models
```

In [ ]:
```
def build_conv_autoencoder():
    # Encoder
    input_img = layers.Input(shape=(28, 28, 1))  # 28x28 pixels with 1 channel
    x = layers.Conv2D(16, (3, 3), activation='relu', padding='same')(input_img)
    x = layers.MaxPooling2D((2, 2), padding='same')(x)
    x = layers.Conv2D(8, (3, 3), activation='relu', padding='same')(x)
    x = layers.MaxPooling2D((2, 2), padding='same')(x)
    x = layers.Conv2D(8, (3, 3), activation='relu', padding='same')(x)
    encoded = layers.MaxPooling2D((2, 2), padding='same')(x)  # down to 4x4x8

    # Decoder
    x = layers.Conv2D(8, (3, 3), activation='relu', padding='same')(encoded)
    x = layers.UpSampling2D((2, 2))(x)
    x = layers.Conv2D(8, (3, 3), activation='relu', padding='same')(x)
```

```
    x = layers.UpSampling2D((2, 2))(x)
    x = layers.Conv2D(16, (3, 3), activation='relu')(x)  # no padding
    x = layers.UpSampling2D((2, 2))(x)
    decoded = layers.Conv2D(1, (3, 3), activation='sigmoid', padding='same')(x)

    autoencoder = models.Model(input_img, decoded)
    autoencoder.compile(optimizer='adam', loss='binary_crossentropy')

    return autoencoder

# Load MNIST dataset
(x_train, _), (x_test, _) = tf.keras.datasets.mnist.load_data()

# Preprocessing
x_train = x_train.astype('float32') / 255.
x_test = x_test.astype('float32') / 255.
x_train = np.reshape(x_train, (len(x_train), 28, 28, 1))  # adapt this if using `channels_first` image data for
x_test = np.reshape(x_test, (len(x_test), 28, 28, 1))  # adapt this if using `channels_first` image data format

# Build and train the autoencoder
autoencoder = build_conv_autoencoder()
autoencoder.fit(x_train, x_train, epochs=50, batch_size=128, shuffle=True, validation_data=(x_test, x_test))

# Encode and decode some digits (visualization)
decoded_imgs = autoencoder.predict(x_test)

# Display the results
n = 10  # number of digits to display
plt.figure(figsize=(20, 4))
for i in range(n):
    # Display original
    ax = plt.subplot(2, n, i + 1)
    plt.imshow(x_test[i].reshape(28, 28))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)

    # Display reconstruction
    ax = plt.subplot(2, n, i + 1 + n)
    plt.imshow(decoded_imgs[i].reshape(28, 28))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)
plt.show()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 ━━━━━━━━━━━━━━━━━━━━ 2s 0us/step
Epoch 1/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 72s 86ms/step - loss: 0.3032 - val_loss: 0.1446
Epoch 2/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 53s 113ms/step - loss: 0.1405 - val_loss: 0.1270
Epoch 3/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 83s 113ms/step - loss: 0.1261 - val_loss: 0.1197
Epoch 4/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 81s 110ms/step - loss: 0.1189 - val_loss: 0.1146
Epoch 5/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 44s 91ms/step - loss: 0.1147 - val_loss: 0.1109
Epoch 6/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 80s 86ms/step - loss: 0.1116 - val_loss: 0.1086
Epoch 7/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 49s 105ms/step - loss: 0.1095 - val_loss: 0.1070
Epoch 8/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 40s 83ms/step - loss: 0.1079 - val_loss: 0.1056
Epoch 9/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 27s 53ms/step - loss: 0.1066 - val_loss: 0.1044
Epoch 10/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 18s 38ms/step - loss: 0.1055 - val_loss: 0.1036
Epoch 11/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 24s 50ms/step - loss: 0.1048 - val_loss: 0.1027
Epoch 12/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 28s 60ms/step - loss: 0.1036 - val_loss: 0.1017
Epoch 13/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 33s 71ms/step - loss: 0.1031 - val_loss: 0.1015
Epoch 14/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 37s 78ms/step - loss: 0.1023 - val_loss: 0.1018
Epoch 15/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 40s 84ms/step - loss: 0.1018 - val_loss: 0.0998
Epoch 16/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 41s 87ms/step - loss: 0.1010 - val_loss: 0.0994
Epoch 17/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 37s 79ms/step - loss: 0.1003 - val_loss: 0.0989
Epoch 18/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 36s 76ms/step - loss: 0.0999 - val_loss: 0.0984
Epoch 19/50
```

```
469/469 ━━━━━━━━━━━━━━━━━━━━ 39s 82ms/step - loss: 0.0995 - val_loss: 0.0978
Epoch 20/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 37s 79ms/step - loss: 0.0991 - val_loss: 0.0973
Epoch 21/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 35s 74ms/step - loss: 0.0987 - val_loss: 0.0970
Epoch 22/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 34s 72ms/step - loss: 0.0982 - val_loss: 0.0970
Epoch 23/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 35s 74ms/step - loss: 0.0979 - val_loss: 0.0966
Epoch 24/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 30s 64ms/step - loss: 0.0975 - val_loss: 0.0963
Epoch 25/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 31s 66ms/step - loss: 0.0968 - val_loss: 0.0955
Epoch 26/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 33s 70ms/step - loss: 0.0967 - val_loss: 0.0953
Epoch 27/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 34s 72ms/step - loss: 0.0965 - val_loss: 0.0949
Epoch 28/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 36s 76ms/step - loss: 0.0960 - val_loss: 0.0950
Epoch 29/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 39s 82ms/step - loss: 0.0957 - val_loss: 0.0944
Epoch 30/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 55s 110ms/step - loss: 0.0955 - val_loss: 0.0941
Epoch 31/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 35s 75ms/step - loss: 0.0953 - val_loss: 0.0936
Epoch 32/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 41s 74ms/step - loss: 0.0947 - val_loss: 0.0939
Epoch 33/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 35s 75ms/step - loss: 0.0944 - val_loss: 0.0937
Epoch 34/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 32s 67ms/step - loss: 0.0945 - val_loss: 0.0929
Epoch 35/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 45s 76ms/step - loss: 0.0939 - val_loss: 0.0933
Epoch 36/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 38s 68ms/step - loss: 0.0939 - val_loss: 0.0927
Epoch 37/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 23s 49ms/step - loss: 0.0937 - val_loss: 0.0922
Epoch 38/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 20s 43ms/step - loss: 0.0934 - val_loss: 0.0922
Epoch 39/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 32s 69ms/step - loss: 0.0934 - val_loss: 0.0920
Epoch 40/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 26s 56ms/step - loss: 0.0931 - val_loss: 0.0919
Epoch 41/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 17s 36ms/step - loss: 0.0929 - val_loss: 0.0917
Epoch 42/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 13s 28ms/step - loss: 0.0928 - val_loss: 0.0916
Epoch 43/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 13s 28ms/step - loss: 0.0924 - val_loss: 0.0913
Epoch 44/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 13s 27ms/step - loss: 0.0924 - val_loss: 0.0912
Epoch 45/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 13s 27ms/step - loss: 0.0921 - val_loss: 0.0909
Epoch 46/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 19s 41ms/step - loss: 0.0922 - val_loss: 0.0913
Epoch 47/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 16s 35ms/step - loss: 0.0920 - val_loss: 0.0909
Epoch 48/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 14s 31ms/step - loss: 0.0920 - val_loss: 0.0908
Epoch 49/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 16s 34ms/step - loss: 0.0919 - val_loss: 0.0908
Epoch 50/50
469/469 ━━━━━━━━━━━━━━━━━━━━ 14s 30ms/step - loss: 0.0918 - val_loss: 0.0908
313/313 ━━━━━━━━━━━━━━━━━━━━ 3s 7ms/step
```