[CV21] Assignment 1 - Di Zhuang

2.2

The accuracy achieved with the linear classifier is as follows:

[Epoch 01] Loss: 1.0115
[Epoch 01] Acc.: 47.4206%
[Epoch 02] Loss: 0.7279
[Epoch 02] Acc.: 47.2222%
[Epoch 03] Loss: 0.6941
[Epoch 03] Acc.: 48.8095%
[Epoch 04] Loss: 0.6929
[Epoch 04] Acc.: 48.0159%
[Epoch 05] Loss: 0.6929
[Epoch 05] Acc.: 49.2063%
[Epoch 06] Loss: 0.6931
[Epoch 06] Acc.: 48.8095%
[Epoch 07] Loss: 0.6931
[Epoch 07] Acc.: 48.4127%
[Epoch 08] Loss: 0.6929
[Epoch 08] Acc.: 49.0079%
[Epoch 09] Loss: 0.6929
[Epoch 09] Acc.: 48.4127%
[Epoch 10] Loss: 0.6932
[Epoch 10] Acc.: 48.6111%

This is an expected result. Single linear layer is only capable of separating linearly separable data points. In our case, it is trying to learn a 2D line that separates cluster 0 and cluster 1. However, the given (x, y) coordinates of the two clusters are not linearly separable, as the two clusters form two concentric rings. Therefore, it is expected that the network cannot separate the two clusters satisfactorily.

2.3

The accuracy achieved with the new classifier is as follows:

[Epoch 01] Loss: 0.5960
[Epoch 01] Acc.: 68.4524%
[Epoch 02] Loss: 0.3900
[Epoch 02] Acc.: 98.6111%

[Epoch 03] Loss: 0.1598
[Epoch 03] Acc.: 99.4048%
[Epoch 04] Loss: 0.0660
[Epoch 04] Acc.: 99.8016%
[Epoch 05] Loss: 0.0347
[Epoch 05] Acc.: 99.6032%
[Epoch 06] Loss: 0.0218
[Epoch 06] Acc.: 99.8016%
[Epoch 07] Loss: 0.0154
[Epoch 07] Acc.: 99.8016%
[Epoch 08] Loss: 0.0112
[Epoch 08] Acc.: 99.6032%
[Epoch 09] Loss: 0.0087
[Epoch 09] Acc.: 99.6032%
[Epoch 10] Loss: 0.0075
[Epoch 10] Acc.: 99.8016%

The results are better compared to the previous linear classifier because now there are two hidden layers in the network. With two hidden layers, the network is capable of forming convex regions in the data space to separate data points, which allows the two clusters of “ring” shape to be separated with high accuracy.

2.4

By changing the cartesian coordinates to polar coordinates and training the linear classifier with the polar coordinates, I obtained the following results:

[Epoch 01] Loss: 0.6599
[Epoch 01] Acc.: 50.1984%
[Epoch 02] Loss: 0.5795
[Epoch 02] Acc.: 51.3889%
[Epoch 03] Loss: 0.5551
[Epoch 03] Acc.: 55.1587%
[Epoch 04] Loss: 0.5312
[Epoch 04] Acc.: 64.4841%
[Epoch 05] Loss: 0.5083
[Epoch 05] Acc.: 77.9762%
[Epoch 06] Loss: 0.4859
[Epoch 06] Acc.: 82.1429%
[Epoch 07] Loss: 0.4646
[Epoch 07] Acc.: 89.8810%
[Epoch 08] Loss: 0.4437
[Epoch 08] Acc.: 91.2698%
[Epoch 09] Loss: 0.4242

[Epoch 09] Acc.: 94.4444%
[Epoch 10] Loss: 0.4052
[Epoch 10] Acc.: 97.0238%

I chose polar coordinates because the two clusters are two rings centered at the origin of the coordinate system and they differ only by the radius of the rings. In this case, the r component in the polar coordinates allows the two clusters to be linearly separable.

3.3

One linear layer perceptron results:

[Epoch 01] Loss: 0.4108
[Epoch 01] Acc.: 90.4000%
[Epoch 02] Loss: 0.3325
[Epoch 02] Acc.: 91.2700%
[Epoch 03] Loss: 0.3227
[Epoch 03] Acc.: 90.9500%
[Epoch 04] Loss: 0.3140
[Epoch 04] Acc.: 90.5000%
[Epoch 05] Loss: 0.3128
[Epoch 05] Acc.: 91.4900%

Multi-layer perceptron results:

[Epoch 01] Loss: 0.3803
[Epoch 01] Acc.: 92.5500%
Epoch 02] Loss: 0.2387
[Epoch 02] Acc.: 93.2300%
[Epoch 03] Loss: 0.2063
[Epoch 03] Acc.: 94.1500%
[Epoch 04] Loss: 0.1893
[Epoch 04] Acc.: 95.1000%
[Epoch 05] Loss: 0.1782
[Epoch 05] Acc.: 95.2600%

3.4

Convolutional network results:

[Epoch 01] Loss: 0.2596
[Epoch 01] Acc.: 96.9700%
[Epoch 02] Loss: 0.0818
[Epoch 02] Acc.: 98.1600%
[Epoch 03] Loss: 0.0601
[Epoch 03] Acc.: 98.3900%
[Epoch 04] Loss: 0.0473
[Epoch 04] Acc.: 98.2300%
[Epoch 05] Loss: 0.0411
[Epoch 05] Acc.: 98.1400%

3.5

The following computation shows that the multi-layer perceptron classifier has more parameters to learn in comparison to the convolutional network, but the performance of the convolutional network is better in terms of accuracy.

Multi-layer perceptron:

I: number of neurons in the input layer = 28 * 28 = 784
H: number of neurons in the hidden layer = 32
O: number of neurons in the output layer = 10

#parameters of the hidden layers = I * H = 25088
#parameters of the output layers = H * O = 320
#bias terms of the hidden layer = H = 32
#bias terms of the output layer = O = 10

In total I * H + H * O + H + O = **25450** parameters.

Convolutional network:

I_1: number of input channels of the first convolutional layer = 1
O_1: number of output channels of the first convolutional layer = 8 = #bias terms
K_1: kernel size of the first convolutional layer = 3 * 3 = 9
I_2: number of input channels of the second convolutional layer = 8
O_2: number of output channels of the second convolutional layer = 16 = #bias terms
K_2: kernel size of the second convolutional layer = 3 * 3 = 9
I_3: number of input channels of the third convolutional layer = 16
O_3: number of output channels of the third convolutional layer = 32 = #bias terms

K_3: kernel size of the third convolutional layer = 3 * 3 = 9
I_4: input size of the linear layer = 32
O_4: output size of the linear layer = 10 = #bias terms

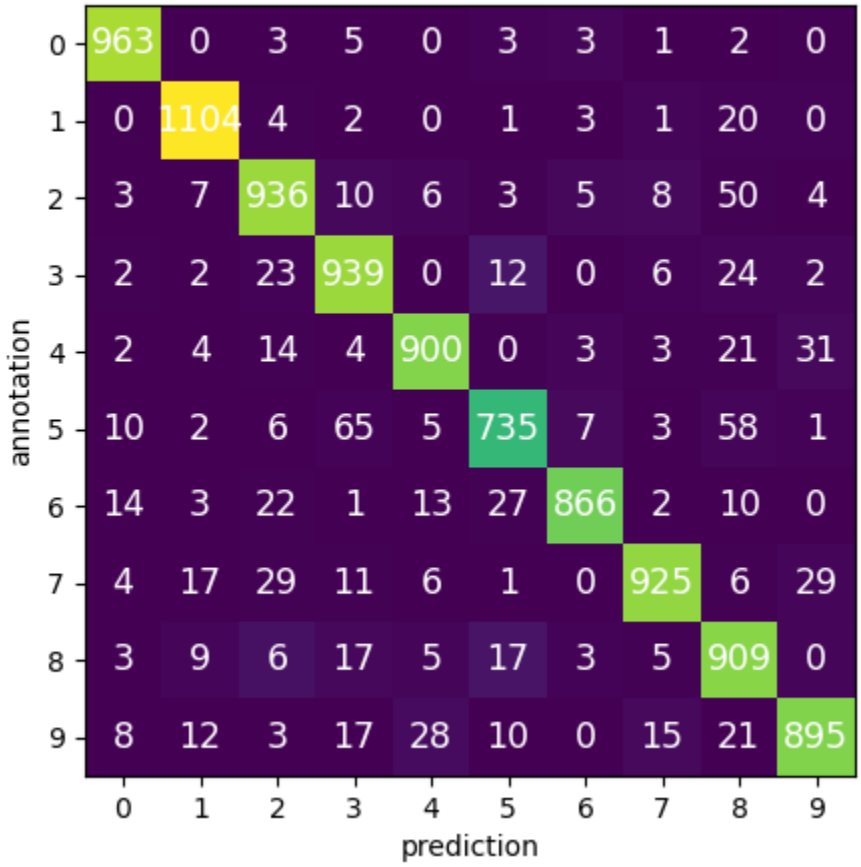
#parameters of the first convolutional layer = I_1 * O_1 * K_1 + O_1 = 1 * 8 * 3 * 3 + 8 = 80
#parameters of the second convolutional layer = I_2 * O_2 * K_2 + O_2 = 8 * 16 * 3 * 3 + 16 = 1168
#parameters of the third convolutional layer = I_3 * O_3 * K_3 + O_3 = 16 * 32 * 3 * 3 + 32 = 4640
#parameters of the linear output layer = I_4 * O_4 + O_4 = 32 * 10 + 10 = 330

In total 80 + 1168 + 4640 + 330 = **6218** parameters

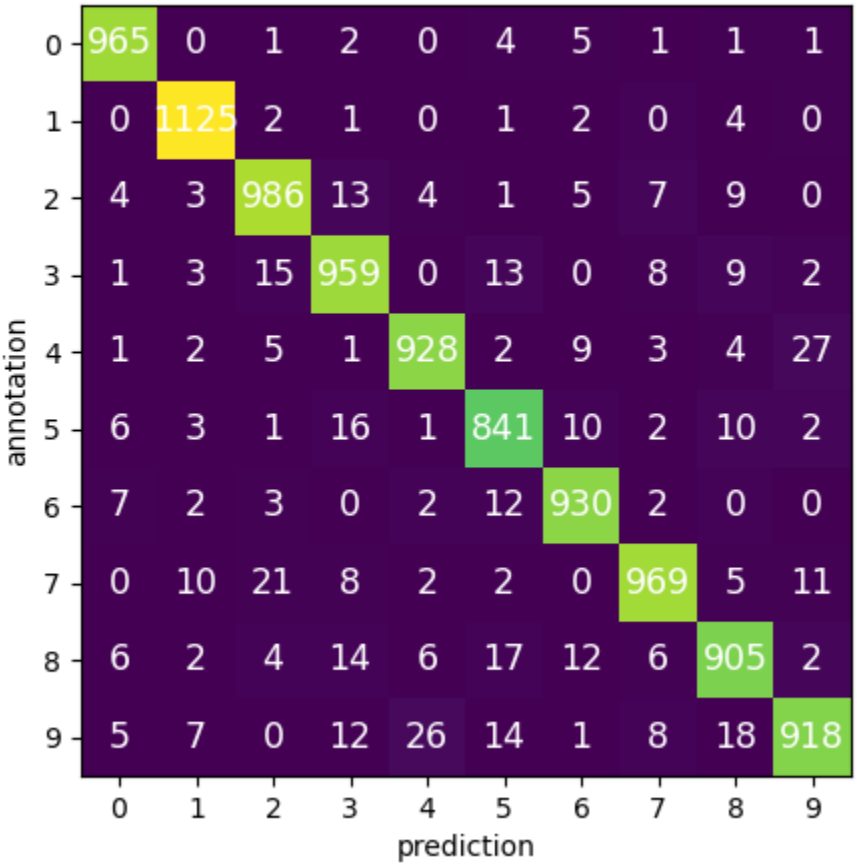
3.6

The confusion matrix for the single layer classifier, multi-layer classifier and the convolutional neural network is as follows. It can be seen that the numbers of the diagonal entries are the biggest, indicating that most of the samples are correctly classified, which corresponds to the 90%+ accuracy these classifiers achieved. In addition, by comparing the numbers and the brightness of the diagonal entries, it can be seen that the convolutional network performs the best, that is, the number of samples that it classified correctly was the highest for all classes, the next is the mult-layer classifier, and the last is the single layer classifier.

Single layer perceptron:



Multi-layer perceptron:



Convolutional network: