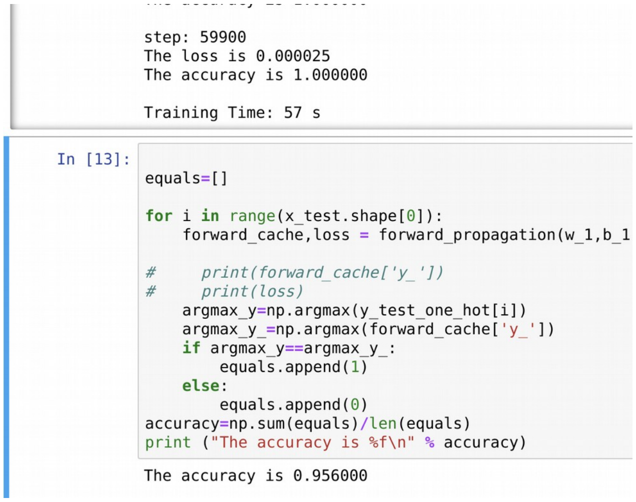
Xin Yang
xy213

HW3
Q1:
For hand written part, I select the mean square error as the loss function. This program runs on a stochastic gradient descent optimizer. The learning rate is 0.01 and number of epochs is 5. The test set accuracy achieves 95.6%, which takes 57 seconds to complete.



```
step: 59900
The loss is 0.000025
The accuracy is 1.000000

Training Time: 57 s
```

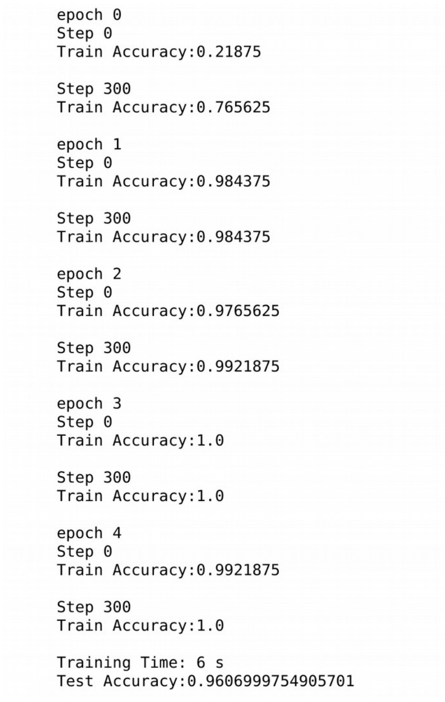
```
In [13]:
equals=[]

for i in range(x_test.shape[0]):
    forward_cache,loss = forward_propagation(w_1,b_1

#     print(forward_cache['y_'])
#     print(loss)
    argmax_y=np.argmax(y_test_one_hot[i])
    argmax_y_=np.argmax(forward_cache['y_'])
    if argmax_y==argmax_y_:
        equals.append(1)
    else:
        equals.append(0)
accuracy=np.sum(equals)/len(equals)
print ("The accuracy is %f\n" % accuracy)

The accuracy is 0.956000
```

Q2:
For the framework based solution, I utilize the TensorFlow with GPU support. The loss function is the cross entropy. My algorithm runs on a mini-batch stochastic gradient descent optimizer. The learning rate is 0.01, batch size is 128 as suggested. By running 5 epochs, the overall accuracy on the test data set is over 96%, and the training duration takes 6 seconds.



```
epoch 0
Step 0
Train Accuracy:0.21875

Step 300
Train Accuracy:0.765625

epoch 1
Step 0
Train Accuracy:0.984375

Step 300
Train Accuracy:0.984375

epoch 2
Step 0
Train Accuracy:0.9765625

Step 300
Train Accuracy:0.9921875

epoch 3
Step 0
Train Accuracy:1.0

Step 300
Train Accuracy:1.0

epoch 4
Step 0
Train Accuracy:0.9921875

Step 300
Train Accuracy:1.0

Training Time: 6 s
Test Accuracy:0.9606999754905701
```