CMPT 412
Assignment 5 Report (1 Late Day Used)

Maoshun Shang

301280479

Introduction

This assignment mainly focuses on the technique studied in lectures, including basics about fully connected layer, max pooling, convolutional layer, and back propagation etc. The programming language used is MATLAB, which helps with reducing the complexity of heavy computation on matrix and vector calculations.

Part 3: Training

3.1 Training

The below shows the test accuracy after training of 3000 iterations. The test accuracy is 97%.

```
test accuracy: 0.970000
```

3.2 Test the Network

The Confusion Matrix is:

```
Confusion Matrix:
    63     0     0     0     0     0     1     0     0     0
     0    54     0     0     0     0     0     0     0     0
     0     0    34     1     0     0     0     1     1     0
     0     0     0    52     0     2     0     0     0     0
     0     0     0     0    53     0     2     0     0     1
     0     0     0     1     0    45     0     1     1     0
     0     0     0     0     0     0    46     0     0     0
     0     0     1     0     0     0     0    38     0     0
     0     0     0     0     1     1     0     0    46     0
     0     0     0     0     1     0     0     2     1    50
```

The top two confuse pairs of classes are:

- 2 test cases that the actual input is class 10, but predicted as class 8
- 1 test cases that the actual input is class 10, but predicted as class 9

We can see that most of the entries settles along the diagonal, meaning the prediction accuracy will be very high. The two pairs of the most confused classes are considered as random errors under such scenario. Therefore, we can simply ignore the off-diagonal entries and conclude the model is very accurate.

In addition, if we output the accuracy of the model, it shows:

```
>> accuracy= sum(diag(confusion))/sum(confusion,'all');
>> accuracy

accuracy =

    0.9620
```

An accuracy of 96.2%, which is considered as good performance.

3.3 Real-world Testing

The top row shows my handwritten digits, while the lower part shows my output, including the truth value, predicted value, and the probability of each category.

```
The actual class is: 10
The predicted class is 9
    0.0001    0.0384    0.0316    0.0001    0.0013    0.0011    0.0001    0.0287    0.8876    0.0110

The actual class is: 4
The predicted class is 4
    0.0001    0.0052    0.1724    0.8207    0.0001    0.0001    0.0000    0.0002    0.0001    0.0011

The actual class is: 5
The predicted class is 5
    0.0000    0.0000    0.0000    0.0000    0.8961    0.0000    0.0000    0.0023    0.0000    0.1016

The actual class is: 7
The predicted class is 7
    0.0004    0.0000    0.0078    0.0000    0.0002    0.3669    0.5884    0.0000    0.0359    0.0004

The actual class is: 3
The predicted class is 3
    0.0000    0.0000    0.7244    0.0000    0.0000    0.0000    0.0000    0.2752    0.0000    0.0002
```

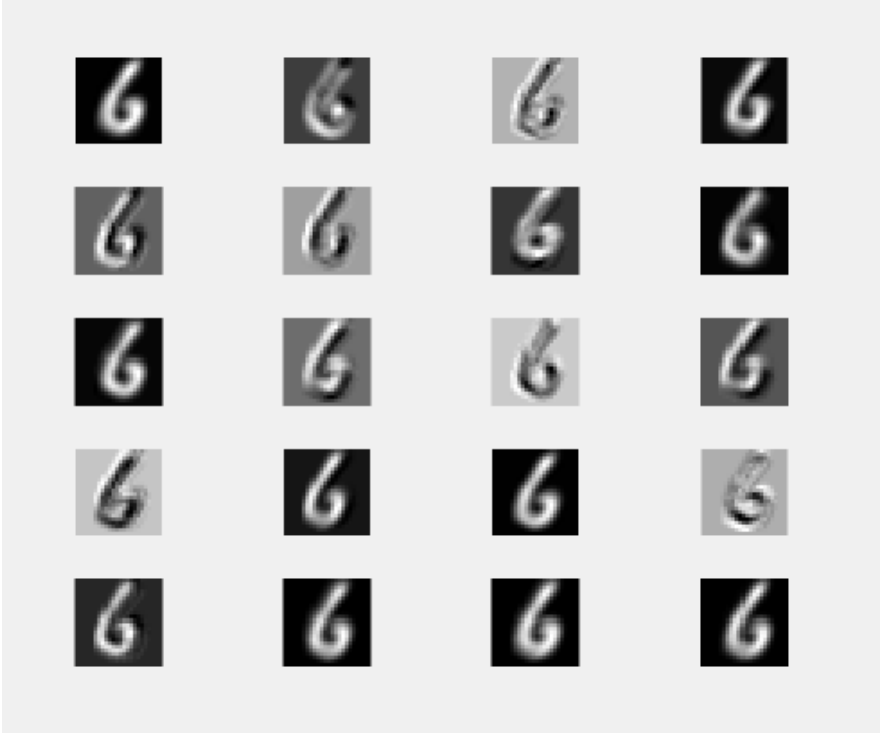
We see that the accuracy is high (80%). Only the digit ’9’ is not correctly predicted.

Because my handwriting is different from that of the training dataset. Therefore, it is reasonable that the model cannot recognize some numbers. Overall, the tests in part 3 indicates the performance of the model is good.

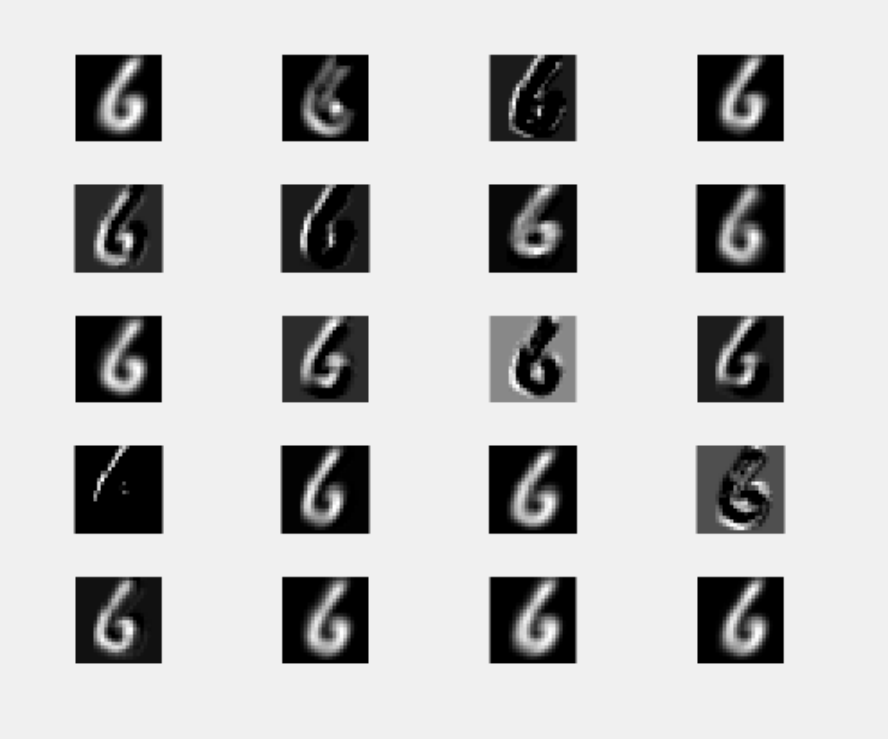
Part 4: Visualization

4.1 Show 20 Images from CONV and ReLU Layers

The subplots below are feature maps from CONV layer



The subplots below are feature maps from ReLU layer



4.2 Compare the Feature Maps and Explain the Difference

The original image is as follows:

While some of the images looks identical, for example, the images at (1,1) from both layers is very close to the original image. However, there are a few images that are different.

Difference 1

The images at (1,4), (3,1), (4,2), (4,3), and the last row from both layers are similar to each other and the original image, but a little bit blurred (like being applied by a gaussian filter).

This is due to in convolutional layer (forward), we take every small portion of the entire input data and apply weight on them, then summing up them and add bias. (see the file 'conv_layer_forward.', line 59). In addition, the blurred effect and the lost sharpness are not from the effect of max pooling layer because the pooling layer is behind the convolutional layer.

Difference 2

The images at (1,3), (3,2), (3,3), (4,1) and (4,4) from the ReLU layer have exactly the opposite colour as the original image: the original digit is white but the images from the ReLU layer is black.

This is because in the ReLU layers' implementation, if the incoming data at (n, m) is less than 0, we simply put a 0 at (n, m). Remember that 0 represent black, this explains the excessive black regions in the ReLU layer's images: the input data at those locations are negative numbers so that ReLU layer output black for them.