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 cifar10  
  
# Load the CIFAR-10 dataset  
(x\_train, \_), (x\_test, \_) = cifar10.load\_data()  
  
# Normalize pixel values to be between 0 and 1  
x\_train = x\_train.astype('float32') / 255.0  
x\_test = x\_test.astype('float32') / 255.0  
  
# Flatten the images for the autoencoder  
x\_train = x\_train.reshape((len(x\_train), np.prod(x\_train.shape[1:])))  
x\_test = x\_test.reshape((len(x\_test), np.prod(x\_test.shape[1:])))  
  
# Define the autoencoder model  
encoding\_dim = 128 # Size of the encoded representations  
input\_img = Input(shape=(3072,)) # 32x32x3 images flattened  
encoded = Dense(encoding\_dim, activation='relu')(input\_img)  
decoded = Dense(3072, activation='sigmoid')(encoded)  
  
autoencoder = Model(input\_img, decoded)  
  
# Compile the autoencoder  
autoencoder.compile(optimizer='adam', loss='binary\_crossentropy')  
  
# Train the autoencoder  
autoencoder.fit(x\_train, x\_train, epochs=50, batch\_size=256, shuffle=True, validation\_data=(x\_test, x\_test))  
  
# Create a separate encoder model  
encoder = Model(input\_img, encoded)  
  
# Encode the test images  
encoded\_imgs = encoder.predict(x\_test)  
  
# Decode the encoded images  
decoded\_imgs = autoencoder.predict(x\_test)  
  
# Display original and reconstructed images  
n = 10 # Number of images to display  
plt.figure(figsize=(20, 4))  
for i in range(n):  
 # Original images  
 ax = plt.subplot(2, n, i + 1)  
 plt.imshow(x\_test[i].reshape(32, 32, 3))  
 plt.gray()  
 ax.get\_xaxis().set\_visible(False)  
 ax.get\_yaxis().set\_visible(False)  
  
 # Reconstructed images  
 ax = plt.subplot(2, n, i + 1 + n)  
 plt.imshow(decoded\_imgs[i].reshape(32, 32, 3))  
 plt.gray()  
 ax.get\_xaxis().set\_visible(False)  
 ax.get\_yaxis().set\_visible(False)  
  
plt.show()

Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz  
170498071/170498071 [==============================] - 4s 0us/step  
Epoch 1/50  
196/196 [==============================] - 13s 58ms/step - loss: 0.6391 - val\_loss: 0.6210  
Epoch 2/50  
196/196 [==============================] - 10s 51ms/step - loss: 0.6133 - val\_loss: 0.6086  
Epoch 3/50  
196/196 [==============================] - 9s 46ms/step - loss: 0.6043 - val\_loss: 0.6009  
Epoch 4/50  
196/196 [==============================] - 9s 45ms/step - loss: 0.5984 - val\_loss: 0.5963  
Epoch 5/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5933 - val\_loss: 0.5969  
Epoch 6/50  
196/196 [==============================] - 10s 51ms/step - loss: 0.5906 - val\_loss: 0.5899  
Epoch 7/50  
196/196 [==============================] - 9s 46ms/step - loss: 0.5883 - val\_loss: 0.5880  
Epoch 8/50  
196/196 [==============================] - 9s 46ms/step - loss: 0.5869 - val\_loss: 0.5888  
Epoch 9/50  
196/196 [==============================] - 9s 47ms/step - loss: 0.5859 - val\_loss: 0.5865  
Epoch 10/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5858 - val\_loss: 0.5860  
Epoch 11/50  
196/196 [==============================] - 10s 51ms/step - loss: 0.5855 - val\_loss: 0.5859  
Epoch 12/50  
196/196 [==============================] - 9s 46ms/step - loss: 0.5848 - val\_loss: 0.5862  
Epoch 13/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5845 - val\_loss: 0.5859  
Epoch 14/50  
196/196 [==============================] - 10s 51ms/step - loss: 0.5847 - val\_loss: 0.5857  
Epoch 15/50  
196/196 [==============================] - 9s 44ms/step - loss: 0.5846 - val\_loss: 0.5850  
Epoch 16/50  
196/196 [==============================] - 9s 47ms/step - loss: 0.5846 - val\_loss: 0.5863  
Epoch 17/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5843 - val\_loss: 0.5850  
Epoch 18/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5845 - val\_loss: 0.5860  
Epoch 19/50  
196/196 [==============================] - 8s 43ms/step - loss: 0.5843 - val\_loss: 0.5850  
Epoch 20/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5841 - val\_loss: 0.5850  
Epoch 21/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5843 - val\_loss: 0.5852  
Epoch 22/50  
196/196 [==============================] - 9s 47ms/step - loss: 0.5844 - val\_loss: 0.5848  
Epoch 23/50  
196/196 [==============================] - 9s 47ms/step - loss: 0.5840 - val\_loss: 0.5854  
Epoch 24/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5842 - val\_loss: 0.5856  
Epoch 25/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5843 - val\_loss: 0.5852  
Epoch 26/50  
196/196 [==============================] - 9s 46ms/step - loss: 0.5841 - val\_loss: 0.5848  
Epoch 27/50  
196/196 [==============================] - 9s 45ms/step - loss: 0.5843 - val\_loss: 0.5855  
Epoch 28/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5840 - val\_loss: 0.5848  
Epoch 29/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5843 - val\_loss: 0.5854  
Epoch 30/50  
196/196 [==============================] - 9s 46ms/step - loss: 0.5839 - val\_loss: 0.5848  
Epoch 31/50  
196/196 [==============================] - 9s 46ms/step - loss: 0.5840 - val\_loss: 0.5853  
Epoch 32/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5842 - val\_loss: 0.5849  
Epoch 33/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5841 - val\_loss: 0.5847  
Epoch 34/50  
196/196 [==============================] - 8s 43ms/step - loss: 0.5839 - val\_loss: 0.5850  
Epoch 35/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5840 - val\_loss: 0.5847  
Epoch 36/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5843 - val\_loss: 0.5851  
Epoch 37/50  
196/196 [==============================] - 9s 46ms/step - loss: 0.5839 - val\_loss: 0.5852  
Epoch 38/50  
196/196 [==============================] - 10s 52ms/step - loss: 0.5839 - val\_loss: 0.5852  
Epoch 39/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5841 - val\_loss: 0.5847  
Epoch 40/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5840 - val\_loss: 0.5853  
Epoch 41/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5841 - val\_loss: 0.5851  
Epoch 42/50  
196/196 [==============================] - 8s 43ms/step - loss: 0.5839 - val\_loss: 0.5846  
Epoch 43/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5840 - val\_loss: 0.5851  
Epoch 44/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5842 - val\_loss: 0.5857  
Epoch 45/50  
196/196 [==============================] - 9s 47ms/step - loss: 0.5839 - val\_loss: 0.5848  
Epoch 46/50  
196/196 [==============================] - 9s 44ms/step - loss: 0.5839 - val\_loss: 0.5846  
Epoch 47/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5839 - val\_loss: 0.5847  
Epoch 48/50  
196/196 [==============================] - 9s 48ms/step - loss: 0.5842 - val\_loss: 0.5848  
Epoch 49/50  
196/196 [==============================] - 9s 46ms/step - loss: 0.5839 - val\_loss: 0.5854  
Epoch 50/50  
196/196 [==============================] - 9s 46ms/step - loss: 0.5839 - val\_loss: 0.5851  
313/313 [==============================] - 1s 3ms/step  
313/313 [==============================] - 1s 3ms/step

