Assignment 1 Introduction to PyTorch

Yang Mu, 20/10/2021

263-5902-00L - Computer Vision HS2021

2 Simple 2D classifier

2.3 Training loop

I obtained 48.2143% accuracy with the linear classifier after 10 epochs. For this data set, the red points and the blue points form two circles inside and outside, and the linear classifier cannot separate these two types of points well. In this sense, this seemingly "not good" accuracy is the expected result.

2.4 Multi-layer perceptron

I obtained 99.8016% accuracy with the MLP after 10 epochs. After using 2 hidden layers with 16 neurons and ReLU layers, the MLP classifier becomes non-linear, it will be better than the linear classifier for these two classes whose boundary is approximately circular.

2.5 Feature transform

I squared two coordinates and added them to get the new two-dimensional coordinates (x, y are equal to the square of the radius), so that the feature is converted to the regions near the two points of a straight line with a slope of approximately 1. There is a certain distance between these two regions, making the linear classifier able to classify new data well.

After the transformation, I obtained 96.8254% accuracy with the linear classifier after 10 epochs, which proves the hypothesis of feature transform.

3 Digit classifier

3.3 Multi-layer perceptron

I obtained 92.5% accuracy with the linear classifier after 5 epochs. After using MLP, the new testing accuracy is 96.59%, which is a certain improvement compared to the linear classifier.

3.4 Convolutional network

With this architecture, I obtained 97.66% accuracy after 5 epochs.

3.5 Comparison of number of parameters

3.5.1 MLP with one hidden layer

The calculation formula for the number of linear layer parameters is:

out_features*in_features (*weights*) + out_features*1 (*biases*)

where out_features is the size of each output sample, and in_features is the size of each input sample.

- Hidden layer of dimension 32:

  $$32*28*28 + 32*1 = 25120$$

- Final linear prediction layer:

  $$10*32 + 10*1 = 330$$

So the overall number of parameters of the MLP with one hidden layer is 25450.

3.5.2 The convolutional network

The calculation formula for the number of convolutional layer parameters is:

kernel_size^2*in_channels* out_channels(*weights*) + out_channels*1 (*biases*)

where kernel_size is the size of the convolving kernel, in_channels is the number of channels in the input image, and out_channels is the number of channels produced by the convolution.

- nn.Conv2d with 3 × 3 kernel and 8 channels:

$$3*3*8 + 8 = 80$$

- nn.Conv2d with 3 × 3 kernel and 16 channels:

$$3*3*8*16 + 16 = 1168$$

- nn.Conv2d with 3 × 3 kernel and 32 channels:

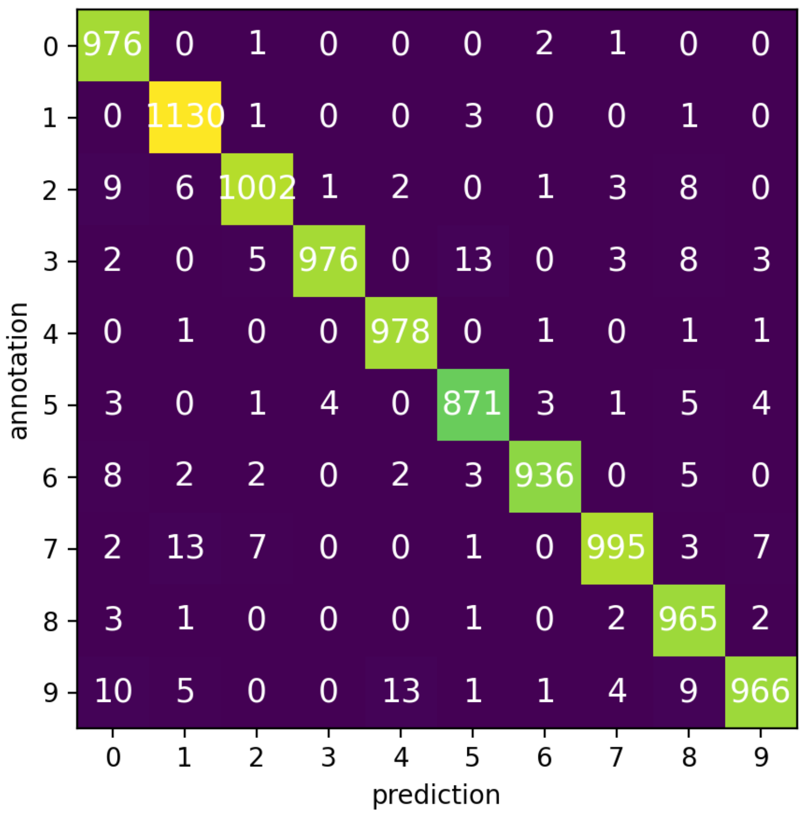
$$3*3*16*32 + 32 = 4640$$

- Final linear prediction layer:

$$10*32+10*1 = 330$$

The overall number of parameters of the convolutional network is 6218.

3.6 Confusion matrix



annotation \ prediction	0	1	2	3	4	5	6	7	8	9
0	976	0	1	0	0	0	2	1	0	0
1	0	1130	1	0	0	3	0	0	1	0
2	9	6	1002	1	2	0	1	3	8	0
3	2	0	5	976	0	13	0	3	8	3
4	0	1	0	0	978	0	1	0	1	1
5	3	0	1	4	0	871	3	1	5	4
6	8	2	2	0	2	3	936	0	5	0
7	2	13	7	0	0	1	0	995	3	7
8	3	1	0	0	0	1	0	2	965	2
9	10	5	0	0	13	1	1	4	9	966

It can be seen from the confusion matrix that the values on the diagonal are very large, which means that the overall accuracy of the model prediction is very high.

Among them, the recall rates for annotations of 0, 1 and 4 are relatively high, being 99.59%, 99.56% and 99.59% respectively. The recall rates for annotations of 3, 7 and 9 are relatively low, 96.63%, 96.79% and 95.74% respectively.