

# Assignment 1 Writeup

Name: Siyuan Li  
GT Email: [siyuan.li@gatech.edu](mailto:siyuan.li@gatech.edu)

# Two-Layer Neural Network

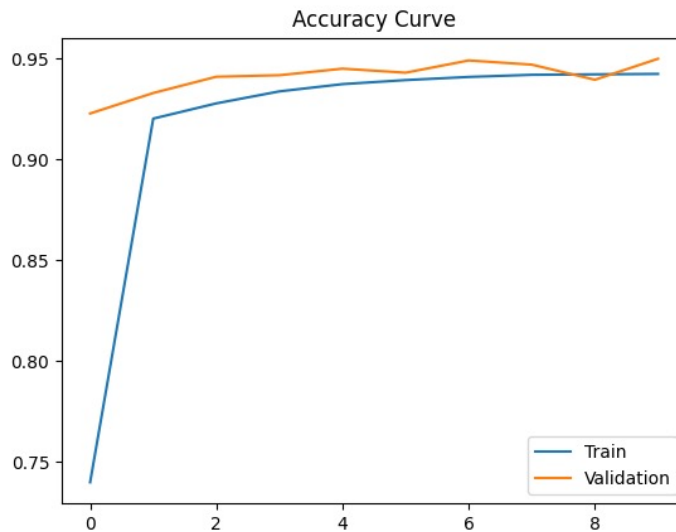
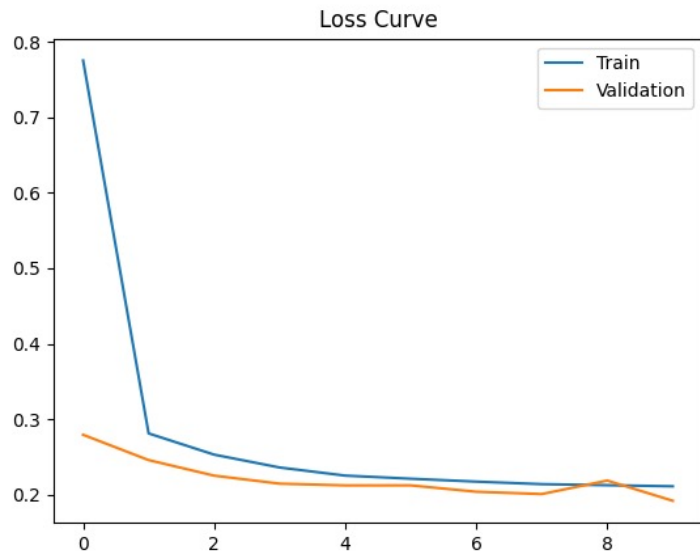
# 1. Learning Rates

Tune the learning rate of the model with all other default hyper-parameters fixed.  
Fill in the table below:

	lr=1	lr=1e-1	lr=5e-2	lr=1e-2
Training Accuracy	0.9425	0.9239	0.9096	0.7293
Test Accuracy	0.9498	0.9221	0.9139	0.7592

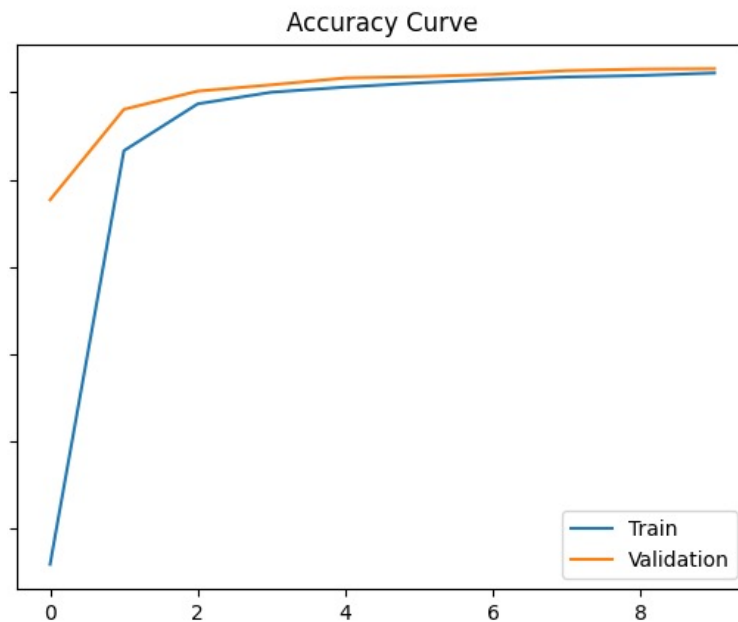
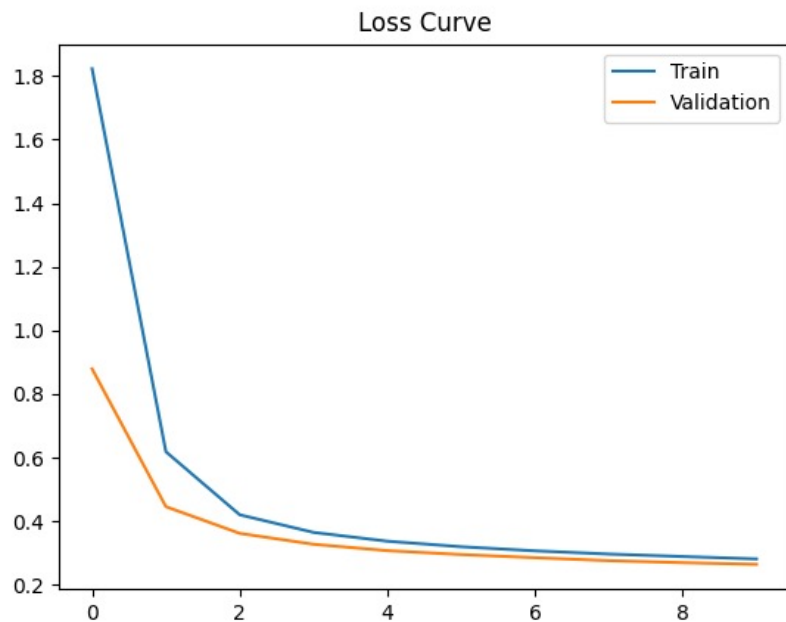
# 1. Learning Curve

Learning rate=1



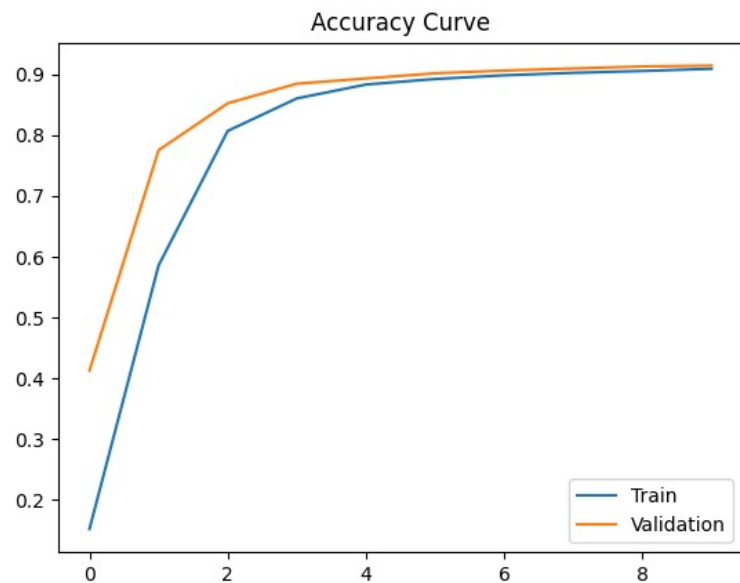
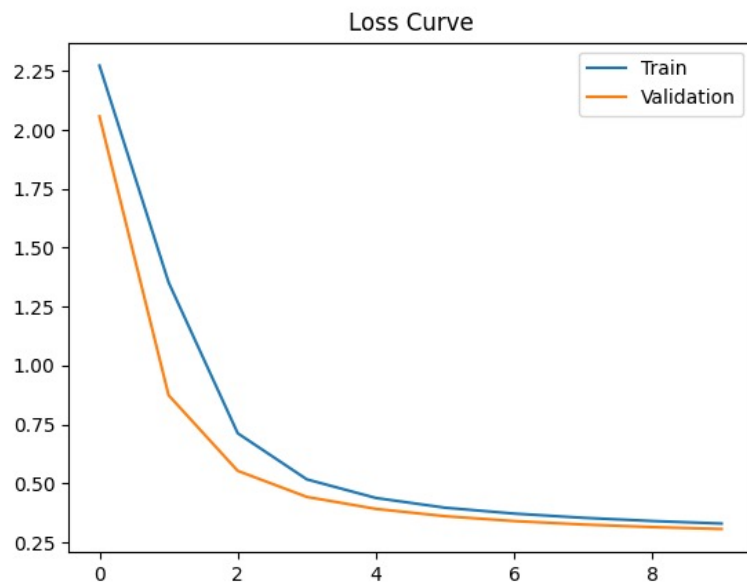
# 1. Learning Curve

Learning rate=0.1



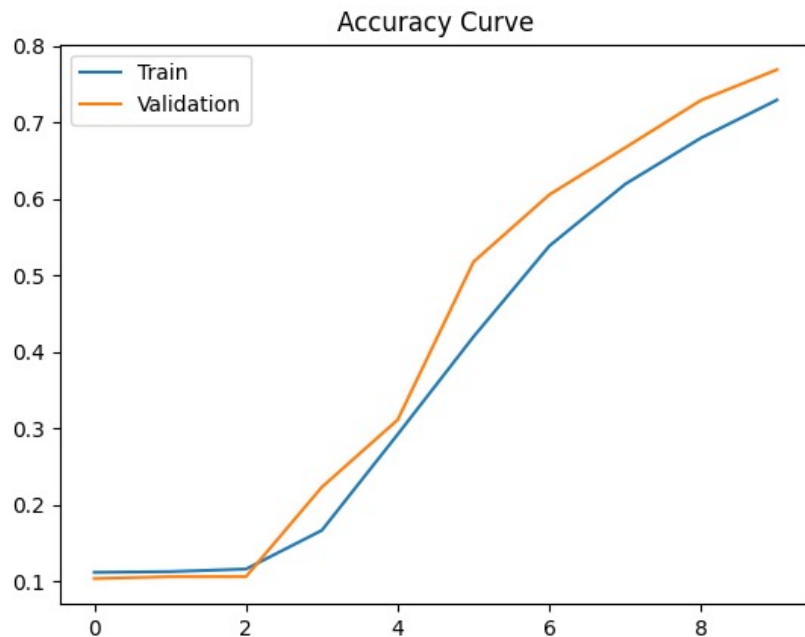
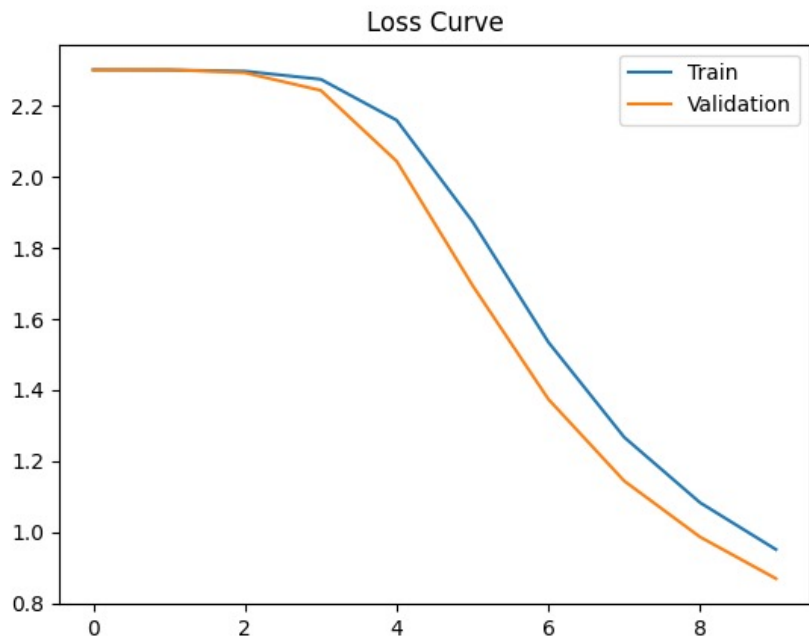
# 1. Learning Curve

Learning rate=0.05



# 1. Learning Curve

Learning rate=0.01



# 1. Learning Rates Findings and Explanation

1. From the numbers, we can easily find that with the decrease of learning rate, the accuracy rate decrease steadily. When learning rate is 1, the test accuracy can reach ~95% while when learning rate is 0.01, the test accuracy is only ~75%. I think this is because given a specific epochs, a large enough learning rate can approach the point quicker. However, I do notice that when learning rate goes from 0.1 to 1, the loss curve shows some divergence at epoch 8-9. I think this is because too large learning rate will cause the jumping over from the solution as the step is too large to catch the correct answer.
2. From the curves, we can find that the loss curve for 0.01 shows very slow tendency to reduce during the first few epochs. I think it is because the loss curve is too small to capture the previous learning and reflect it in the new epoch. Suppose we have larger epoch, a smaller learning curve may also performs, but certainly with a much longer running time.



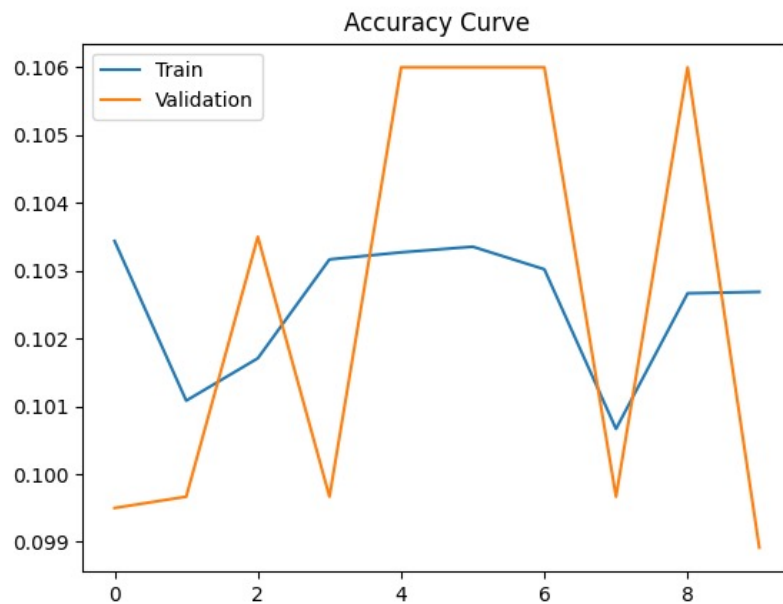
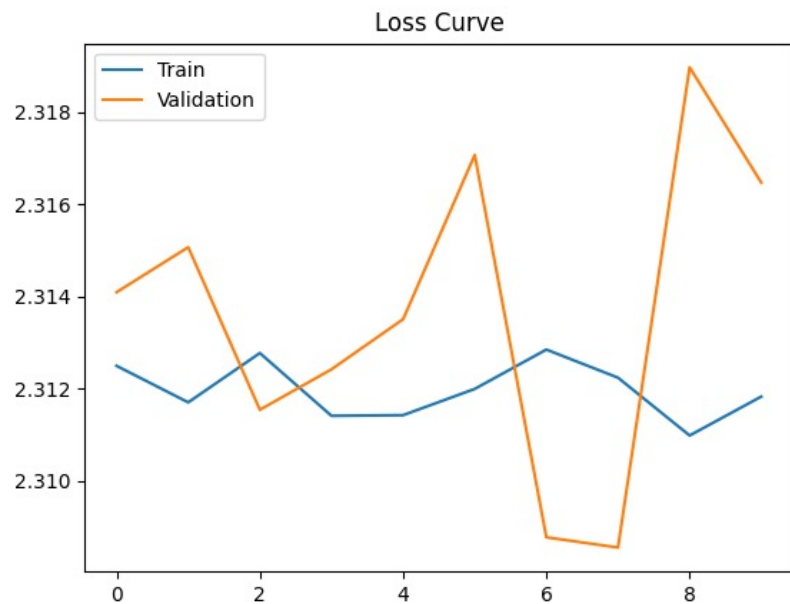
## 2. Regularization

Tune the regularization coefficient of the model with all other default hyperparameters fixed. Fill in the table below:

	alpha=1	alpha=1e-1	alpha=1e-2	alpha=1e-3	alpha=1e-4
Training Accuracy	0.1027	0.3356	0.8855	0.9226	0.9302
Validation Accuracy	0.0989	0.3783	0.8947	0.9273	0.9347
Test Accuracy	0.1135	0.3766	0.8915	0.9250	0.9326

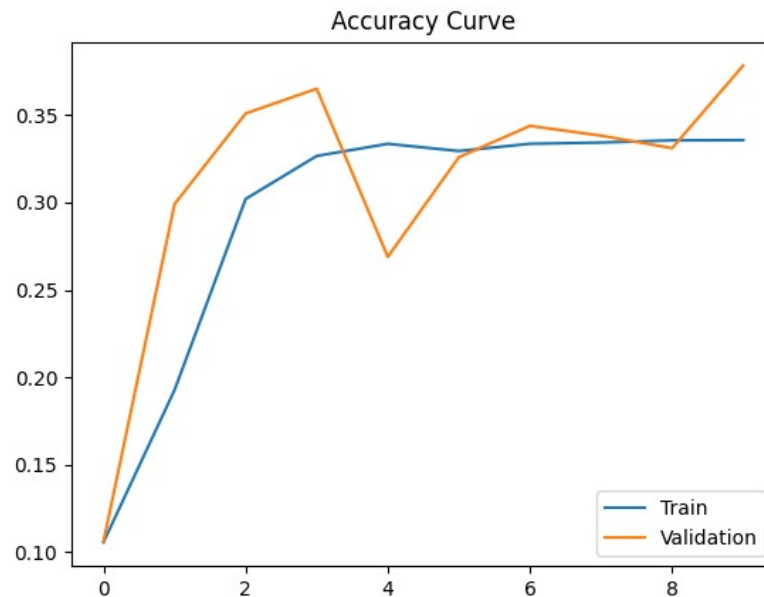
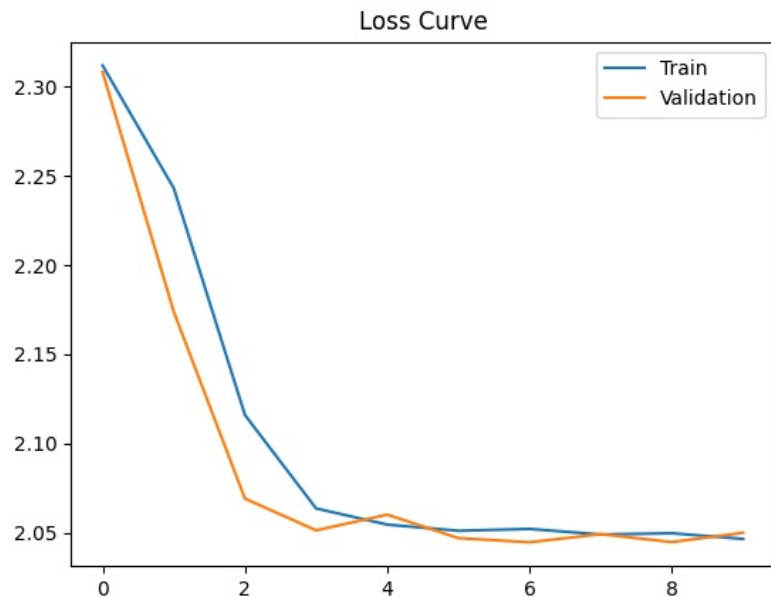
## 2. Regularization

alpha=1 learning\_rate=0.1



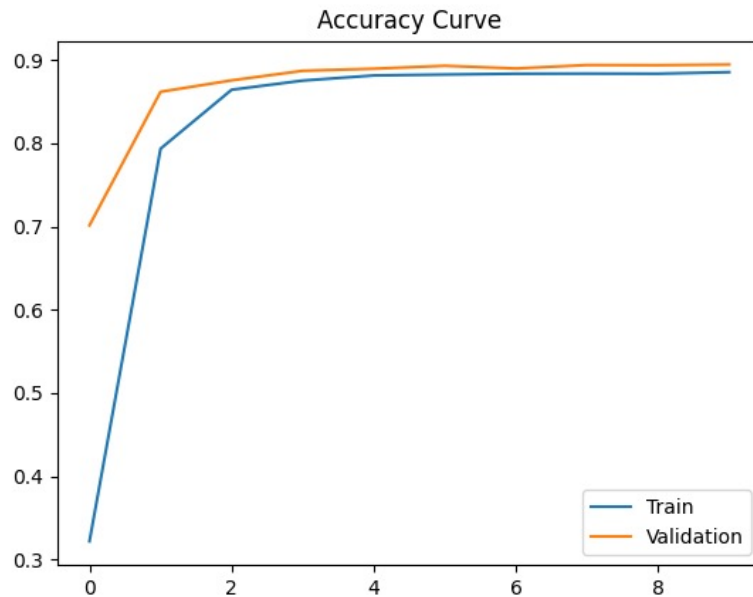
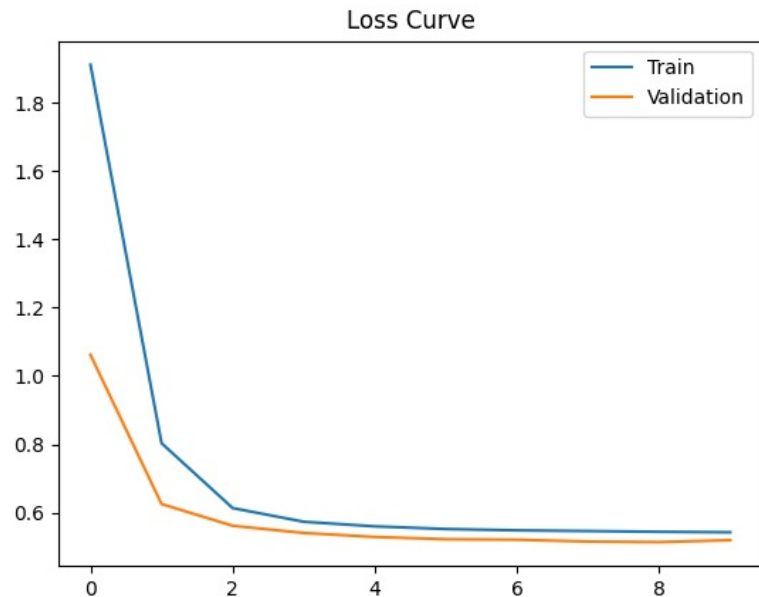
## 2. Regularization

alpha=0.1 learning\_rate=0.1



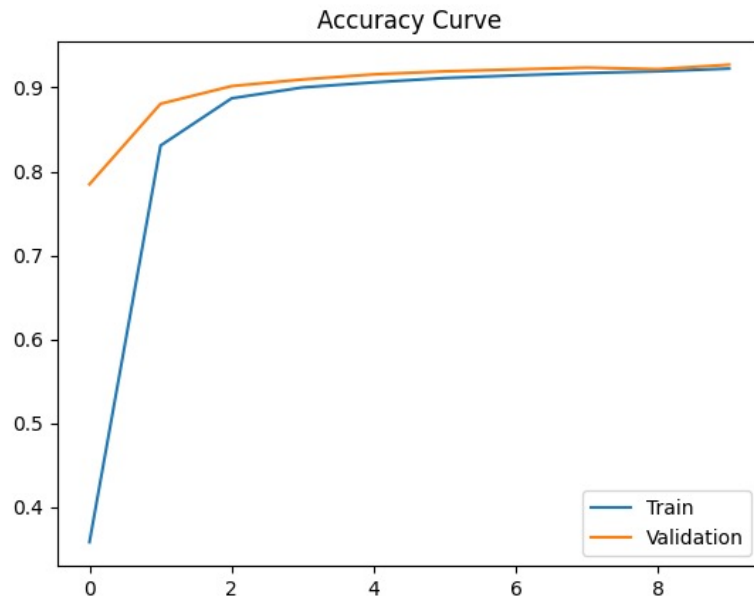
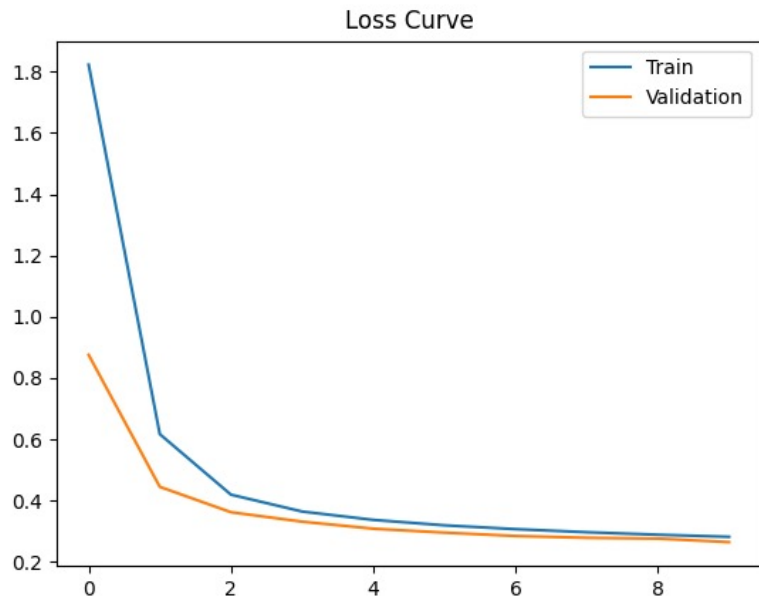
## 2. Regularization

alpha=0.01 learning\_rate=0.1



## 2. Regularization

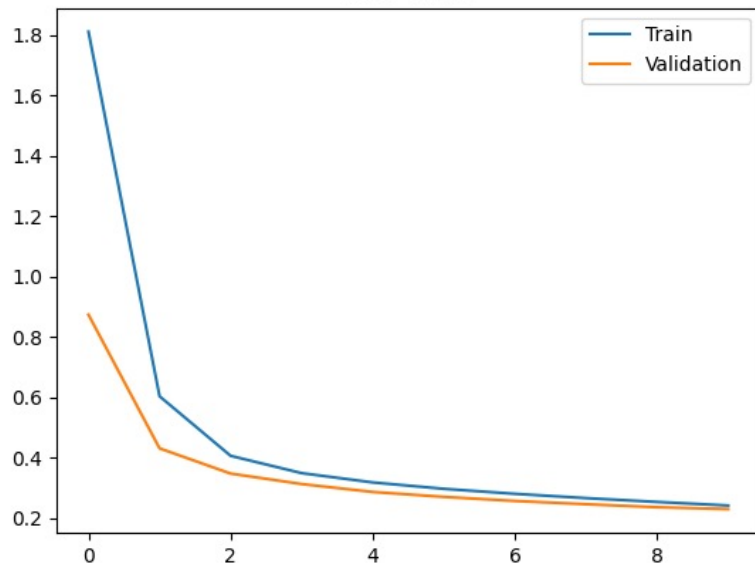
alpha=0.001 learning\_rate=0.1



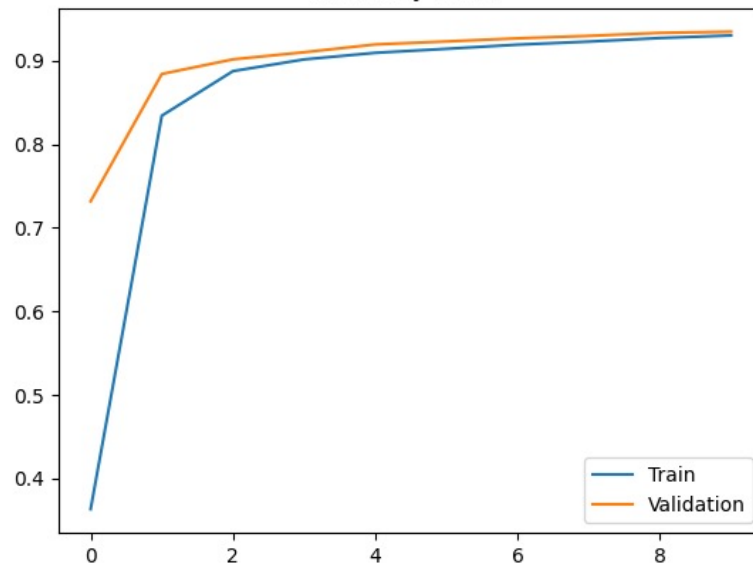
## 2. Regularization

alpha=0.0001 learning\_rate=0.1

Loss Curve



Accuracy Curve



## 2. Regularization Findings and Explanation

1. From the numbers, we can find that the accuracy rate increase while the alpha decreases. While the accuracy rate is only ~10% when the alpha is 1, it can reach ~95% when alpha is 0.0001. (All experiment holds learning rate to 0.1 in this part). As the optimal point is reached by find the value that can minimize cost function, an increasing regularization parameter will force the optimization function to choose a smaller theta. In turn, given a certain epoch, it will generate a lower accuracy.
2. When alpha is 1, it is clear from the learning curve that the learning does not converge. There is no trend that the learning took place because both loss curve and learning curve are messy. This is because an alpha of 1 will make the learning always random, the messy graphs is because the accuracy is not evolving from previous learning, making the learning meaningless.

### 3. Hyper-parameter Tuning

	Epoch=20, learning_rate=0.5, reg=0.0001
Training Accuracy	0.9847
Validation Accuracy	0.9719
Test Accuracy	0.9729

Explain why your choice works:

From previous experiment, we can find that with increase in learning rate and decrease in reg, the accuracy will get improved. However, we also finds that too large learning rate will make learning a little unstable, so I chose learning\_rate as 0.1 and reg as 0.0001. Also, we can find that from almost every learning curve (except the one with reg=1), the accuracy increase with a larger epochs. Considering the increased running time, I chose Epoch as 20. In this way, I got a test accuracy rate of 97.29%, which is higher than any of previous test and my other experiments.