Ex.No: 6 Roll.No: 210701182

# PYTHON PROGRAM TO BUILD AUTOENCODERS WITH KERAS

Aim:

To build autoencoders with keras in python.

# Procedure:

1. Import NumPy and Keras modules for building and training the autoencoder.
2. Define the input dimension (e.g., 784 for flattened 28x28 MNIST images) and encoding dimension.
3. Create an input layer with the specified input dimension.
4. Build the encoder part of the autoencoder with three Dense layers, reducing dimensionality to the encoding dimension.
5. Construct the decoder part of the autoencoder with three Dense layers, reconstructing the input to the original dimension.
6. Define the autoencoder model with the input layer and the reconstructed output.
7. Create a separate encoder model to extract encoded features from the input.
8. Compile the autoencoder using Adam optimizer and binary cross- entropy loss.
9. Generate dummy training and test data and train the autoencoder for 50 epochs.
10. Use the encoder to obtain encoded representations and the autoencoder to reconstruct the input, printing the shapes of the results.

# Code:

import numpy as np

from keras.layers import Input, Dense from keras.models import Model

# Define input dimension

input\_dim = 784 # For example, flattened 28x28 MNIST images

# Define encoding dimension encoding\_dim = 32

# Input layer

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

# Encoder layers

encoded = Dense(128, activation='relu')(input\_img) encoded = Dense(64, activation='relu')(encoded)

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

# Decoder layers

decoded = Dense(64, activation='relu')(encoded) decoded = Dense(128, activation='relu')(decoded)

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

# Autoencoder model

autoencoder = Model(input\_img, decoded)

# Separate encoder model

encoder = Model(input\_img, encoded) # Compile the model

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

# Generate dummy data for demonstration x\_train = np.random.random((1000, input\_dim)) x\_test = np.random.random((200, input\_dim))

# Train the autoencoder autoencoder.fit(x\_train, x\_train,

epochs=50, batch\_size=256, shuﬄe=True,

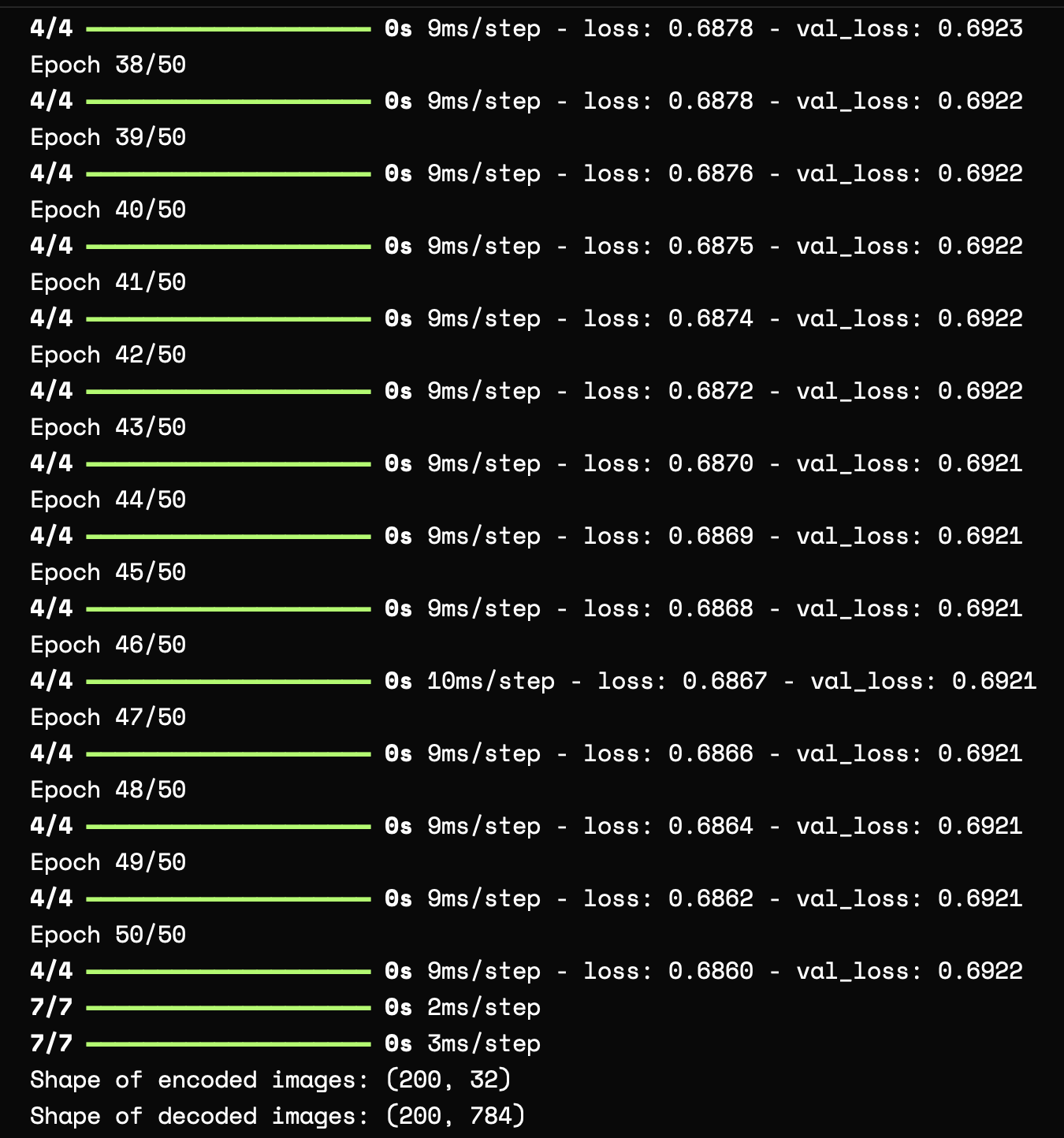
validation\_data=(x\_test, x\_test))

# Use the encoder to encode some input encoded\_imgs = encoder.predict(x\_test)

# Use the autoencoder to reconstruct some input decoded\_imgs = autoencoder.predict(x\_test)

print("Shape of encoded images:", encoded\_imgs.shape) print("Shape of decoded images:", decoded\_imgs.shape)

# Output:



Result:

Thus, to build autoencoders with Keras has been done successfully.