```
import keras
from keras import layers

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')


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)
autoencoder.fit(x_train, x_train,epochs=50,batch_size=256,shuffle=True,validation_data=(x_tes
encoded_imgs = encoder.predict(x_test)
decoded_imgs = decoder.predict(encoded_imgs)
```

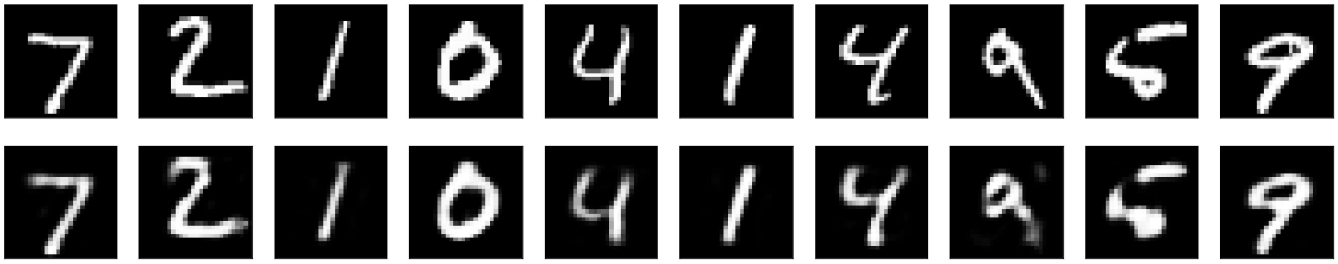
```
Epoch 24/50
235/235 [==============================] - 3s 14ms/step - loss: 0.0933 - val_loss: 0.
Epoch 25/50
235/235 [==============================] - 3s 15ms/step - loss: 0.0933 - val_loss: 0.
Epoch 26/50
235/235 [==============================] - 3s 14ms/step - loss: 0.0933 - val_loss: 0.
Epoch 27/50
235/235 [==============================] - 3s 14ms/step - loss: 0.0932 - val_loss: 0.
Epoch 28/50
235/235 [==============================] - 3s 14ms/step - loss: 0.0932 - val_loss: 0.
Epoch 29/50
235/235 [==============================] - 5s 20ms/step - loss: 0.0931 - val_loss: 0.
Epoch 30/50
235/235 [==============================] - 4s 15ms/step - loss: 0.0931 - val_loss: 0.
Epoch 31/50
235/235 [==============================] - 3s 15ms/step - loss: 0.0931 - val_loss: 0.
Epoch 32/50
235/235 [==============================] - 3s 15ms/step - loss: 0.0931 - val_loss: 0.
Epoch 33/50
235/235 [==============================] - 3s 14ms/step - loss: 0.0930 - val_loss: 0.
Epoch 34/50
235/235 [==============================] - 3s 14ms/step - loss: 0.0930 - val_loss: 0.
Epoch 35/50
235/235 [==============================] - 3s 15ms/step - loss: 0.0930 - val_loss: 0.
```

```
Epoch 36/50
235/235 [==============================] - 3s 14ms/step - loss: 0.0930 - val_loss: 0.
Epoch 37/50
235/235 [==============================] - 4s 17ms/step - loss: 0.0930 - val_loss: 0.
Epoch 38/50
235/235 [==============================] - 4s 16ms/step - loss: 0.0929 - val_loss: 0.
Epoch 39/50
235/235 [==============================] - 3s 14ms/step - loss: 0.0929 - val_loss: 0.
Epoch 40/50
235/235 [==============================] - 3s 14ms/step - loss: 0.0929 - val_loss: 0.
Epoch 41/50
235/235 [==============================] - 3s 13ms/step - loss: 0.0929 - val_loss: 0.
Epoch 42/50
235/235 [==============================] - 3s 14ms/step - loss: 0.0929 - val_loss: 0.
Epoch 43/50
235/235 [==============================] - 3s 14ms/step - loss: 0.0929 - val_loss: 0.
Epoch 44/50
235/235 [==============================] - 3s 14ms/step - loss: 0.0929 - val_loss: 0.
Epoch 45/50
235/235 [==============================] - 3s 14ms/step - loss: 0.0928 - val_loss: 0.
Epoch 46/50
235/235 [==============================] - 3s 14ms/step - loss: 0.0928 - val_loss: 0.
Epoch 47/50
235/235 [==============================] - 3s 13ms/step - loss: 0.0928 - val_loss: 0.
Epoch 48/50
235/235 [==============================] - 3s 14ms/step - loss: 0.0928 - val_loss: 0.
Epoch 49/50
235/235 [==============================] - 3s 14ms/step - loss: 0.0928 - val_loss: 0.
Epoch 50/50
235/235 [==============================] - 3s 14ms/step - loss: 0.0928 - val_loss: 0.
313/313 [==============================] - 1s 2ms/step
313/313 [==============================] - 1s 1ms/step
```

```
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()
```



✓ 1s completed at 2:48 PM