Lab Report

Name--Devyani Gorkar
Roll no.--B20ME027

Ques1-About code

First we import all the necessary libraries and files.**dataset** is the data frame consisting of the mnist data.Y is the labels column and X is the input matrix.Using train_test_split X and Y 70-30 split is done.Using training and testing filter 1 and 0 is extracted from X .So Y_train and Y_test has only 1 and 0 as labels .In mnist dataset input dimension is 784, so size_input=784, size of 3 encoding hidden layers is 616,524,480 and the size of bottleneck layer or middle layer is 420.Now we start making decoder layers size of decoder layers are and the order of decoding layer is reverse of encoding layer i.e. 480,524,616.We use model_autoencoder.fit() to fit the training data and then predict the results on X _test.While compiling optimizer = adamax and loss=mse.The original image as well as reconstructed image was plotted for both 0 and 1.In part © of question 1 we construct an encoder model which reduces the dimension of the input.encoded_x_train is the training array with reduced dimension of the input data.Using LinearSVC() we use the encoded_x_train to train the svm model and check the accuracy score on encoded_x_test

Ques 1-Observations and Results:

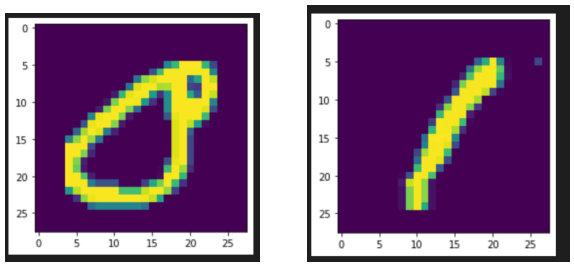
- Model Summary

```
Model: "model_21"
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
input_26 (InputLayer)       [(None, 784)]             0
_________________________________________________________________
dense_185 (Dense)           (None, 616)               483560
_________________________________________________________________
dense_186 (Dense)           (None, 524)               323308
_________________________________________________________________
dense_187 (Dense)           (None, 480)               252000
_________________________________________________________________
dense_188 (Dense)           (None, 420)               202020
_________________________________________________________________
dense_189 (Dense)           (None, 480)               202080
_________________________________________________________________
dense_190 (Dense)           (None, 524)               252044
_________________________________________________________________
dense_191 (Dense)           (None, 616)               323400
_________________________________________________________________
dense_192 (Dense)           (None, 784)               483728
=================================================================
Total params: 2,522,140
Trainable params: 2,522,140
```

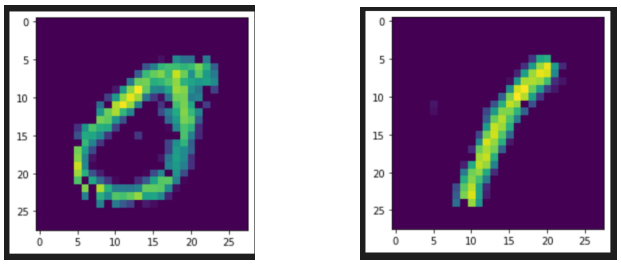
So total parameters required for training the autoencoder=2,522,140

- Plot of original and reconstructed

  Original:



  Reconstructed:



So it can be seen that during reconstruction information is lost but still the reconstructed image is close to the original image.

- Part c-Accuracy of svm on data with reduced dimensions.

  So the accuracy obtained on LinearSVC is `0.9974463738508682`

Question-2-About code

In this question we create a CNN model with 2 convolutional layers.Here input size considered is (200,200,3) and activation function used is relu for convolutional layers.First Convolutional layer has kernel size(5,5) and filters=28 .For second layer kernel size(5,5) an filters=28.0 padding.And one pooling layer is also used in between layers the size is (14,14)

Questions-2-Observations and results.

The model summary.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 196, 196, 28)      2128
_________________________________________________________________
 max_pooling2d (MaxPooling2D) (None, 14, 14, 28)       0
_________________________________________________________________
 conv2d_1 (Conv2D)           (None, 11, 11, 32)        14368
_________________________________________________________________
 flatten (Flatten)           (None, 3872)              0
_________________________________________________________________
 dense (Dense)               (None, 5)                 19365
=================================================================
Total params: 35,861
Trainable params: 35,861
Non-trainable params: 0
_________________________________________________________________
```

Number of params are given by (F*F*C*K)+K biases where F=Filter size, C=depth of input,K=no. Of filters.

Question 3:About code

In this question dataset used is cifar-10 dataset.Using Ordinal encode we convert the y_train and y_test samples.Later we define a Sequential model with Convolutional layers and 2 times pooling.The activation function used is relu.And in the last layer activation function used is softmax.Then x_train and y_train is used to train the model with 2 epochs.The input image dimensions are (32,32,3).The confusion matrix and classification report is calculated.

Question3-Observations and results

Model Summary:

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_8 (Conv2D)           (None, 30, 30, 32)        896
_________________________________________________________________
 max_pooling2d_7 (MaxPooling2 (None, 15, 15, 32)       0
_________________________________________________________________
 conv2d_9 (Conv2D)           (None, 13, 13, 64)        18496
_________________________________________________________________
 max_pooling2d_8 (MaxPooling2 (None, 6, 6, 64)         0
_________________________________________________________________
 flatten_4 (Flatten)         (None, 2304)              0
_________________________________________________________________
 dense_80 (Dense)            (None, 64)                147520
_________________________________________________________________
 dense_81 (Dense)            (None, 10)                650
=================================================================
Total params: 167,562
Trainable params: 167,562
Non-trainable params: 0
_________________________________________________________________
```

Confusion matrix:

```
array([[783,  15,  66,   9,  25,   5,  10,   3,  61,  23],
       [ 79, 732,  12,  11,  16,   6,  19,   5,  29,  91],
       [ 93,   8, 510,  39, 161,  61,  82,  20,  13,  13],
       [ 33,  10, 133, 331, 143, 139, 155,  23,  21,  12],
       [ 53,   5,  99,  32, 639,  14, 106,  38,  10,   4],
       [ 28,   7, 168, 115,  96, 465,  79,  25,  12,   5],
       [ 10,   5,  64,  26,  55,  15, 812,   4,   6,   3],
       [ 39,   3,  66,  43, 142,  87,  22, 566,   6,  26],
       [202,  42,  27,  13,  10,  10,  13,   3, 655,  25],
       [ 97, 124,  24,  16,  14,   7,  30,  11,  39, 638]], dtype=int64)
```

Classification report:
The accuracy score obtained through the classification report is 0.67.

Google colab link:
https://colab.research.google.com/drive/1qJuVmyXUilufl-WezQgkfynUevhbQthM?usp=sharing