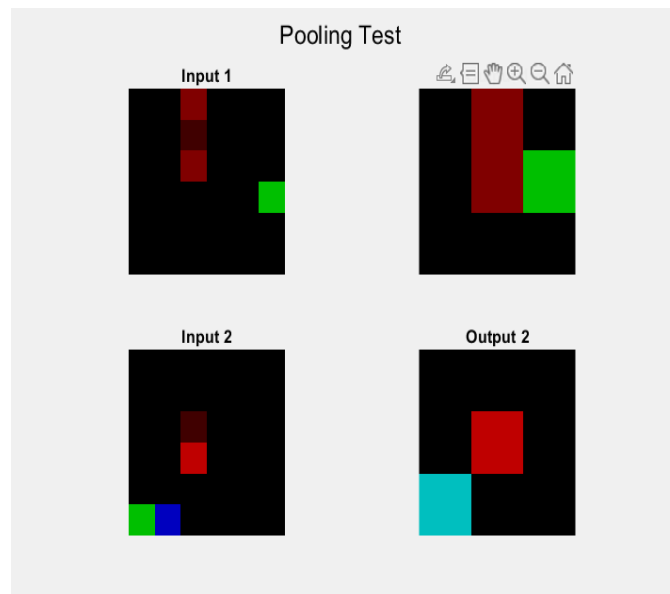
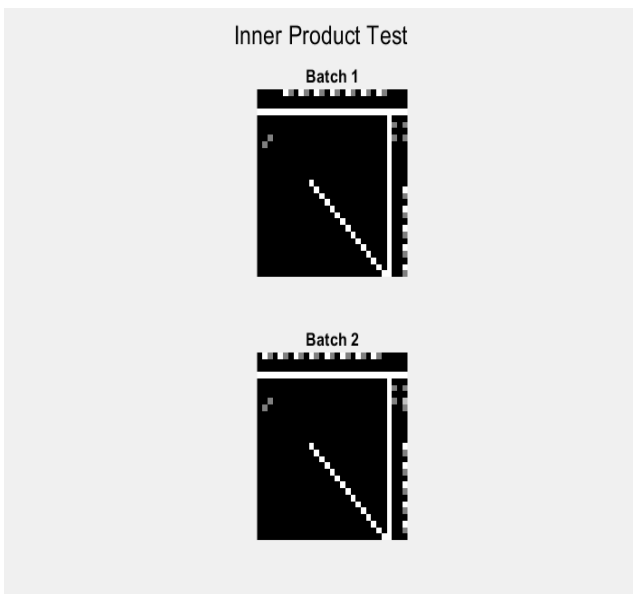
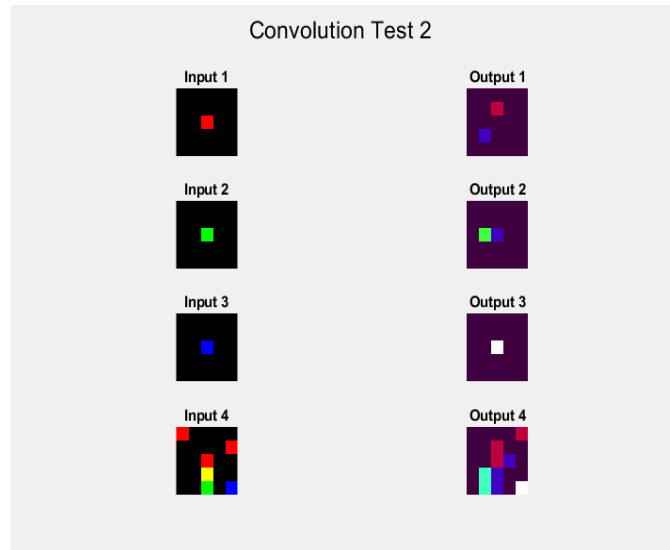
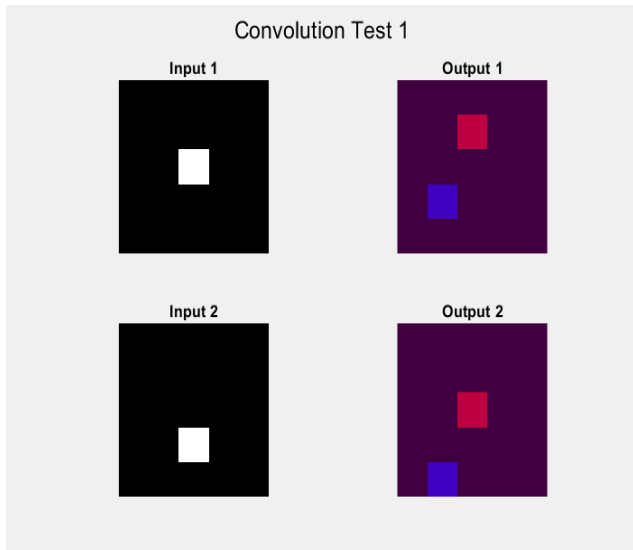


Report

Part 1 Forward Pass



Part3 Training

Q3.1

The accuracy after training is 97%

```
cost = 0.049833 training_percent = 0.990000  
test accuracy: 0.970000
```

Q3.2

The confusion matrix is:

45	0	0	0	1	0	0	0	0	0
0	63	1	0	0	0	0	0	1	0
0	0	47	1	0	0	0	2	0	0
0	0	0	48	0	0	0	0	1	0
0	0	0	1	48	0	0	0	0	1
0	1	0	0	0	43	0	0	3	0
0	0	0	0	1	1	41	0	0	0
1	0	0	0	0	0	0	46	1	0
0	0	0	3	0	0	0	0	47	0
1	2	0	0	0	0	0	3	0	46

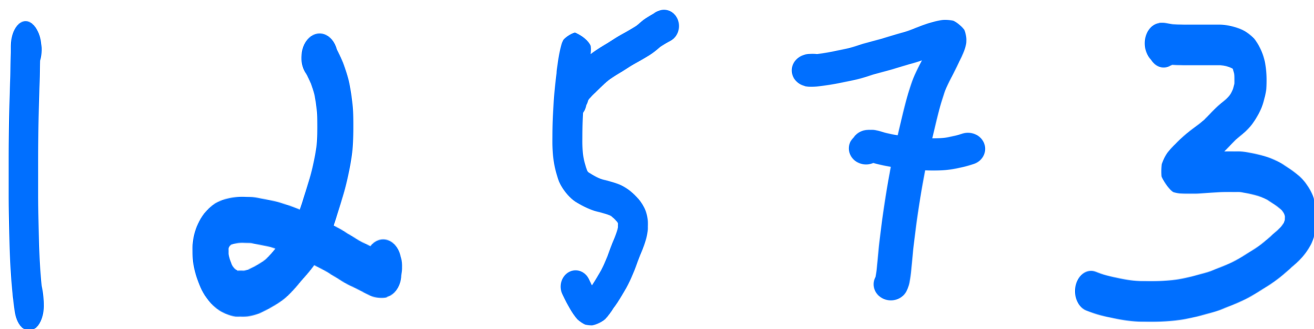
The top 2 confused pairs are (2,2) and (4,4) or (5,5)

The confusing here actually stands for matching count between the ground truth number and the prediction number

Therefore, number 1 and number 3/4 are the two highest matching pairs.

The reason that they are such "confusing" might be there are more such number training sample in the training set. Also the shape of these numbers are more unique than others. That's the two main reasons I think that they get well predicted.

Q3.3



GT: 1
predict: 1

GT: 2
predict: 1

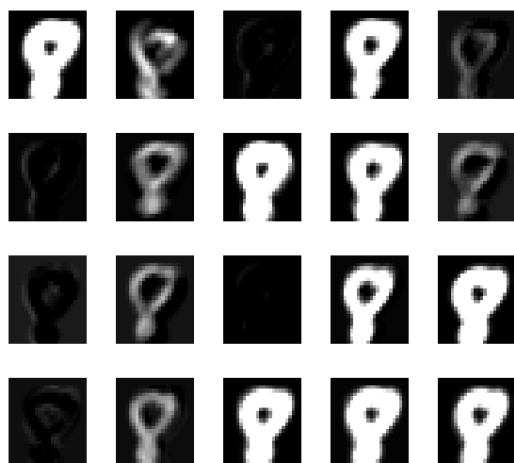
GT: 5
predict: 5

GT: 7
predict: 3

GT: 3
predict: 3

Part 4 Visualization

Q4.1



feature map of the CONV layer



feature map of the ReLU layer

Q4.2



original input

These features in feature maps comparing to the original figure all look more blurry. The reason cause that is in both feature map there are always some extra bright or dark pixels surrounding the original number. And that is due to the different weights in the different kernels. While these 2 feature maps are actually the same since negative pixel equal to 0 value pixel in visualization.

Part5 Image Classification

final estimation result:

```
>> ec
The prediction numbers for the first image are:
  1   2   3   4   5   6   7   8   9   7

The prediction numbers for the second image are:
  1   6   3   9   5   5   7   8   7   0

The prediction numbers for the third image are:
  6   0   6   2   4

The prediction numbers for the fourth image are:
Columns 1 through 19
  7   0   7   7   6   7   2   6   1   3   9   6   4   1   4   2   0   0   5

Columns 20 through 38
  4   4   7   1   1   9   2   5   5   1   7   7   4   9   1   7   4   2   9

Columns 39 through 50
  1   4   5   4   0   2   9   7   4   4   1   1

fx >> |
```

Error recognition: (GT->prediction)

image 1:

0 -> 7

image 2:

2->6 4->9 6->5 9->7

image 3:

none

image 4: accuracy(90%)

9->7 3->7 3->1 0->9 3->5

Note: The actual order of the sub images in image 4

following the cropping order given by function *regionprops()*

To visualize the sub image order, uncomment the line 132 to plot each sub image in order.