

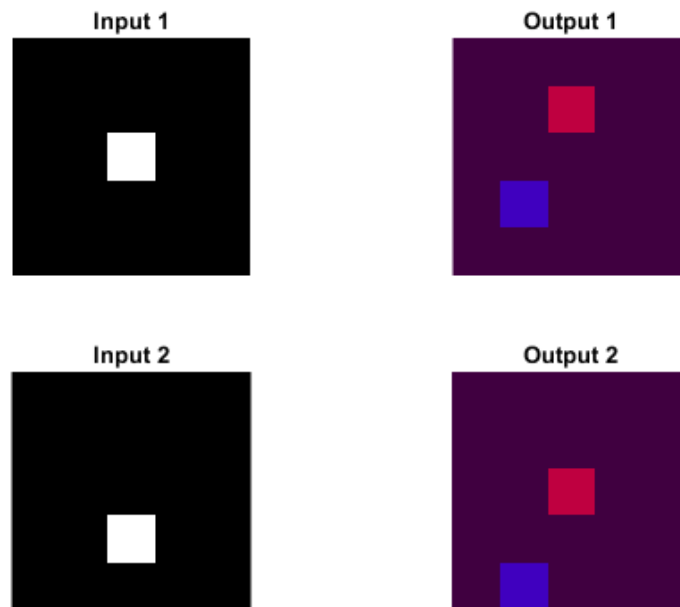
CMPT 412 Lab 1 Report

Zhuo Ning 301421723

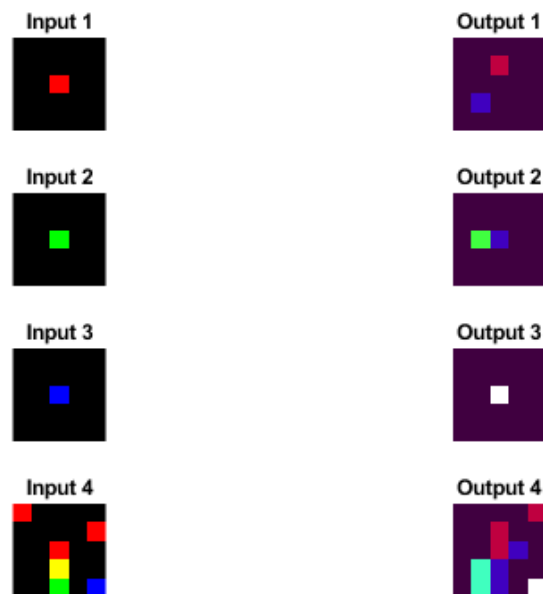
Sept 2021

Q1

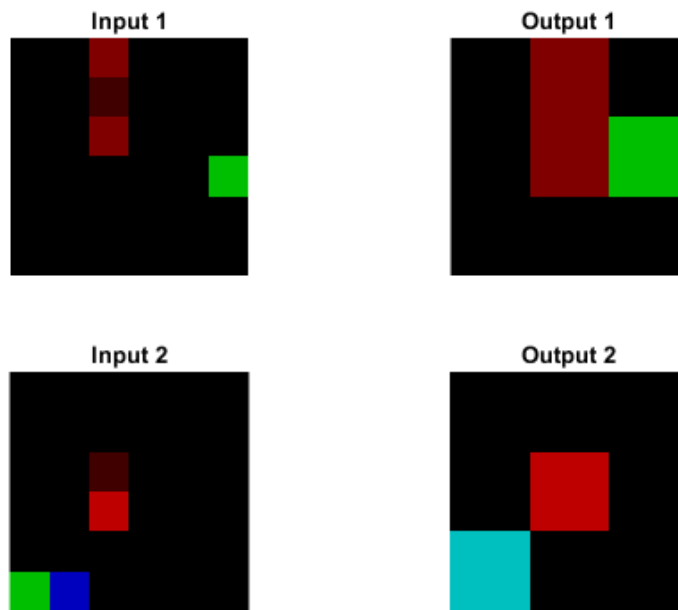
Convolution Test 1



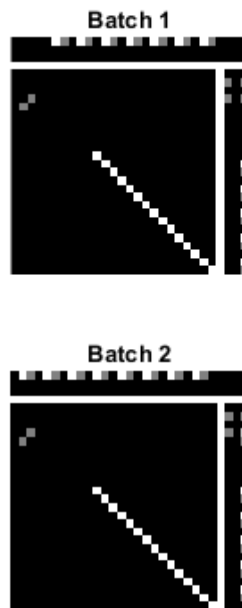
Convolution Test 2



Pooling Test



Inner Product Test



Q3.1

Test Accuracy: 0.97

Q3.2

```
>> test_network
```

	0	1	2	3	4	5	6	7	8	9
0	955	0	3	0	3	5	4	5	3	2
1	0	1120	3	2	0	0	3	1	6	0
2	7	0	982	15	2	1	1	13	11	0
3	0	1	4	970	2	12	0	7	9	5
4	1	0	7	1	945	0	7	2	2	17
5	2	3	1	12	1	855	6	2	8	2
6	9	3	1	1	3	3	935	0	3	0
7	1	8	19	5	1	0	0	982	1	11
8	4	1	6	14	4	5	6	3	926	5
9	7	8	1	11	12	4	1	19	11	935

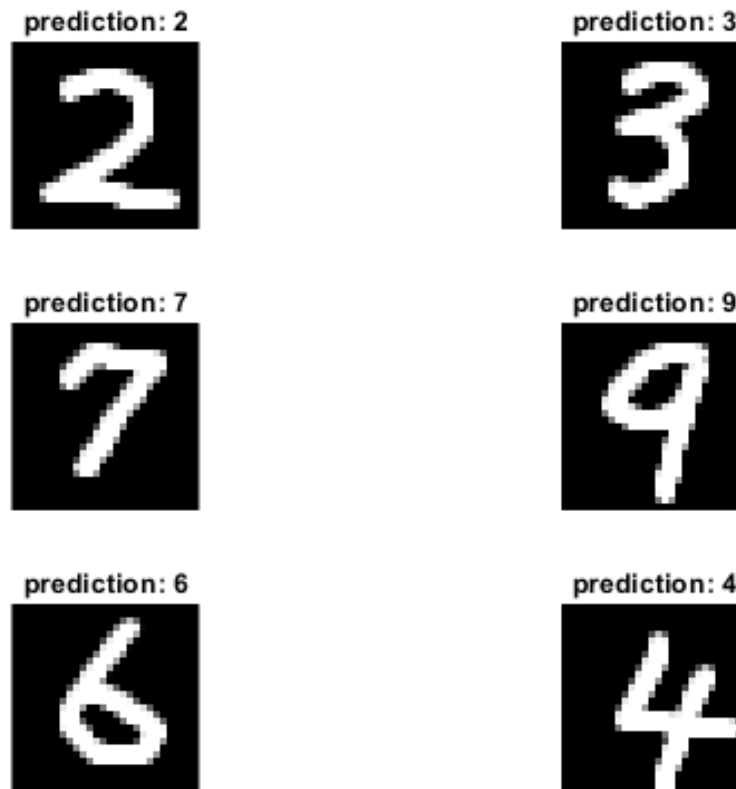
From the full mnist dataset, I generated the confusion matrix above. As showing from the result, there are a few confused pairs, the top 2 are:

1. 7 and 2 (predict 7 as 2: 19 times; predict 2 as 7: 13 times; total 32 times)
2. 7 and 9 (predict 7 as 9: 11 times; predict 9 as 7: 19 times; total 30 times)

For 7 and 2 pair, I think when 2 is written with blur bottom line, even for human it is very hard to distinguish them.

For 7 and 9 pair, basically 7 and 9 has very similar shape. By the nature of LeNet, after each convolution, the line become thicker. Therefore, the top part of 9 to the network might be recognized as a thicker horizontal line.

Q3.3



I found that our conv net is good at recognizing numbers with thick line size. Therefore, I used thick size brush to paint the numbers.

Q4.1

original



conv 20



relu 20



Q4.2

As mentioned in Q3.2, after each convolution, the strokes of the original image become much thicker. In the first convolution layer, we can already see the circle of the 9 is disappearing for some of the kernels. I can see after max pooling and more convolution, the circle might be completely unrecognizable in some of the kernels. That is the reason why many number 9s are predicted as 7. Another very obvious difference is the image become much more 'blur' even though the actual resolution is only reduced to 24 by 24 from 28 by 28. I think both stroke and 'blurriness' come from the nature of how convolution layer works, combining all the pixel in the kernel to one weighted sum. The 'informative' pixels are given more weight and spread out in the feature map, so I think the difference between feature maps and original image represents the feature extraction and generalization.

Q5

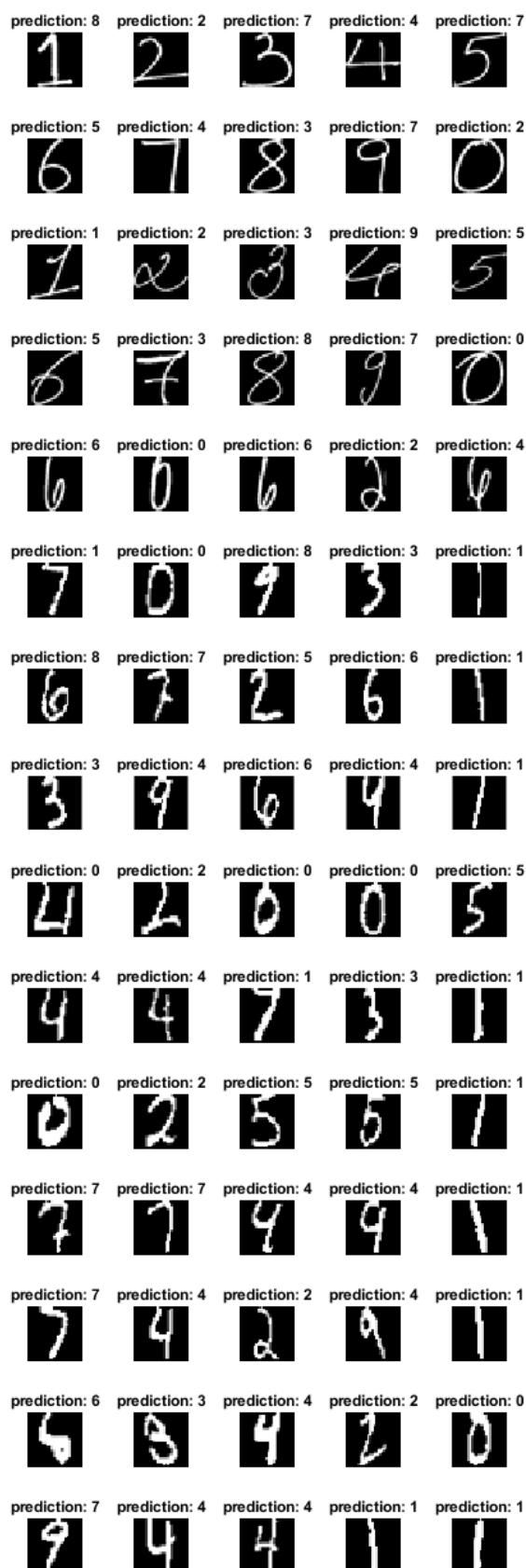


image1.jpg

Correct rate: 2/10

Error rate: 8/10

Correct predictions are: 2, 4

Incorrect predictions are: 1, 3, 5, 6, 7, 8, 9, 0

image2.jpg

Correct rate: 6/10

Error rate: 4/10

Correct predictions are: 1, 2, 3, 5, 8, 0

Incorrect predictions are: 4, 6, 7, 9

image3.png

Correct rate: 5/5

Error rate: 0/5

All numbers are predicted correctly.

image4.jpg

Correct rate: 39/50

Error rate: 11/50

Most of the digits from 0 - 8 were predict correctly, except digit 9s were all predicted incorrectly.