Vrushti sanghavi

52 cse(ds)

Dl EXPERIMENT 5

**CODE:--**

import keras from keras import layers from keras.datasets import mnist import numpy as np

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

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

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:]))) print(x\_train.shape) print(x\_test.shape)

encoding\_dim = 32 input\_img = keras.Input(shape=(784,)) encoded = layers.Dense(encoding\_dim,activation='relu')(input\_img) decoded =layers.Dense(784,activation = 'sigmoid')(encoded) autoencoder = keras.Model(input\_img,decoded) encoder = keras.Model(input\_img,encoded)

encoded\_input = keras.Input(shape=(encoding\_dim,)) decoder\_layer = autoencoder.layers[-1] decoder = keras.Model(encoded\_input,decoder\_layer(encoded\_input)) autoencoder.compile(optimizer='adam',loss='binary\_crossentropy')

autoencoder.fit(x\_train,x\_train,epochs=20,batch\_size=64,shuffle=True,validation\_data=(x\_test,x

\_test))

encoded\_imgs = encoder.predict(x\_test) decoded\_imgs = decoder.predict(encoded\_imgs) import matplotlib.pyplot as plt

n = 10

plt.figure(figsize = (20,4)) for i in range(n):

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)

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()

**OUTPUT:-**







