Assignment 1 Writeup

Name: Chen Peng
GT Email: c.peng@gatech.edu
GT ID: 903646937

Two-Layer Neural Network

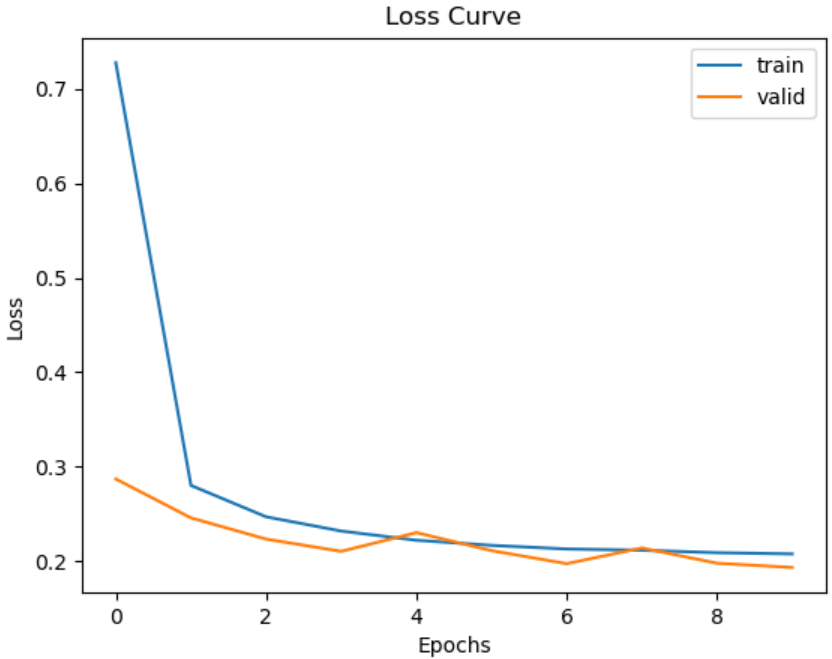
1. Learning Rates

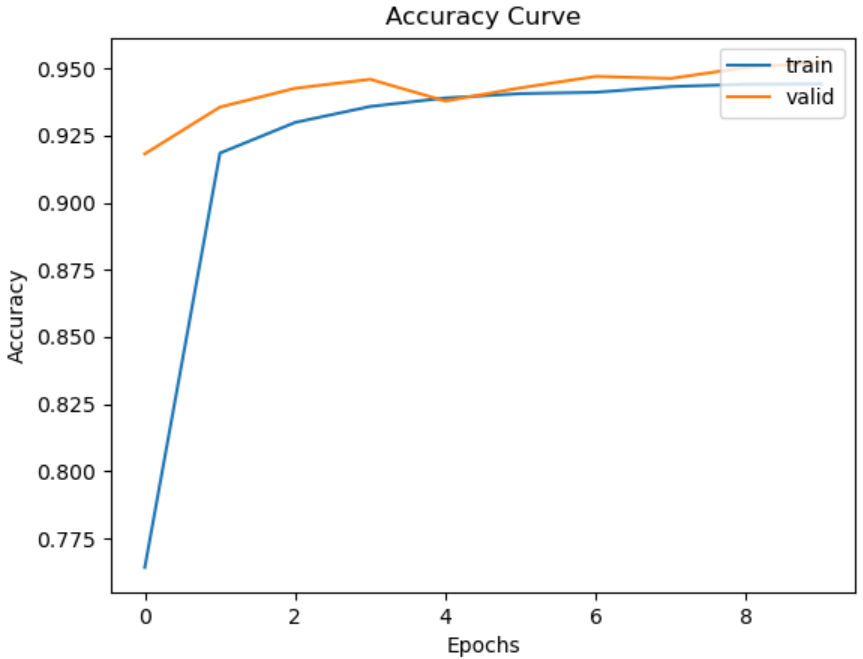
Tune the learning rate of the model with all other default hyper-parameters fixed. Fill in the table below:

	lr=1	lr=1e-1	lr=1e-2	lr=5e-2
Training Accuracy	0.9452	0.9214	0.7297	0.9088
Validation Accuracy	0.9514	0.9283	0.7725	0.9144
Test Accuracy	0.9486	0.9266	0.7602	0.9139

1. Learning Curve (lr=1)

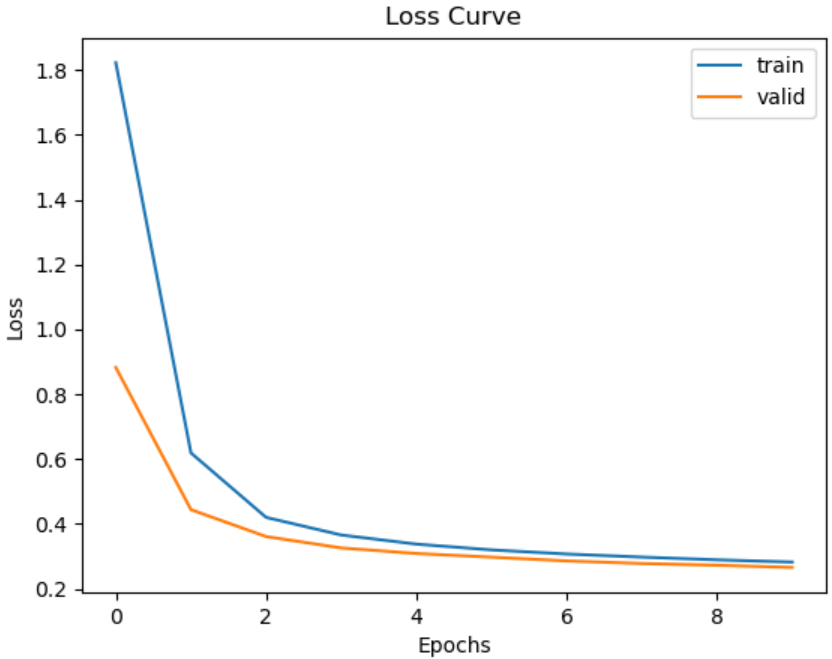
Plot the learning curves and put it below:

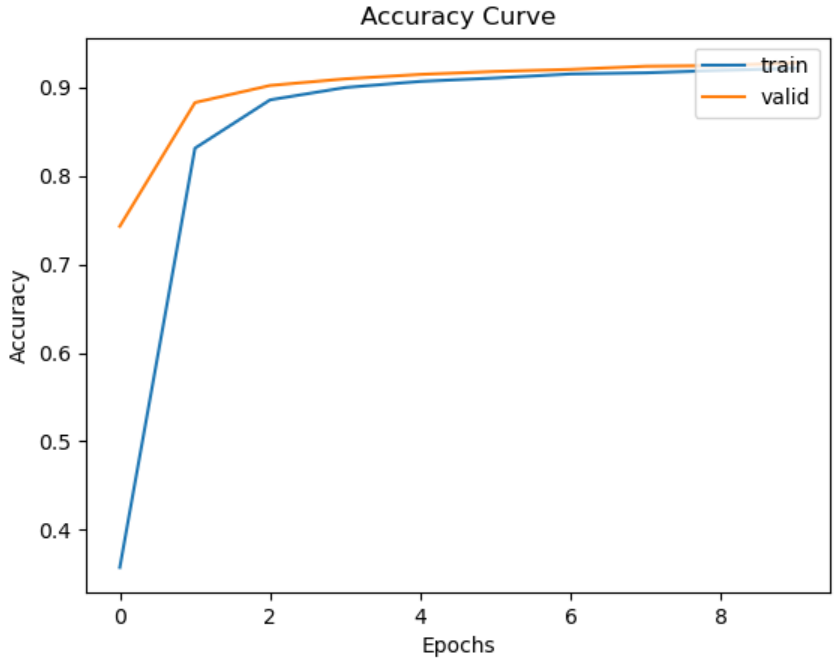




1. Learning Curve (lr=1e-1)

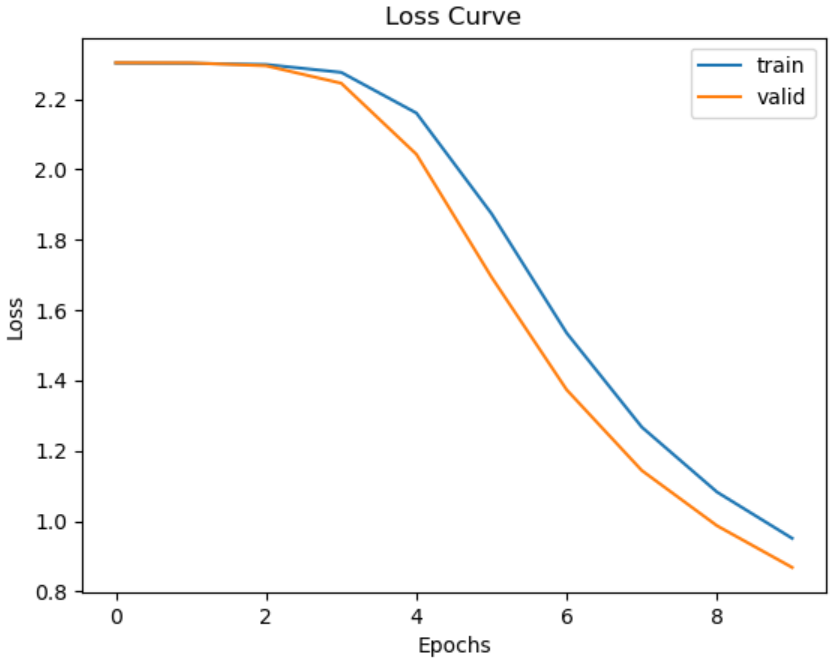
Plot the learning curves and put it below:

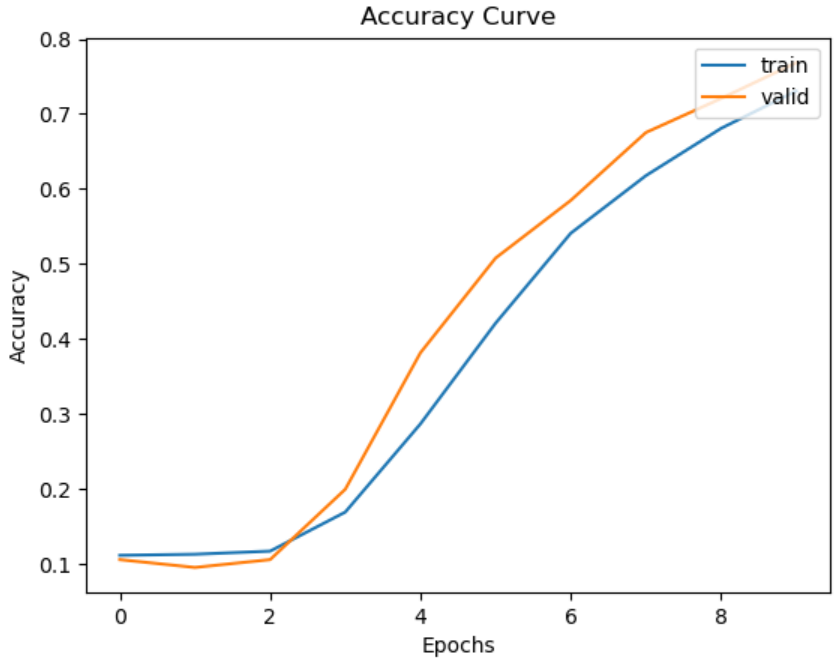




1. Learning Curve (lr=1e-2)

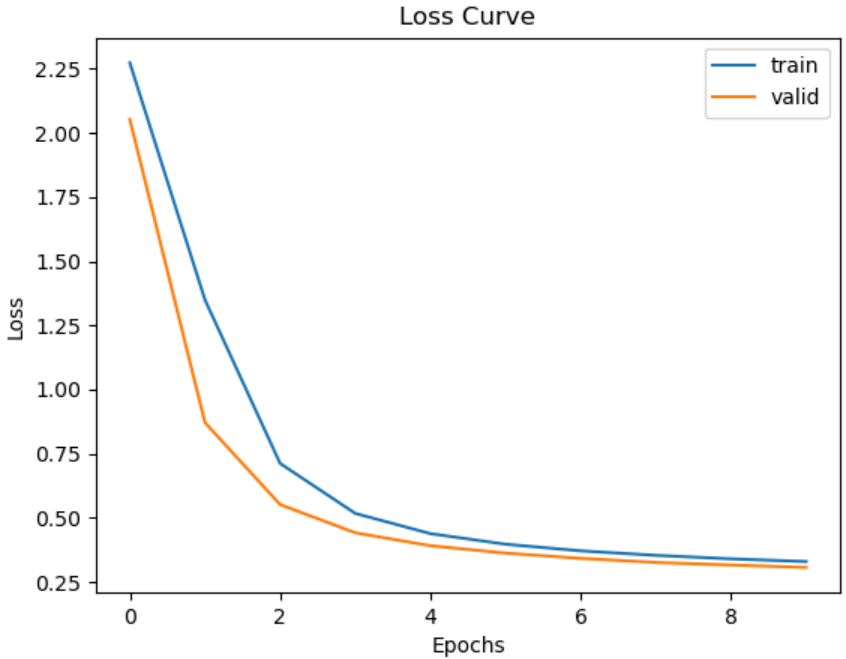
Plot the learning curves and put it below:

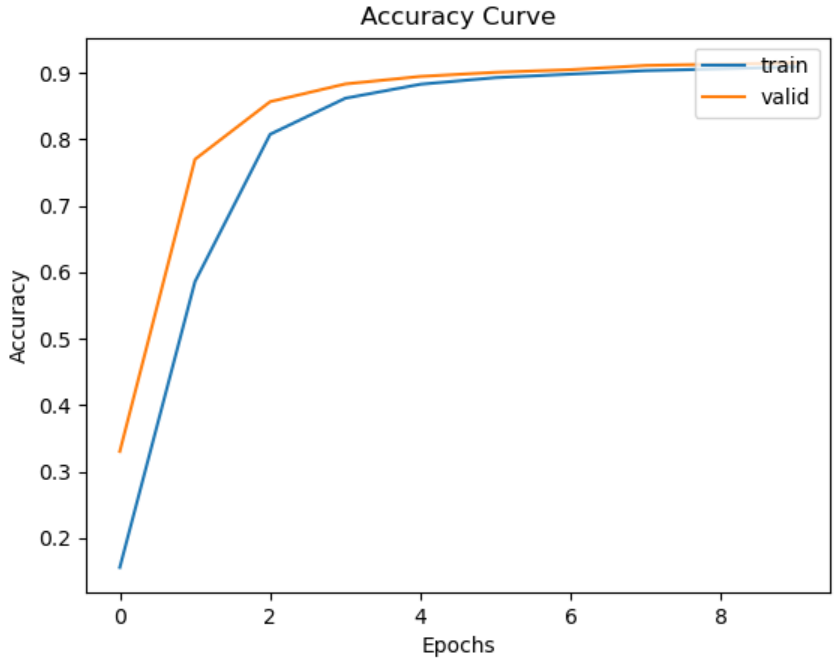




1. Learning Curve (lr=5e-2)

Plot the learning curves and put it below:





1. Learning Rates

Describe and Explain your findings:

Large learning rate (lr = 1) trains fast, but the validation accuracy oscillates and don’t converge.

Small learning rate (lr = 0.01) trains slow at the beginning, the validation and test accuracy is not as high as others after 10 epochs of training.

Proper learning rate (lr = 0.1, 0.05) can converge quickly in 10 training epochs, and obtain a good accuracy.

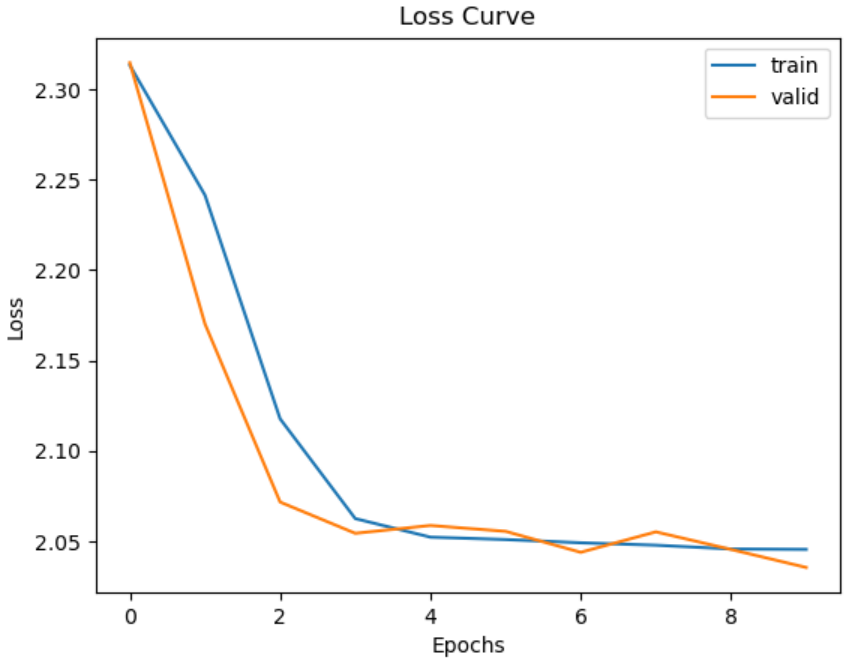
2. Regularization

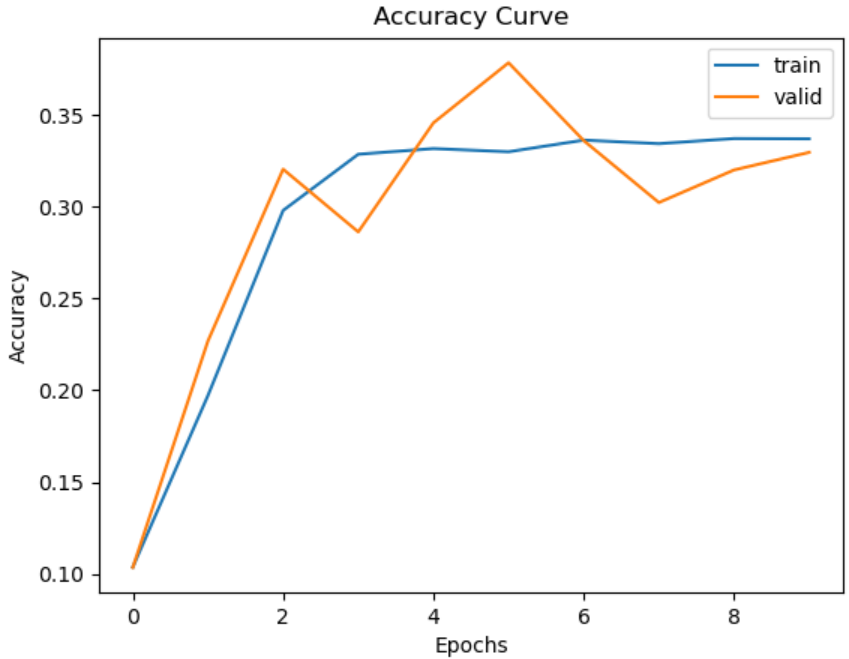
Tune the regularization coefficient of the model with all other default hyper-parameters fixed. Fill in the table below:

	alpha=1e-1	alpha=1e-2	alpha=1e-3	alpha=1e-4	alpha=1e-0
Training Accuracy	0.3380	0.8838	0.9214	0.9294	0.1044
Validation Accuracy	0.3368	0.8884	0.9283	0.9338	0.1060
Test Accuracy	0.3727	0.8920	0.9266	0.9334	0.1028

2. Regularization (alpha=1e-1)

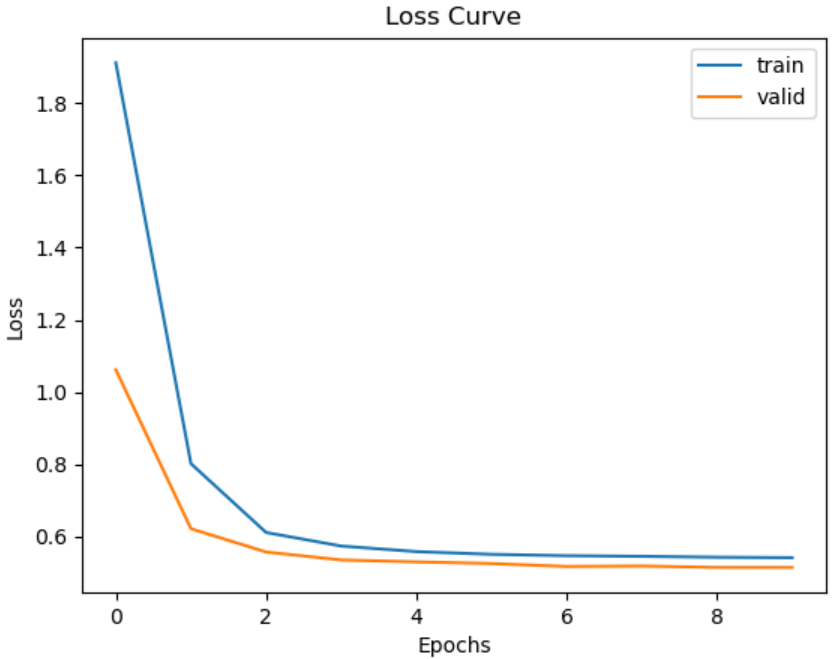
Plot the learning curves and put it below:

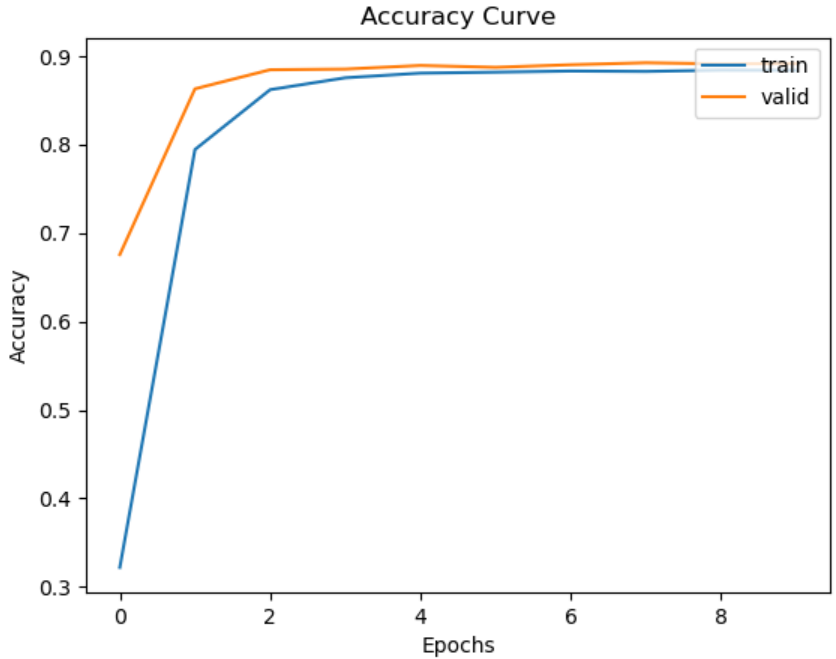




2. Regularization (alpha=1e-2)

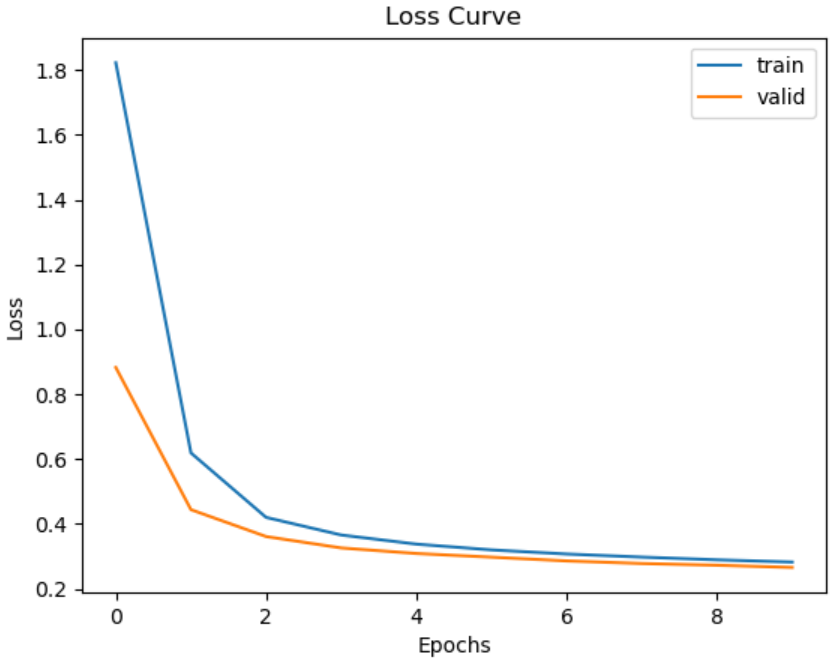
Plot the learning curves and put it below:

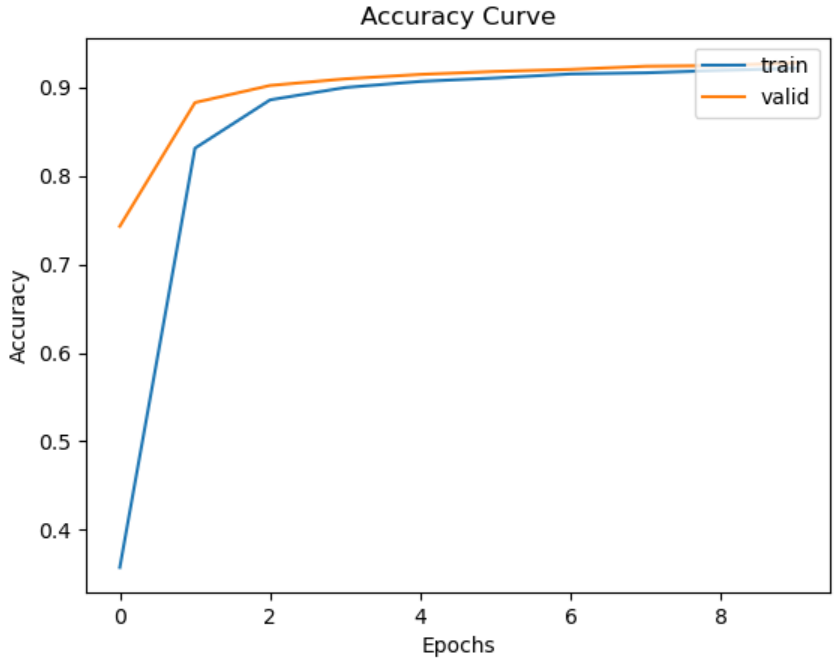




2. Regularization (alpha=1e-3)

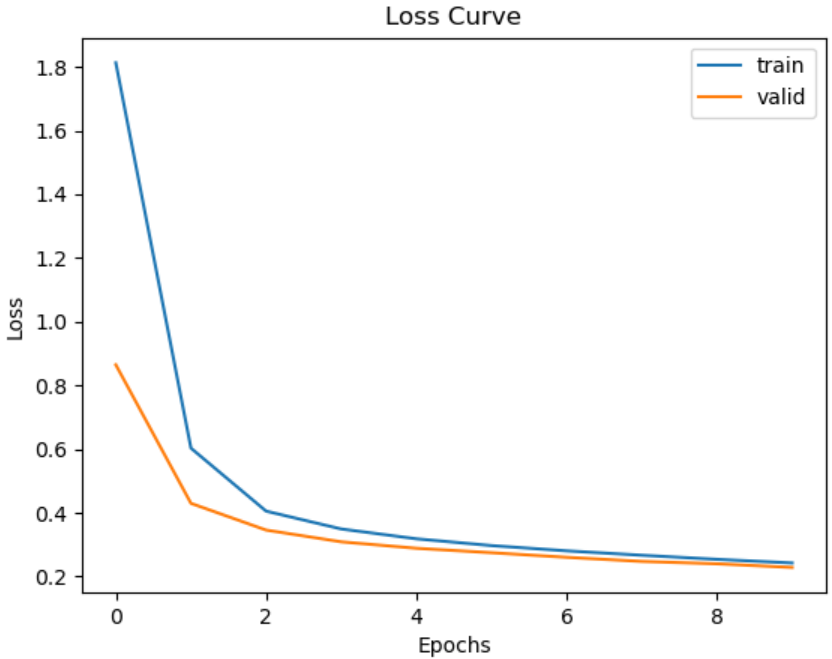
Plot the learning curves and put it below:

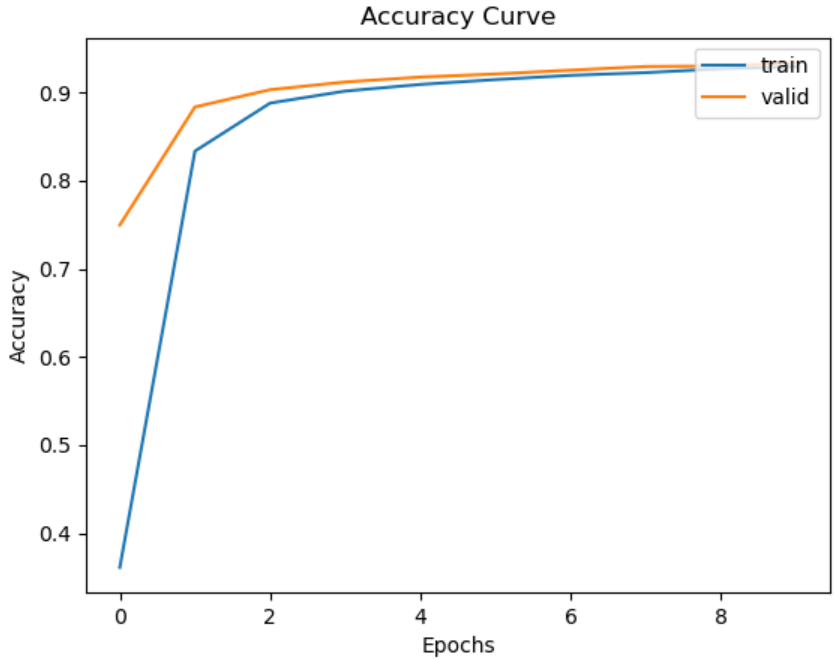




2. Regularization (alpha=1e-4)

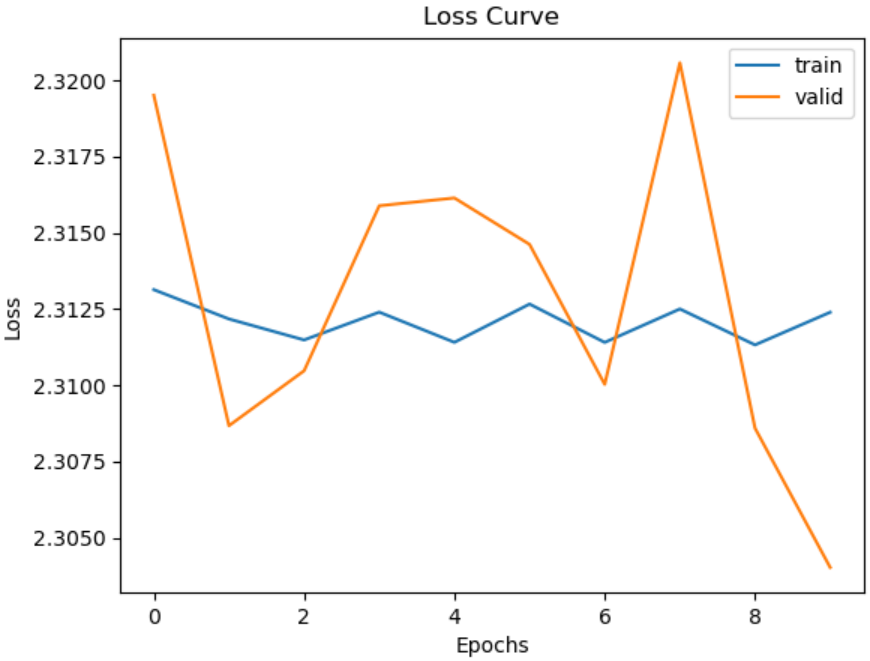
Plot the learning curves and put it below:

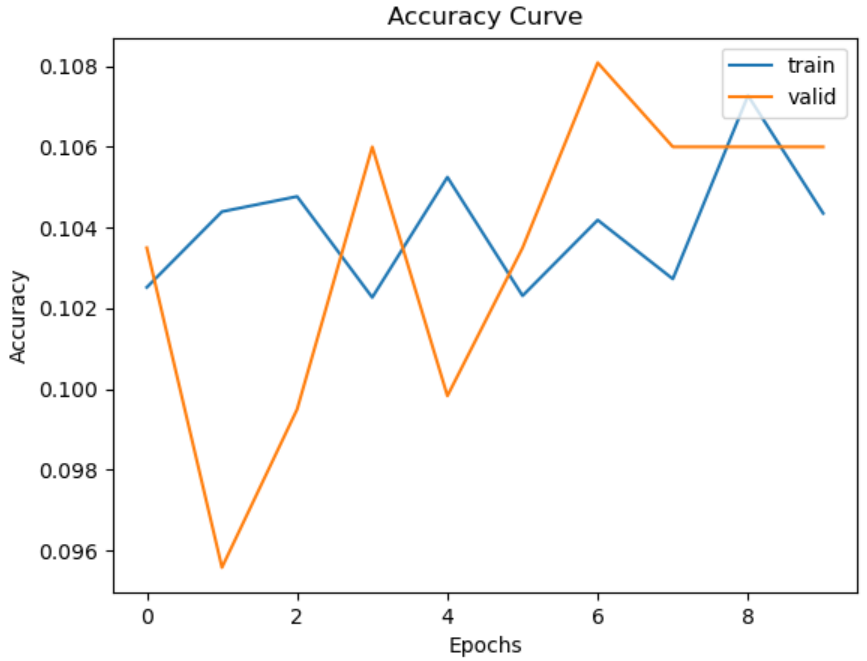




2. Regularization (alpha=1e-0)

Plot the learning curves and put it below:





2. Regularization

Describe and Explain your findings:

Large regularization coefficient force the model to choose small weights, therefore, the model cannot learn anything.

Regularization is used to limit a few high weight value that dominant the model and avoid overfitting. In this experiment, the model does not show a strong overfitting effect during 10 epochs of training process. Thus, set a small regularization coefficient makes the training process faster and obtains a higher accuracy after 10 epochs of training.

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:

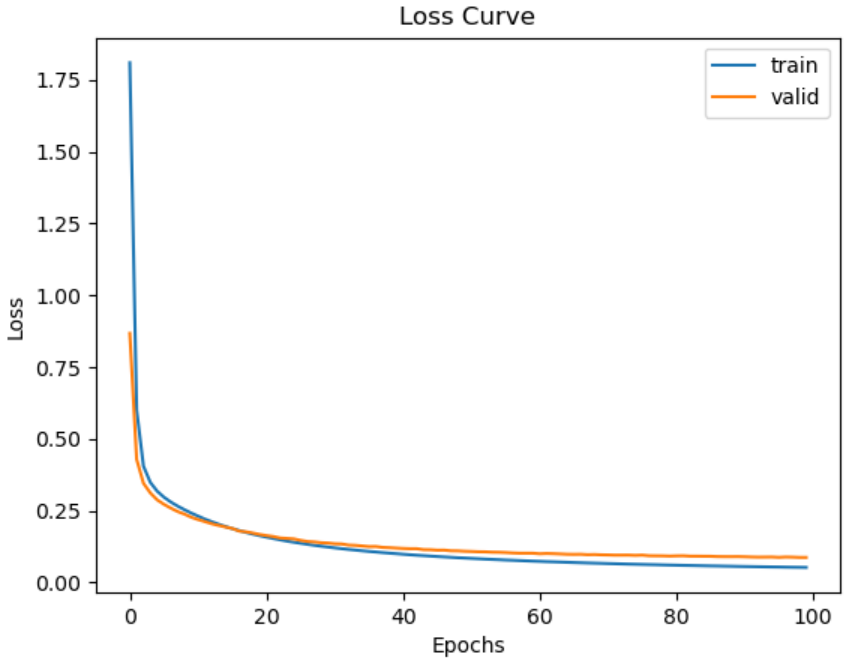
Model Type	Hidden Size	Batch size	Learning rate	Regularization coeff	Epochs	Momentum
Two Layer Net	128	32	0.05	0.0001	100	0.9

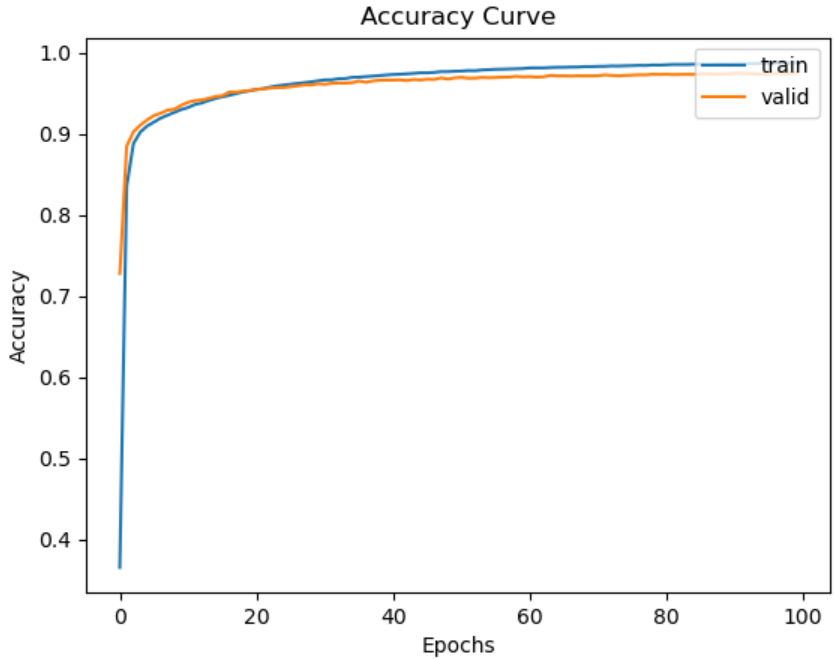
Briefly explain why your choice works:

Small batch size (32) makes learning process faster. Small learning rate (0.05) makes the model stable. Large epochs (100) makes the model learning sufficient.

3. Hyper-parameter Tuning (Learning Curve)

Plot the learning curves and put it below:





3. Hyper-parameter Tuning (Result)

The result of training, validation, and test accuracy are shown in the table:

Training Accuracy	Validation Accuracy	Test Accuracy
0.9880	0.9754	0.9760

The test result shows the test data set accuracy is up to 97.6% after 100 epochs of training. In the learning curve figure, the training loss become lower than the validation loss and the training accuracy become higher than the validation accuracy after about 20 epochs of training. In the learning curve figure, no obvious over fitting effect are shown, we may train more epochs.