Exp 9 Build and demonstrate an autoencoder network using neural layers for data compression on image dataset

import numpy as np

import matplotlib.pyplot as plt

from tensorflow.keras.layers import Input, Dense

from tensorflow.keras.models import Model

from tensorflow.keras.datasets import fashion\_mnist

# Load Fashion MNIST data

(x\_train, \_), (x\_test, \_) = fashion\_mnist.load\_data()

# Normalize data

x\_train = x\_train.astype('float32') / 255.

x\_test = x\_test.astype('float32') / 255.

# Flatten images to vectors

x\_train\_flat = x\_train.reshape((len(x\_train), 28 \* 28))

x\_test\_flat = x\_test.reshape((len(x\_test), 28 \* 28))

# Autoencoder architecture parameters

input\_dim = 28 \* 28

encoding\_dim = 64

# Encoder

input\_img = Input(shape=(input\_dim,))

encoded = Dense(128, activation='relu')(input\_img)

encoded = Dense(encoding\_dim, activation='relu')(encoded)

# Decoder

decoded = Dense(128, activation='relu')(encoded)

decoded = Dense(input\_dim, activation='sigmoid')(decoded)

# Autoencoder model

autoencoder = Model(input\_img, decoded)

# Encoder model for compressed output

encoder = Model(input\_img, encoded)

# Compile autoencoder

autoencoder.compile(optimizer='adam', loss='binary\_crossentropy')

# Train autoencoder

autoencoder.fit(x\_train\_flat, x\_train\_flat,

                epochs=50,

                batch\_size=256,

                shuffle=True,

                validation\_data=(x\_test\_flat, x\_test\_flat))

# Encode and decode test images

encoded\_imgs = encoder.predict(x\_test\_flat)

decoded\_imgs = autoencoder.predict(x\_test\_flat)

# Display original and reconstructed images

n = 10

plt.figure(figsize=(20, 4))

for i in range(n):

    # Original

    ax = plt.subplot(2, n, i + 1)

    plt.imshow(x\_test[i], cmap='gray')

    plt.title("Original")

    plt.axis('off')

    # Reconstructed

    ax = plt.subplot(2, n, i + 1 + n)

    plt.imshow(decoded\_imgs[i].reshape(28, 28), cmap='gray')

    plt.title("Reconstructed")

    plt.axis('off')

plt.show()

output

Downloading data from <https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz>

**29515/29515** ━━━━━━━━━━━━━━━━━━━━ **0s** 0us/step

Downloading data from <https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz>

**26421880/26421880** ━━━━━━━━━━━━━━━━━━━━ **0s** 0us/step

Downloading data from <https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz>

**5148/5148** ━━━━━━━━━━━━━━━━━━━━ **0s** 1us/step

Downloading data from <https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz>

**4422102/4422102** ━━━━━━━━━━━━━━━━━━━━ **0s** 0us/step

Epoch 1/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **5s** 16ms/step - loss: 0.4438 - val\_loss: 0.3123

Epoch 2/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 18ms/step - loss: 0.3060 - val\_loss: 0.3006

Epoch 3/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 16ms/step - loss: 0.2954 - val\_loss: 0.2938

Epoch 4/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **5s** 15ms/step - loss: 0.2905 - val\_loss: 0.2902

Epoch 5/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **5s** 22ms/step - loss: 0.2870 - val\_loss: 0.2894

Epoch 6/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 15ms/step - loss: 0.2841 - val\_loss: 0.2848

Epoch 7/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 15ms/step - loss: 0.2827 - val\_loss: 0.2835

Epoch 8/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **6s** 21ms/step - loss: 0.2805 - val\_loss: 0.2815

Epoch 9/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 16ms/step - loss: 0.2786 - val\_loss: 0.2803

Epoch 10/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **5s** 16ms/step - loss: 0.2782 - val\_loss: 0.2793

Epoch 11/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **5s** 21ms/step - loss: 0.2763 - val\_loss: 0.2788

Epoch 12/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 15ms/step - loss: 0.2754 - val\_loss: 0.2782

Epoch 13/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 15ms/step - loss: 0.2759 - val\_loss: 0.2772

Epoch 14/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **5s** 21ms/step - loss: 0.2749 - val\_loss: 0.2766

Epoch 15/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 17ms/step - loss: 0.2740 - val\_loss: 0.2759

Epoch 16/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 16ms/step - loss: 0.2737 - val\_loss: 0.2754

Epoch 17/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **5s** 21ms/step - loss: 0.2730 - val\_loss: 0.2750

Epoch 18/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 15ms/step - loss: 0.2721 - val\_loss: 0.2746

Epoch 19/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 16ms/step - loss: 0.2721 - val\_loss: 0.2743

Epoch 20/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **5s** 19ms/step - loss: 0.2719 - val\_loss: 0.2739

Epoch 21/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 16ms/step - loss: 0.2716 - val\_loss: 0.2735

Epoch 22/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 16ms/step - loss: 0.2710 - val\_loss: 0.2732

Epoch 23/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 17ms/step - loss: 0.2710 - val\_loss: 0.2730

Epoch 24/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 19ms/step - loss: 0.2709 - val\_loss: 0.2727

Epoch 25/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 15ms/step - loss: 0.2701 - val\_loss: 0.2724

Epoch 26/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **6s** 20ms/step - loss: 0.2703 - val\_loss: 0.2724

Epoch 27/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 16ms/step - loss: 0.2697 - val\_loss: 0.2718

Epoch 28/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 15ms/step - loss: 0.2686 - val\_loss: 0.2720

Epoch 29/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **6s** 20ms/step - loss: 0.2694 - val\_loss: 0.2715

Epoch 30/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 15ms/step - loss: 0.2687 - val\_loss: 0.2713

Epoch 31/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **5s** 15ms/step - loss: 0.2690 - val\_loss: 0.2711

Epoch 32/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **5s** 21ms/step - loss: 0.2688 - val\_loss: 0.2710

Epoch 33/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 16ms/step - loss: 0.2684 - val\_loss: 0.2708

Epoch 34/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 16ms/step - loss: 0.2687 - val\_loss: 0.2707

Epoch 35/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **5s** 21ms/step - loss: 0.2687 - val\_loss: 0.2708

Epoch 36/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 16ms/step - loss: 0.2683 - val\_loss: 0.2705

Epoch 37/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 15ms/step - loss: 0.2672 - val\_loss: 0.2703

Epoch 38/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **5s** 20ms/step - loss: 0.2682 - val\_loss: 0.2701

Epoch 39/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 17ms/step - loss: 0.2682 - val\_loss: 0.2700

Epoch 40/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 16ms/step - loss: 0.2679 - val\_loss: 0.2699

Epoch 41/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **6s** 21ms/step - loss: 0.2671 - val\_loss: 0.2700

Epoch 42/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 15ms/step - loss: 0.2668 - val\_loss: 0.2698

Epoch 43/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **5s** 15ms/step - loss: 0.2672 - val\_loss: 0.2697

Epoch 44/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **5s** 21ms/step - loss: 0.2675 - val\_loss: 0.2698

Epoch 45/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 16ms/step - loss: 0.2666 - val\_loss: 0.2697

Epoch 46/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **5s** 16ms/step - loss: 0.2668 - val\_loss: 0.2693

Epoch 47/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **5s** 17ms/step - loss: 0.2662 - val\_loss: 0.2699

Epoch 48/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **5s** 15ms/step - loss: 0.2670 - val\_loss: 0.2692

Epoch 49/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **5s** 19ms/step - loss: 0.2669 - val\_loss: 0.2692

Epoch 50/50

**235/235** ━━━━━━━━━━━━━━━━━━━━ **4s** 17ms/step - loss: 0.2670 - val\_loss: 0.2693

**313/313** ━━━━━━━━━━━━━━━━━━━━ **1s** 2ms/step

**313/313** ━━━━━━━━━━━━━━━━━━━━ **1s** 2ms/step

