Homework 3

Student Number: 118033910019
Name: Xingyi Wang

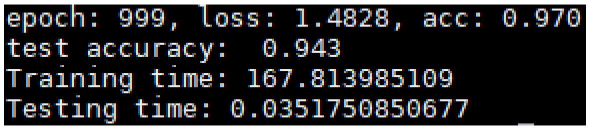
Problem 1. In this assignment, convolutional neural network (CNN) will be used to deal with multi-class classification problems. CNN is a class of deep, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery.

Two problems are given below. The dataset used in this homework is the MNIST database (Modified National Institute of Standards and Technology database), which is commonly used for training and testing in the field of machine learning.

The MNIST database contains 60,000 training images and 10,000 testing images. You need to use the training set to build the ten-category classification model and validate it on the test set.

Solving the ten-class classification problem in the given dataset using feed- forward neural network. You need to finetune your network and only present your best result.

Solution. In this section, I'm going to solve this problem using the Feed-Forward Neural Network (FFNN). This network has three layers, which are the input layer, the hidden layer, and the output layer. The size of input image in MNIST is 28×28, which will be concatenated into a 784-d vector. Then the input layer has 784 inputs. The number of hidden units in the hidden layer is set to 512. And the number of outputs in the output layer is 10, which is equal to the number of labels in MNIST (digit 0 to digit 9). The loss function is cross entropy function. The optimizer is Adam Optimizer. The learning rate is 0.01, and the training process runs for 1000 epochs. The training and testing results are shown below:



```
epoch: 999, loss: 1.4828, acc: 0.970
test accuracy:  0.943
Training time: 167.813985109
Testing time: 0.0351750850677
```

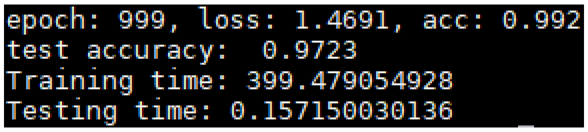
Figure 1: Caption

The training accuracy is 97.0%, and the testing accuracy is 94.3%. The training time is about 168s, and the testing time is about 35.2ms.

Problem 2. Solving the ten-class classification problem using CNN.

1. You need to implement LeNet[1] and use it to solve this problem.
2. Compare the results and training time with problem 1.
3. Visualize the deep features which can be extracted before feed-forward layers, and discuss the results.

Solution. In this section, I'm going to solve this problem using LeNet, which is an CNN model which has two convolution layers followed by max pooling layers and three fully-connected layers. The first convolution layer has six 5×5 kernels (6 channels). The first max pooling layer downsamples the 28×28 matrix to 14×14. The second convolution layer has sixteen 5×5 kernels (16 channels). The second max pooling layer downsamples the 14×14 matrix to 7×7. So all of the sixteen 7×7 matrices are flattened into a 784-d vector which is input into the following fully-connected layers. These layers have 128, 84, 10 units. The loss function is cross entropy function. The optimizer is Adam Optimizer. The CNN is more complex than FFNN, so it is harder to comverge. So the learning rate is set to 0.005. And the training process runs for 1000 epochs. The training and testing results are shown below:



```
epoch: 999, loss: 1.4691, acc: 0.992
test accuracy:  0.9723
Training time: 399.479054928
Testing time: 0.157150030136
```

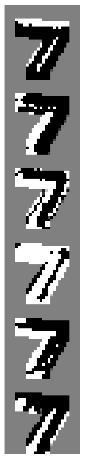
Figure 2: Caption

The training accuracy is 99.2%, and the testing accuracy is 97.3%. The training time is about 399s, and the testing time is about 157ms. We can see that the training time and testing time are both higher than that of FFNN. This indicates that CNN achieves higher accuracy with the cost of both higher training and testing time.

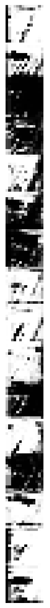
The visualization results of two testing samples are shown below. I use tensorflow to extract the deep features from two convolution layers. The convolution layers are responsible for the feature extraction, so the visualization results of them can show the prediction mechanism of the CNN model.

Fig. 3 shows a handwritten digit *7* which is correctly predicted by the CNN model. In these figures, a darker pixel means a larger value of the corresponding number in the feature matrix. We can see that the feature from the first convolution layer (conv1) is very similar to the digit 7. In the second convolution layer, we can still vaguely see the outline of digit 7. Fig. 4 shows another sample of digit *2*, and the patterns are the same. There are more samples in the folder *fig*.

(a)
img



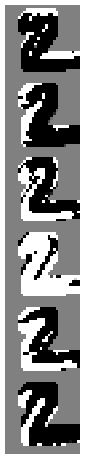
(b)
conv1



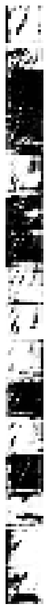
(c)
con2

Figure 3: handwritten digit 7

(a)
img



(b)
conv1



(c)
con2

Figure 4: handwritten digit 2