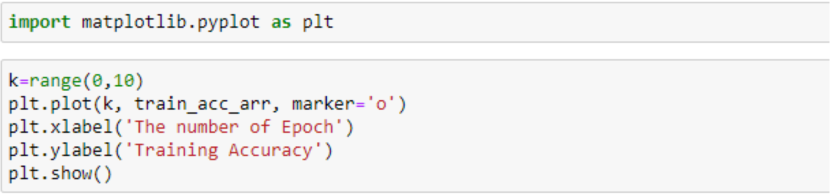
In this project part3, you are required to understand the whole process of compiling different layers (Convolutional Layer, Fully-Connected Layer, Pooling Layer, Activation Layer, Loss function) of a simple Convolutional Neural Network (CNN) for the visual classification task. And you need to compile your own evaluation code to evaluate the trained CNN to obtain the training and testing results.

Similar to Part1, I used the dataset from the MNIST. The demo code will randomly select four different categories and 500 training and 100 testing samples for each category. Therefore, the total size of the training and testing samples is 2000 and 400 respectively.
The batch size for training is 100, so I need to train 100 batch for 20 times. The total epoch number should be 10 and the learning rate should be 0.001.

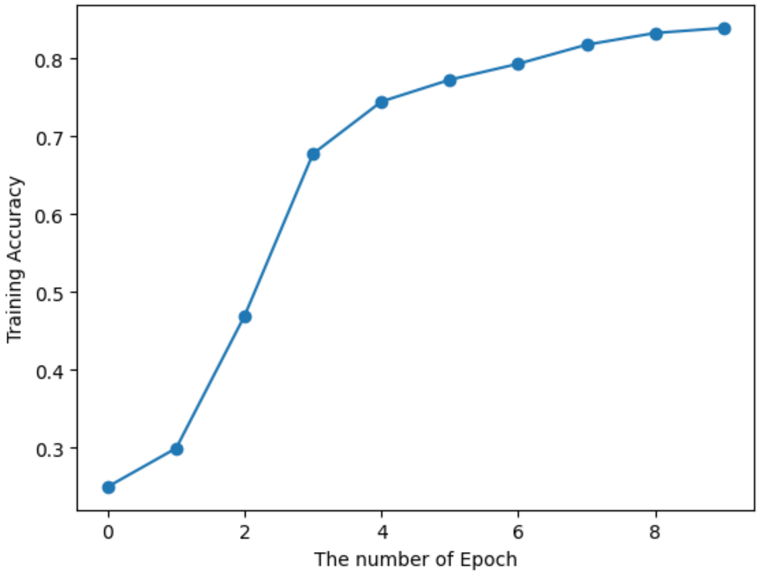
Four values, final training accuracy, training loss, testing accuracy and testing loss after 10 epochs.

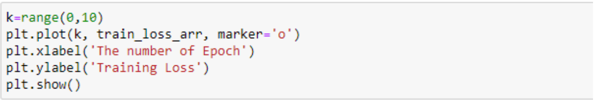
The following four plots show training accuracy vs epochs, training loss vs epochs, testing accuracy vs epochs, and testing loss vs epochs.



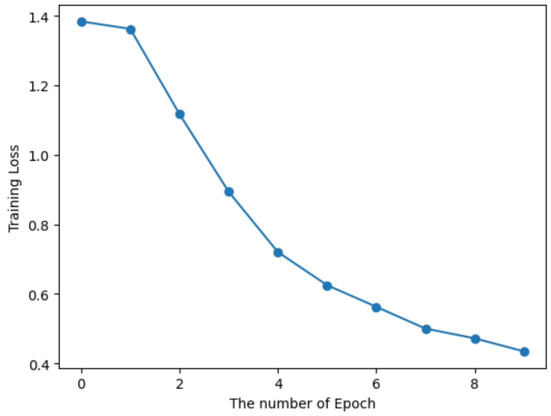
```
import matplotlib.pyplot as plt
```

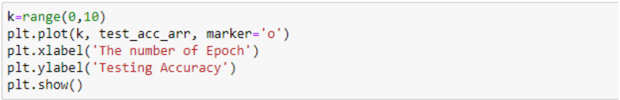
```
k=range(0,10)
plt.plot(k, train_acc_arr, marker='o')
plt.xlabel('The number of Epoch')
plt.ylabel('Training Accuracy')
plt.show()
```



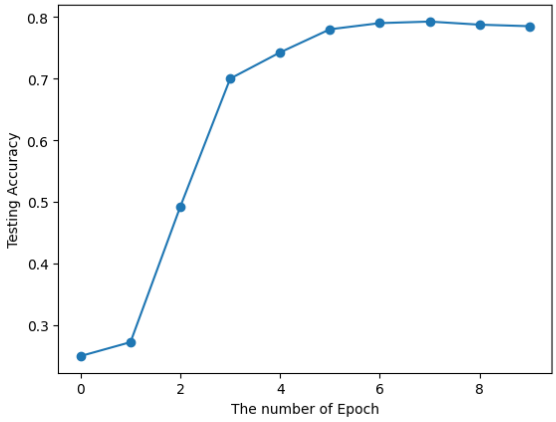


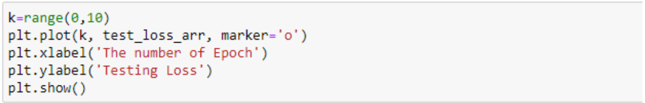
```
k=range(0,10)
plt.plot(k, train_loss_arr, marker='o')
plt.xlabel('The number of Epoch')
plt.ylabel('Training Loss')
plt.show()
```



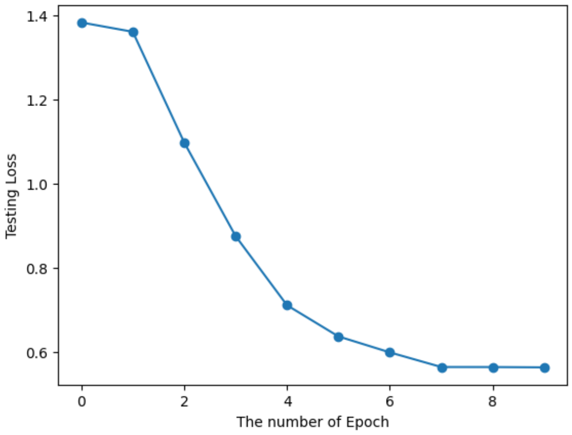


```
k=range(0,10)
plt.plot(k, test_acc_arr, marker='o')
plt.xlabel('The number of Epoch')
plt.ylabel('Testing Accuracy')
plt.show()
```





```
k=range(0,10)
plt.plot(k, test_loss_arr, marker='o')
plt.xlabel('The number of Epoch')
plt.ylabel('Testing Loss')
plt.show()
```

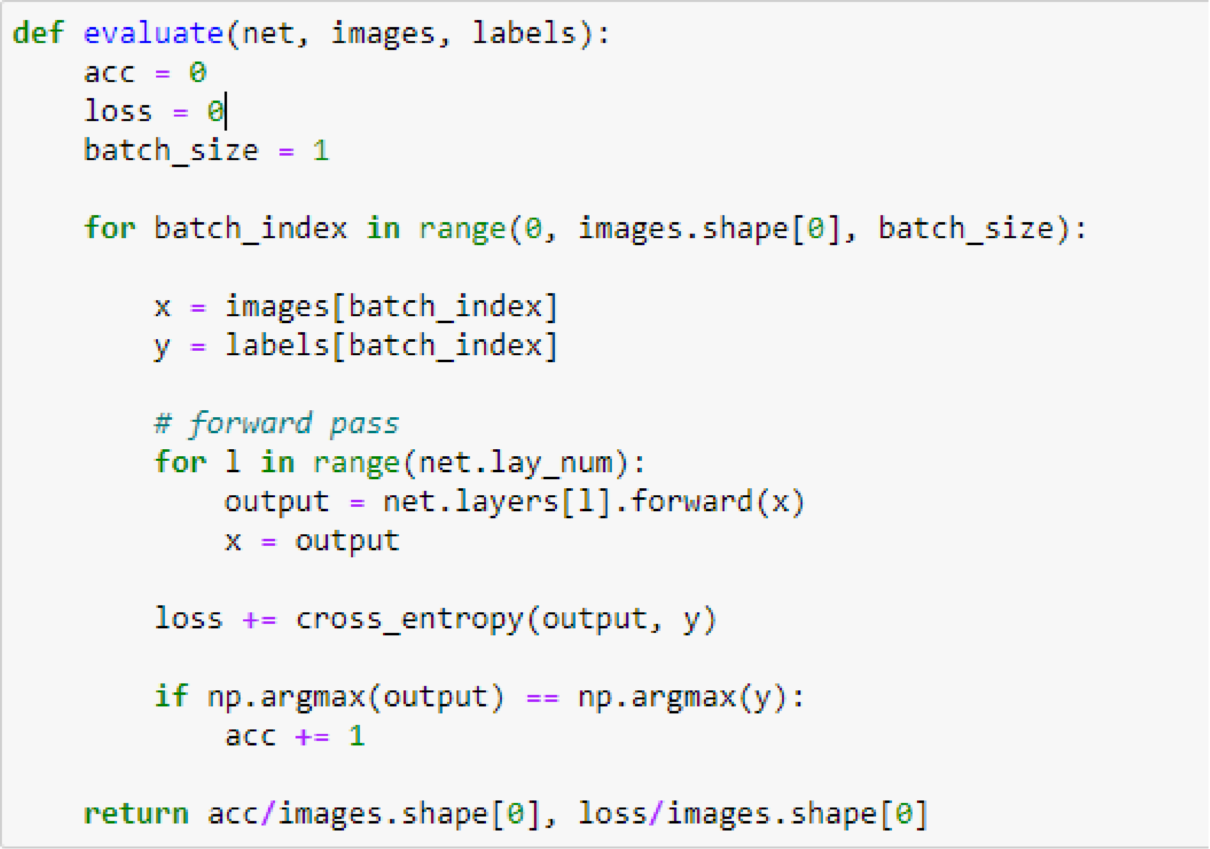


From the plots, I figure out that both training and testing accuracy is increasing as training set (Epoch) is increasing. And both training and testing loss is decreasing as training set (Epoch) is increasing. Also, I found the plots are saturating at 7 or 8 Epochs for both training and testing. For deep learning, training accuracy is increasing as training more (increasing Epochs). However, testing accuracy is saturating and even decreasing at 8 or 9 Epochs. Therefore, we should stop training set at 8 or 9 Epochs, in order to prevent overfitting.

Here are final four values that were submitted in “Quiz” section of Project 3.

```
=== Epoch:9 Train Size:2000, Train Acc:0.839, Train Loss:0.435 ===
=== Epoch:9 Test Size:400, Test Acc:0.785, Test Loss:0.564 ===
```

The code for the evaluate() function



```
def evaluate(net, images, labels):
    acc = 0
    loss = 0
    batch_size = 1

    for batch_index in range(0, images.shape[0], batch_size):

        x = images[batch_index]
        y = labels[batch_index]

        # forward pass
        for l in range(net.lay_num):
            output = net.layers[l].forward(x)
            x = output

        loss += cross_entropy(output, y)

        if np.argmax(output) == np.argmax(y):
            acc += 1

    return acc/images.shape[0], loss/images.shape[0]
```