

## Homework-3: MNIST Classification with ConvNet

**1.Record the training and testing accuracy, plot the training loss curve and training accuracy curve in the report.**

### My Results:

#### Training:

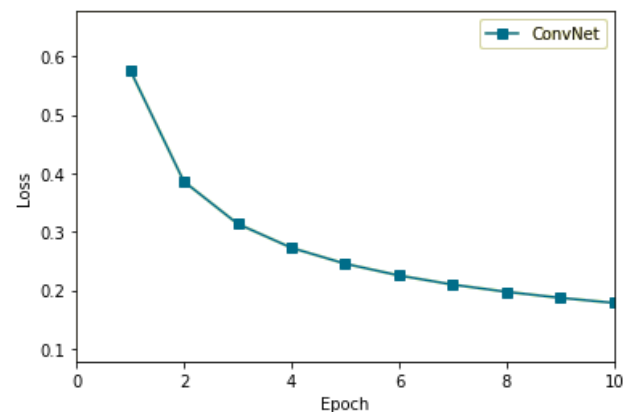
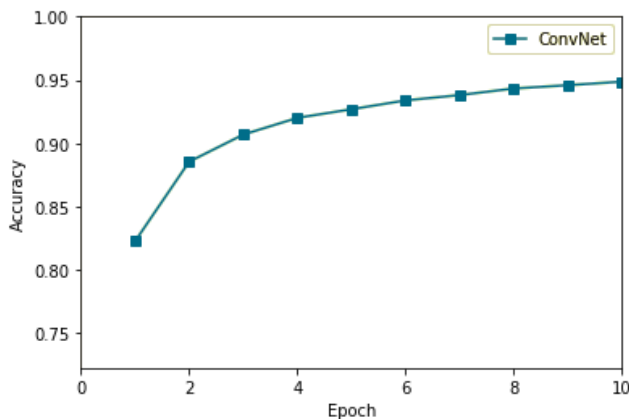
```
Epoch [9]          Average training loss 0.2382   Average training accuracy 0.9281
Epoch [9]          Average validation loss 0.1781   Average validation accuracy 0.9486
```

#### Testing:

Testing...

The test accuracy is 0.9341.

#### Plots:



**2.Compare the difference of results when using ConvNet and MLP (done in homework-2) (you can discuss the difference from the aspects of training time, convergence and accuracy).**

When I compared my HW2 and HW3 I was able to observe that even though my HW2 training time was faster in most MLP tasks. The results of some MLP tasks were more accurate and faster such as my: MLP with Euclidean Loss and ReLU Activation Function performed with an accuracy rate of 0.9656. We cannot really say if really MLP or ConvNet is better. I think it all depends on what kind of task are we doing and how accurate and clean is our software implementation.

**3.The given hyperparameters may be performed not very well. You can modify the hyperparameters by your own, and observe how does these hyperparameters affect the classification performance. Write down your observation and record these new results in the report.**

I didn't need to change the hyperparameters as I think my program performed well enough with a 0.9341 accuracy.

**4.Implement the Dropout layer and use it to train the network again. Choose the dropout rate and the location of dropout layer on your own, and observe the effects.**

**My results:**

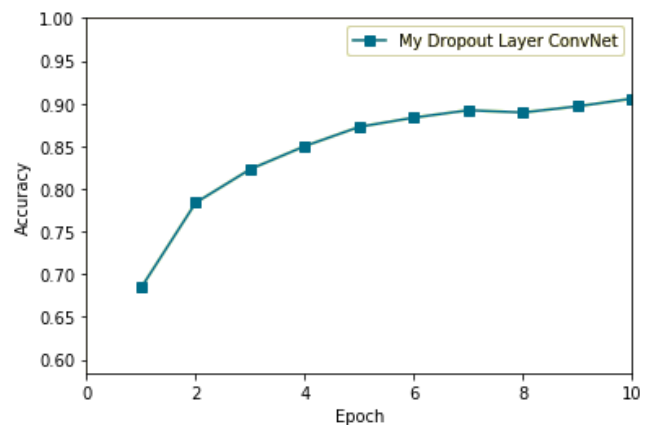
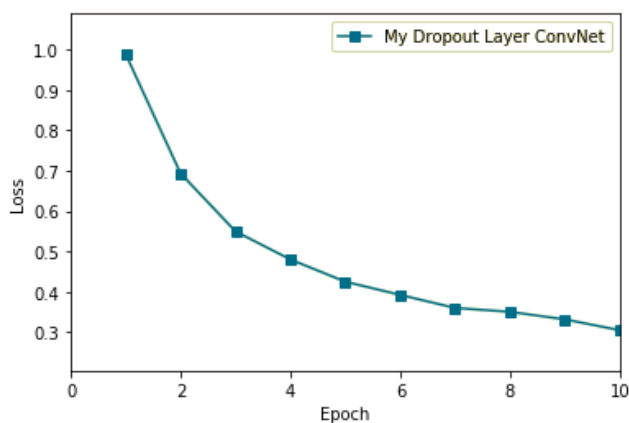
**Training:**

Epoch [9]	Average training loss 0.4022	Average training accuracy 0.8754
Epoch [9]	Average validation loss 0.3046	Average validation accuracy 0.9056
Epoch [9]	Average training loss 0.4022	Average training accuracy 0.8754
Epoch [9]	Average validation loss 0.3046	Average validation accuracy 0.9056

**Testing:**

Testing...  
The test accuracy is 0.8885.

**Plot:**



To observe when I implemented one dropout layer is gave me an accuracy rate of 0.9218 but when added a second dropout layer the accuracy rate dropped significantly to 0.8885 as you can see the results above maybe putting the dropout layer in an appropriate place with one dropout layer in our network would have outperformed our 0.9341 but I wasn't able to find any.