# 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:

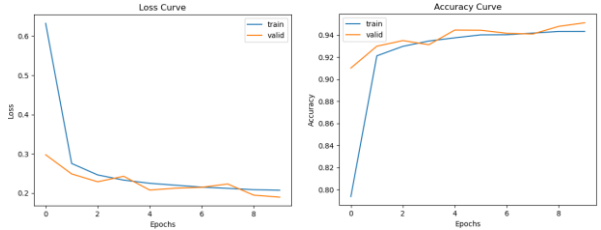
| | lr=1 | lr=1e-1 | lr=5e-2 | lr=1e-2 |
|---|---|---|---|---|
| Training Accuracy | 0.9512 | 0.9266 | 0.9153 | 0.7706 |
| Test Accuracy | 0.9507 | 0.9250 | 0.9133 | 0.7607 |

*Validation Accuracy is used for Training Accuracy

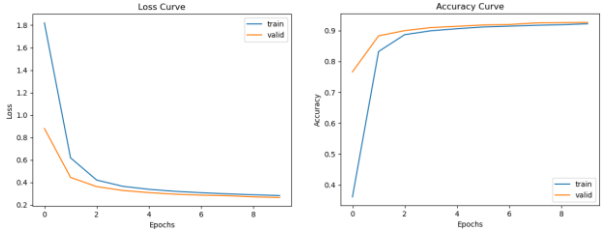
# 1. Learning Curve

Plot the learning curves using the learning rates from the previous slide and put them below (you may add additional slides if needed).

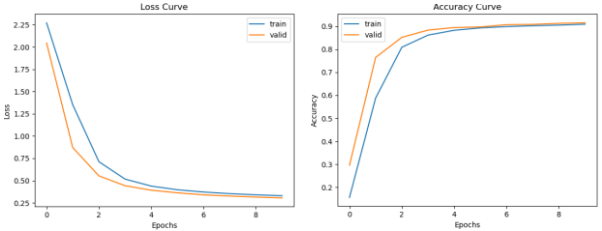
Learning Rate = 1



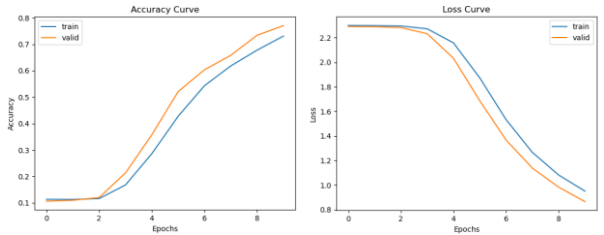

Learning Rate = 1e-1




Learning Rate = 5e-2




Learning Rate = 1e-2



# 1. Learning Rates

## Describe and Explain your findings:

Learning rate in deep learning neural network is a hyper parameter in optimization algorithm (SGD) which refers to the step size (iterations) to get to the minimum loss function. Either very high learning rate or very low learning rate might impact the overall performance when training the model. For this experiment, 4 learning rates were tested (1, 1e-1, 5e-2, 1e-2). My expectation was higher learning rate will have better results since learning rate of 1 is not very high. As expected, learning rate 1 had testing accuracy of 0.9512 and training accuracy of 0.9507. This accuracy score was satisfying and fulfilled my expectation. Both training accuracy and testing accuracy went down as learning rate decreased. Learning rate of 1e-2 had training accuracy of 0.7706 and testing accuracy of 0.7607 which still needs improvement.

# 2. Regularization

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

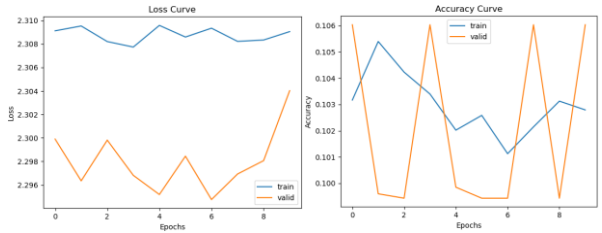
| | alpha=1 | alpha=1e-1 | alpha=1e-2 | alpha=1e-3 | alpha=1e-4 |
|---|---|---|---|---|---|
| Training Accuracy | 0.1028 | 0.3392 | 0.8842 | 0.9220 | 0.9304 |
| Validation Accuracy | 0.1060 | 0.3519 | 0.8957 | 0.9261 | 0.9343 |
| Test Accuracy | 0.1135 | 0.3918 | 0.8944 | 0.9260 | 0.9343 |

*Average accuracy of the last epoch is used for Training Accuracy

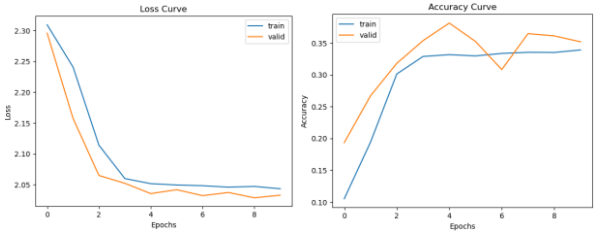
# 2. Regularization

Plot the learning curves using the regularization coefficients from the previous slide and put them below (you may add additional slides if needed).

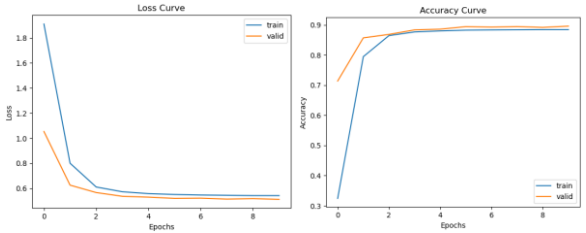
Alpha = 1



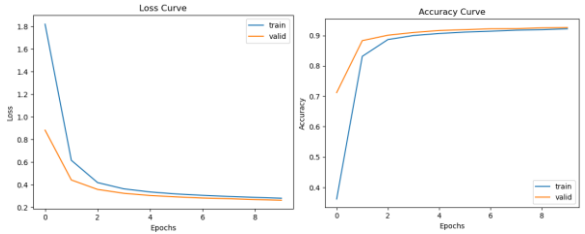

Alpha = 1e-1



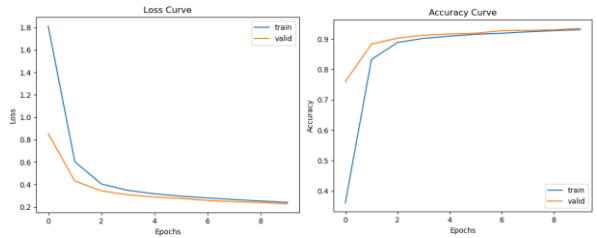

Alpha = 1e-2




Alpha = 1e-3




Alpha = 1e-4



# 2. Regularization

## Describe and Explain your findings:

Alpha is a hyperparameter in deep learning regularization which is related to overfitting. It constrains the size of the weights. Like learning rates, either high alpha value or low alpha value might cause the model to either overfit or underfit. My hypothesis was higher alpha value will have better testing and training accuracy since high alpha value  might solve issues with high variance by smaller weights. The result was different compare to my hypothesis. Alpha value of 1 had testing accuracy of 0.1028, validation accuracy of 0.1060, and training accuracy of 0.1135. Result was very bad compare to what is expected. Accuracy increased as alpha value decreased. As a result, alpha value of 1e-4 had testing accuracy of 0.9304, validation accuracy of 0.9343, and training accuracy of 0.9343. This might have happened since weight that we have is small enough that we don’t really need higher alpha values.

# 3. Hyper-parameter Tuning

You are now free to tune any hyper-parameters for better accuracy. Create a table below and put the configuration of your best model and accuracy into the table:

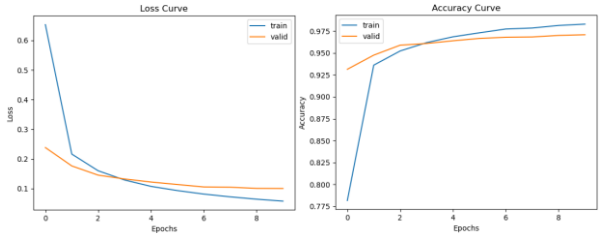
| | alpha=1e-4<br>lr = 1 | alpha=1e-4<br>lr = 2 | alpha=1e-5<br>lr = 1 | **alpha=1e-6<br>lr = 1** | alpha=1e-7<br>lr = 1 |
|---|---|---|---|---|---|
| Training Accuracy | 0.9796 | 0.9630 | 0.9829 | **0.9857** | 0.9797 |
| Validation Accuracy | 0.9706 | 0.9589 | 0.9708 | **0.9718** | 0.9667 |
| Test Accuracy | 0.9711 | 0.9618 | 0.9712 | **0.9733** | 0.9673 |

*Average accuracy of the last epoch is used for Training Accuracy

# 3. Hyper-parameter Tuning

## Explain why your choice works:

Alpha = 1e-6, lr = 1




As a benchmark, I used alpha value and learning rate value from the previous experiment. Alpha value of 1e-4 and learning rate of 1 gave the training accuracy of 0.9796, validation accuracy of 0.9706, and testing accuracy of 0.9711. Accuracy was good enough, but I tuned to get better results. First, I tried to increase the learning rate to 2 with same alpha but the result was not good compare to my benchmark. Second, I used smaller alpha values with learning rate of 1. It started to give better results. My hypothesis here was “smaller alpha values will always give better results”. When alpha value reached 1e-7, accuracy score got lower compare to the alpha value of 1e-6. I got the best result using alpha value of 1e-6 and learning rate of 1 which had training accuracy of 0.9857, validation accuracy of 0.9718, and testing accuracy of 0.9733. My choice of hyperparameter works because I tried different learning rates and alpha values. If the result had lower accuracy compare to the previous experiment, I kept the previous value and tuned another hyperparameters. As a result, my experiment got accuracy nearly 0.98.