Assignment 1 Writeup

- Name: Bojun Yang
- GT Email: byang301@gatech.edu
- GT ID: 903254309

Two-Layer Neural Network

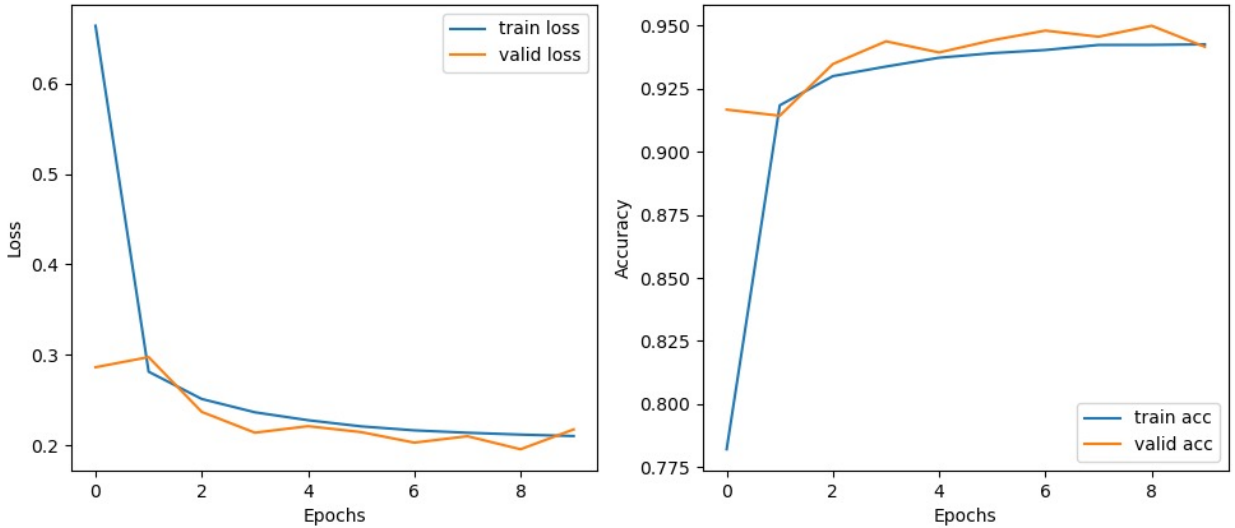
1. Learning Rates

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	0.9424	0.9223	0.7295	0.9087
Test Accuracy	0.9495	0.9248	0.7623	0.9127

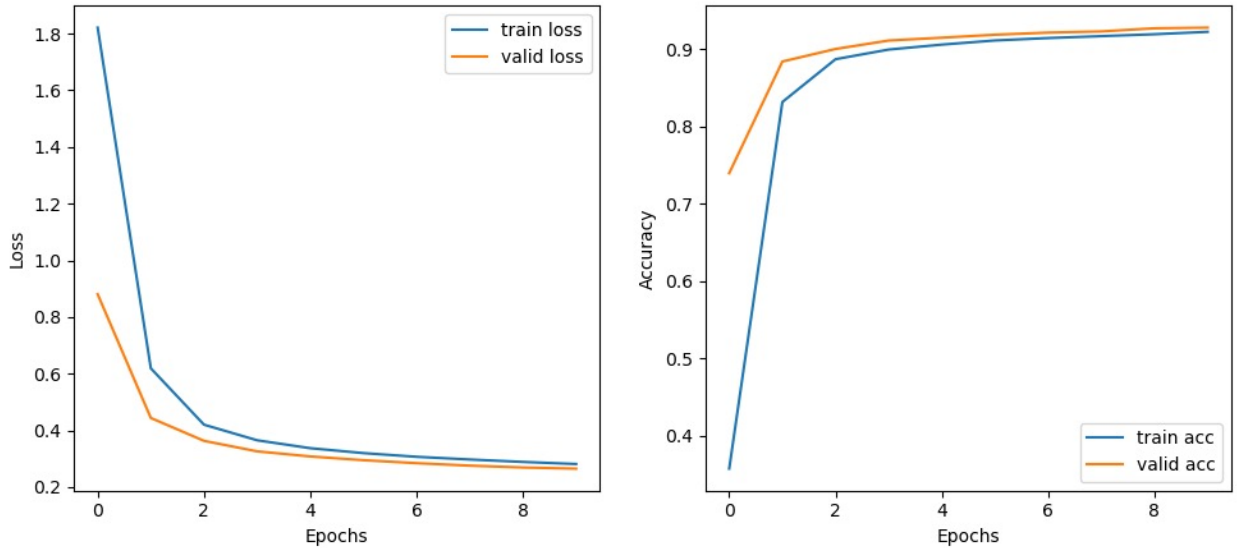
1. Learning Curve

Lr = 1



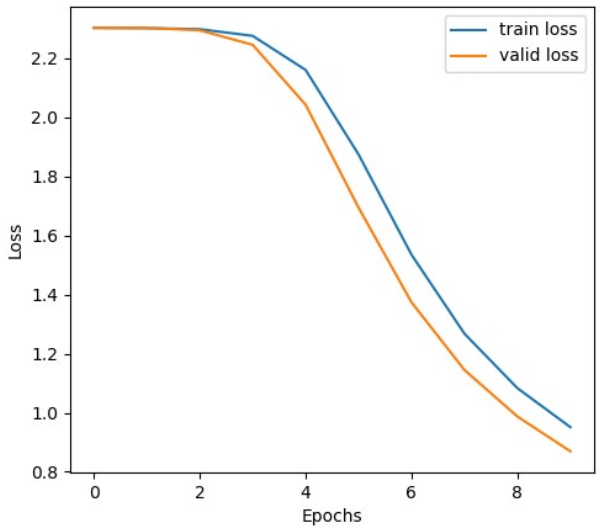
1. Learning Curve

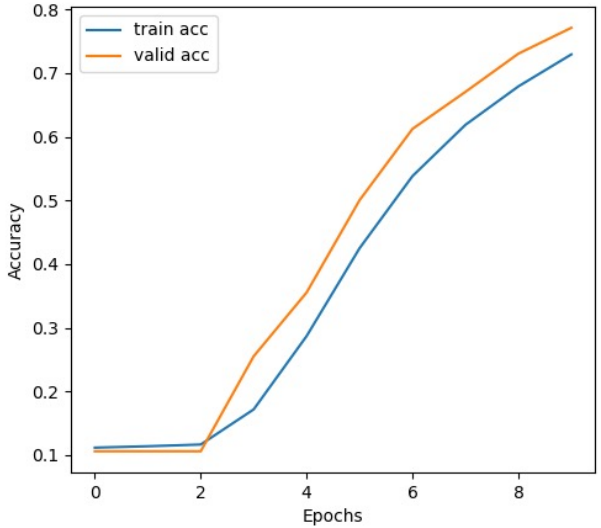
Lr = 1e-1



1. Learning Curve

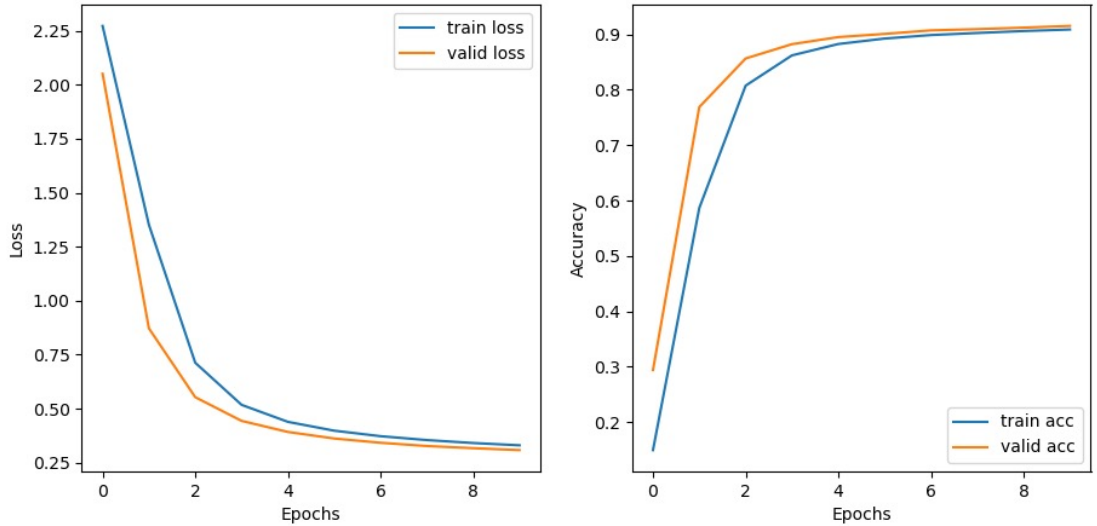
Lr = 1e-2





1. Learning Curve

Lr = 5e-2



1. Learning Rates

The accuracy for lr=1 was the highest out of all the models. However, its loss and accuracy curve are unstable. The validation accuracy rises very fast in the beginning and then goes up and down for the rest. For large learning rates, overfitting is more likely and the model can overshoot the optima if the steps are too large. This is what can cause the up and down learning curves. Since our models only had 10 epochs, a large learning rate meant that it could get its accuracy up fast, but it potentially overshot the optima.

The smaller the learning rate, the slower the accuracy will increase. With smaller learning rates, the model is less likely to overfit but it will take the model longer to reach the optima. For lr=1e-2, the accuracy was the lowest because 10 epochs was not enough for it to reach the optima. However, this model did not overfit as the testing accuracy was generally higher than the training accuracy.

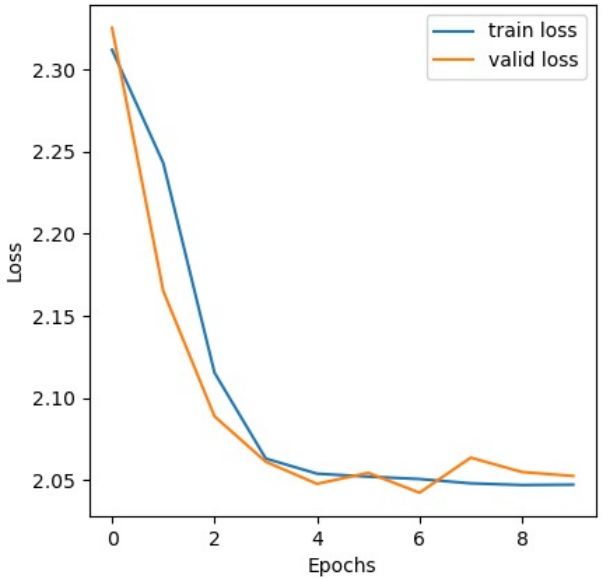
2. Regularization

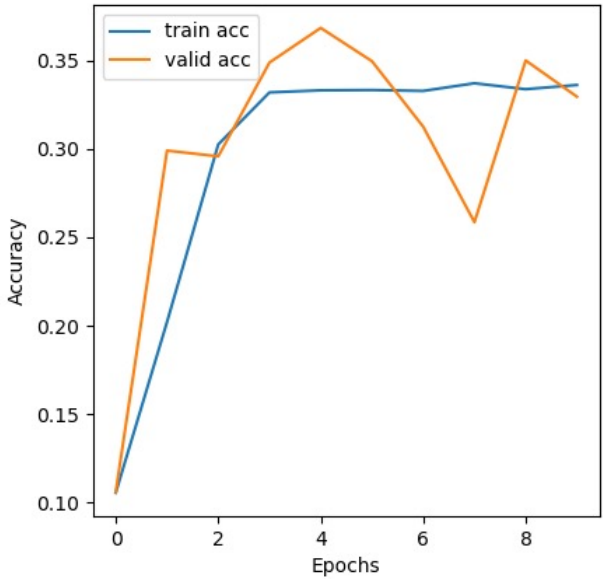
Tune the regularization coefficient of the model with all other default hyper-parameters fixed. Fill in the table below:

	alpha=1e-1	alpha=1e-2	alpha=1e-3	alpha=1e-4	alpha=1e-0
Training Accuracy	0.3319	0.8844	0.9212	0.9299	0.0989
Validation Accuracy	0.3685	0.8945	0.9282	0.9358	0.106
Test Accuracy	0.3776	0.8932	0.9256	0.9316	0.1135

2. Regularization

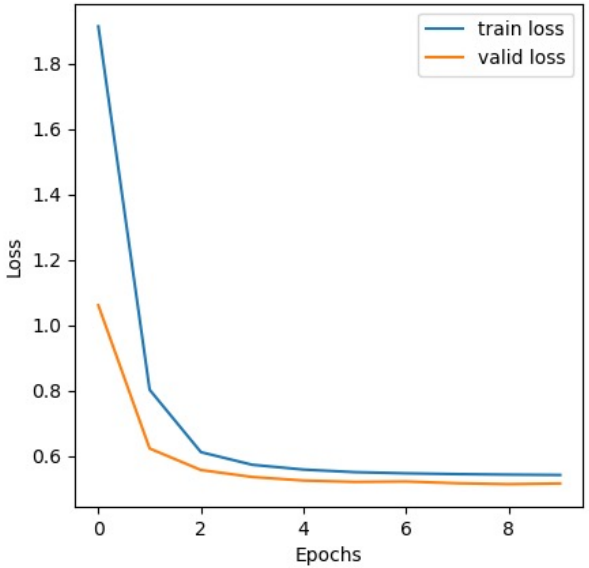
alpha=e-1

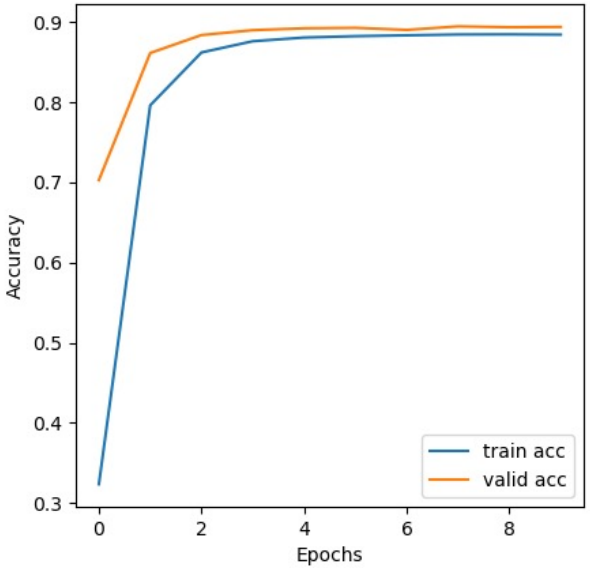




2. Regularization

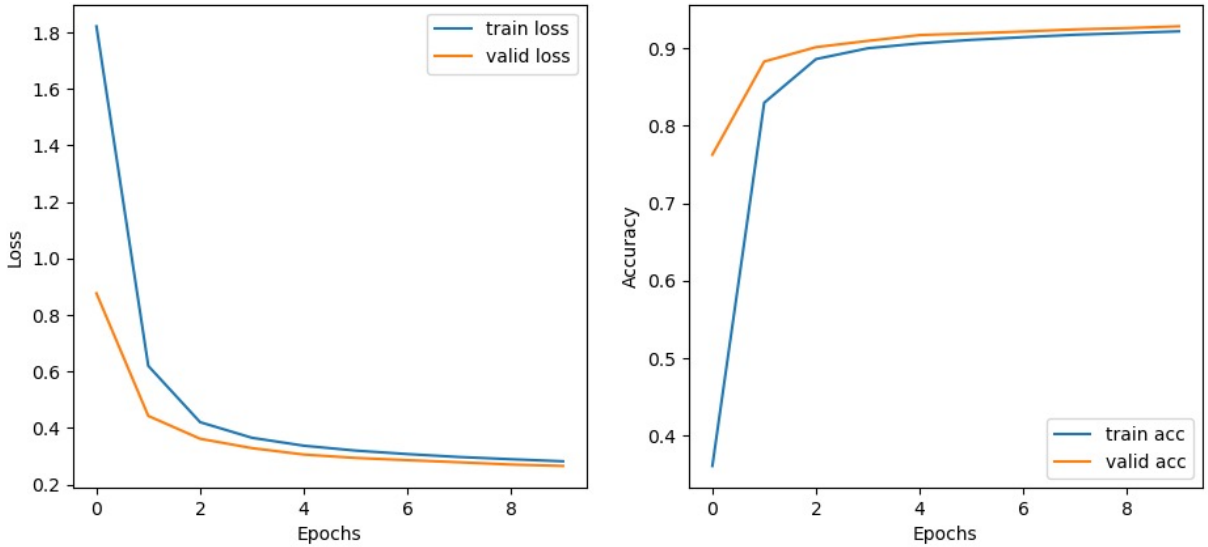
alpha=e-2





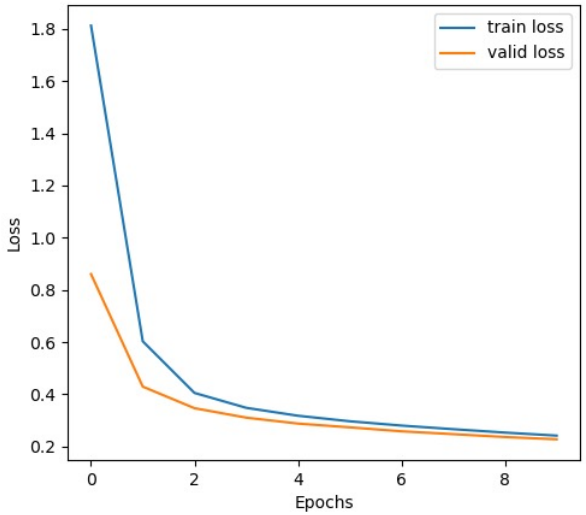
2. Regularization

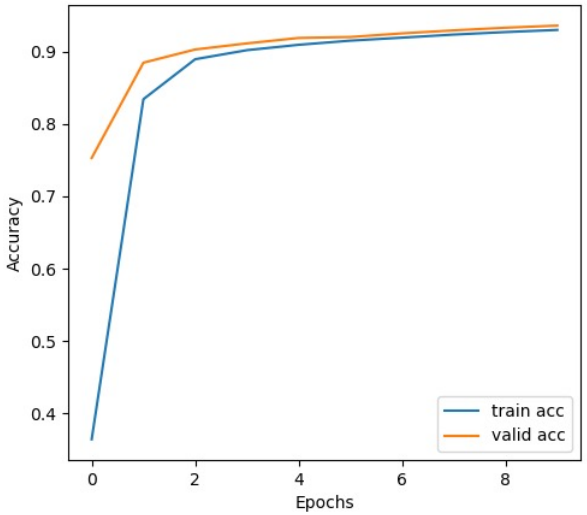
alpha=e-3



2. Regularization

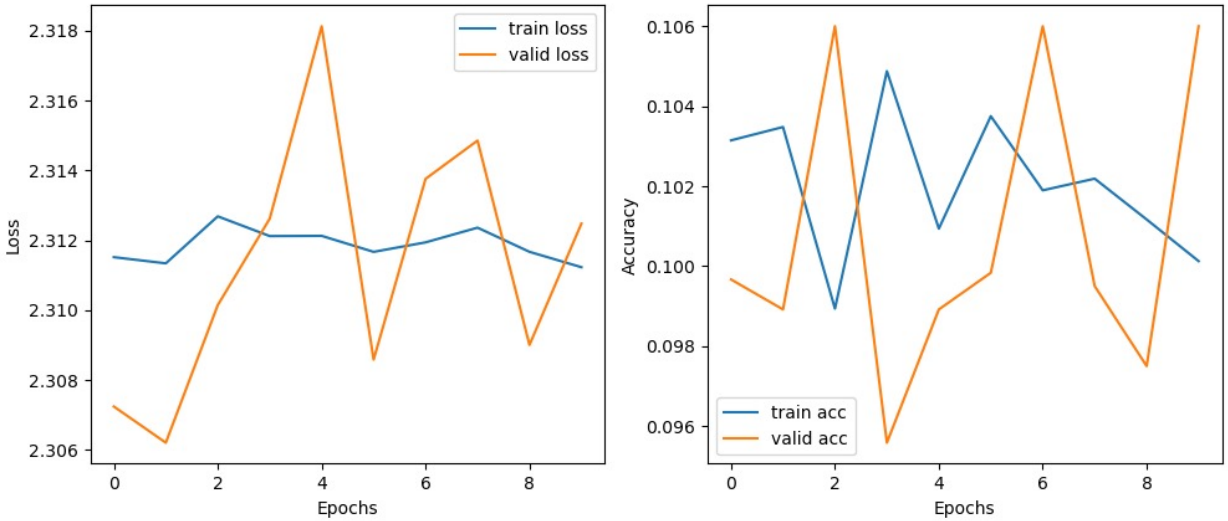
alpha=e-4





2. Regularization

alpha=e-0



2. Regularization

Regularization penalizes higher terms in the model to avoid overfitting. The L2 regularization we implemented basically tries to push the weights towards 0. The higher the regularization coeff, the more likely the model is to underfit, so if the coeff is too small, the model will tend to overfit. We can see this in our graphs where a coeff of 1 is way too big and the model doesn't learn anything, with learning curves of no trend. Based on our graphs, it seems an optimal regularization coeff is around 1e-3 to 1e-4. We still saw some random up and downs with a coeff of 1e-1.

To pair with learning rate, if the learning rate is higher, there is a higher chance of overfitting. To counteract this, we should up the regularization coeff to counteract overfitting.

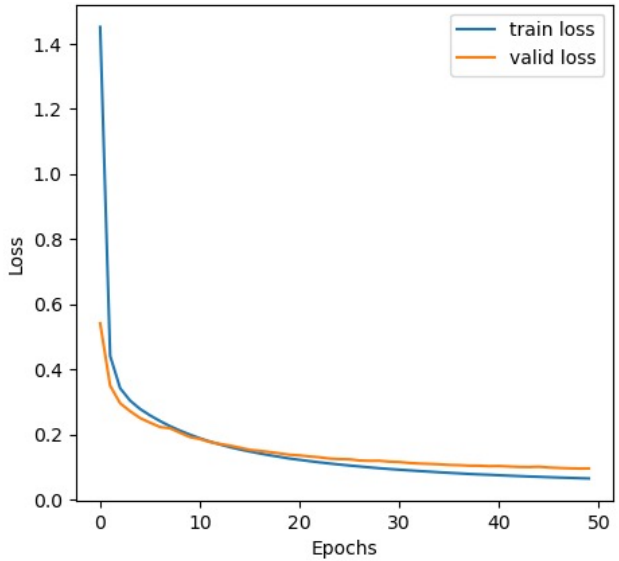
3. Hyper-parameter Tuning

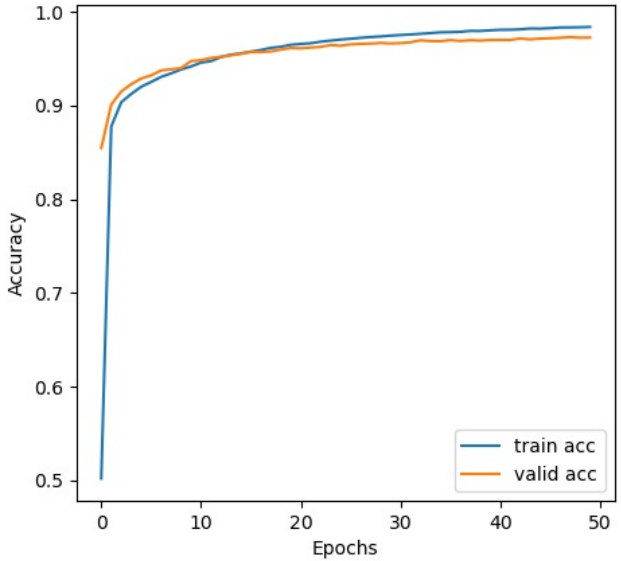
Bach-size	Lr	Reg	Epochs	Momentum
64	0.15	0.0001	50	0.9

Training acc	Validation acc	Test acc
0.9833	0.9729	0.9735

Briefly explain why your choice works:

I experimented with many values at 10 and 20 epochs. I couldn't get the accuracy to go above 95-95% so I increased the number of batches with the best combination of learning rate and regularization coefficient. It didn't seem my model was overfitting too much so I set the regularization coefficient to 1e-4 and bumped the lr up a bit so I didn't have to run too many epochs.


train loss
valid loss
Loss
Epochs
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
0
10
20
30
40
50


train acc
valid acc
Accuracy
Epochs
0.5
0.6
0.7
0.8
0.9
1.0
0
10
20
30
40
50