

CSCI 635: Introduction to Machine Learning
Homework 4: Nonlinear Prediction

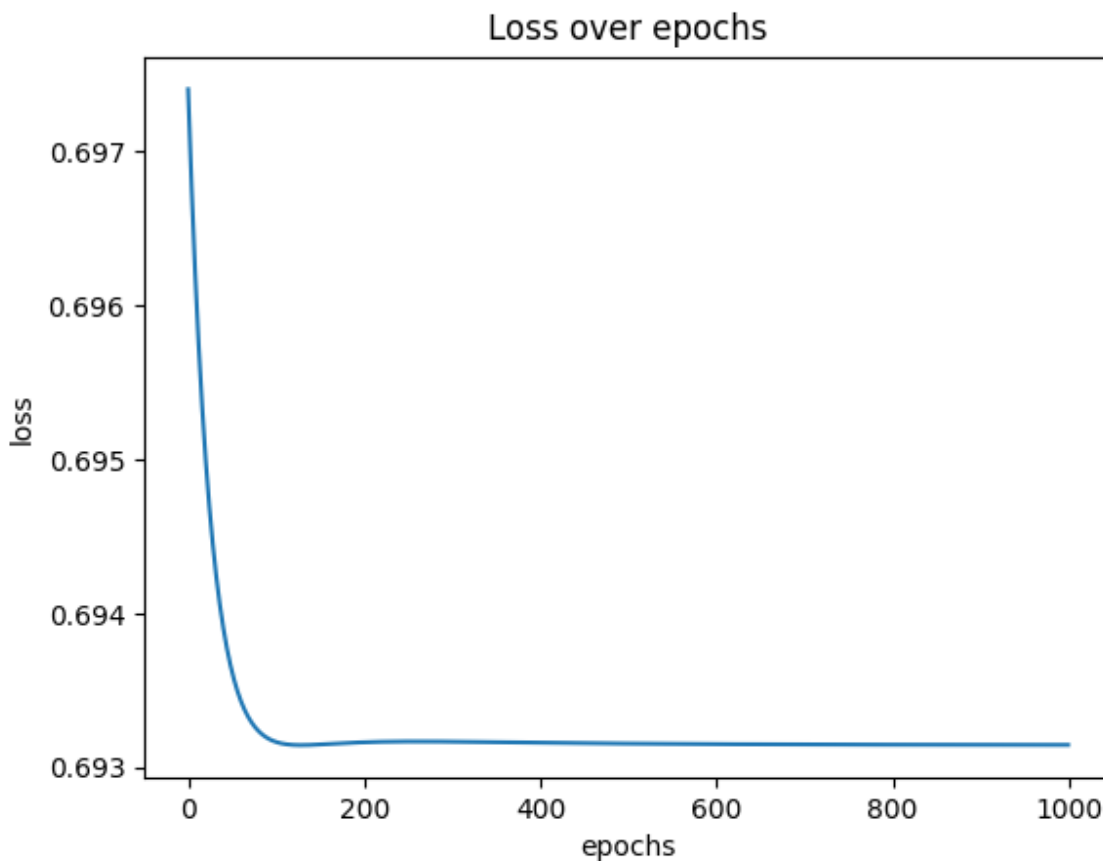
Anushree Sitaram Das (ad1707)

Problem #1: Learning a Multilayer Perceptron Model

Problem #1a: The XOR Problem Revisited

Optimal Parameters(weights and bias) can be viewed in parametersq1a.txt

Loss curve:



Number of epochs = 1000, learning rate = 0.009, number of units in hidden layer = 30.

For number of epochs less than 1000 such as 100 and 500, the gradient checking is resulting as incorrect for some parameters. For number of epochs equal and greater than 1000, the gradient checking is resulting as correct for all parameters.

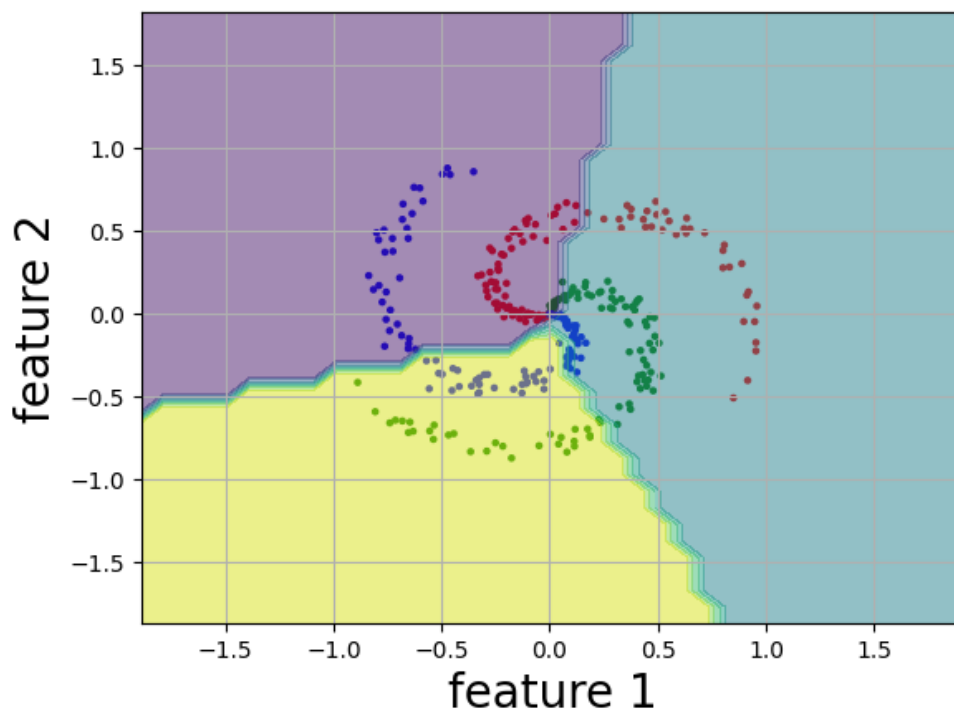
For learning rate less than 0.009, the loss is not reaching a constant value for 1000 epochs. For learning rate more than 0.009, the loss is reaching a constant value for 1000 very soon.

The final accuracy is: 50.0 %

When this model's performance is compared with that of the maximum entropy model you fit in the last homework, it is quite similar. The accuracy achieved is 50% for both and the loss curves are similar as well.

Problem #1b: The Spiral Problem Revisited

The final decision plot:

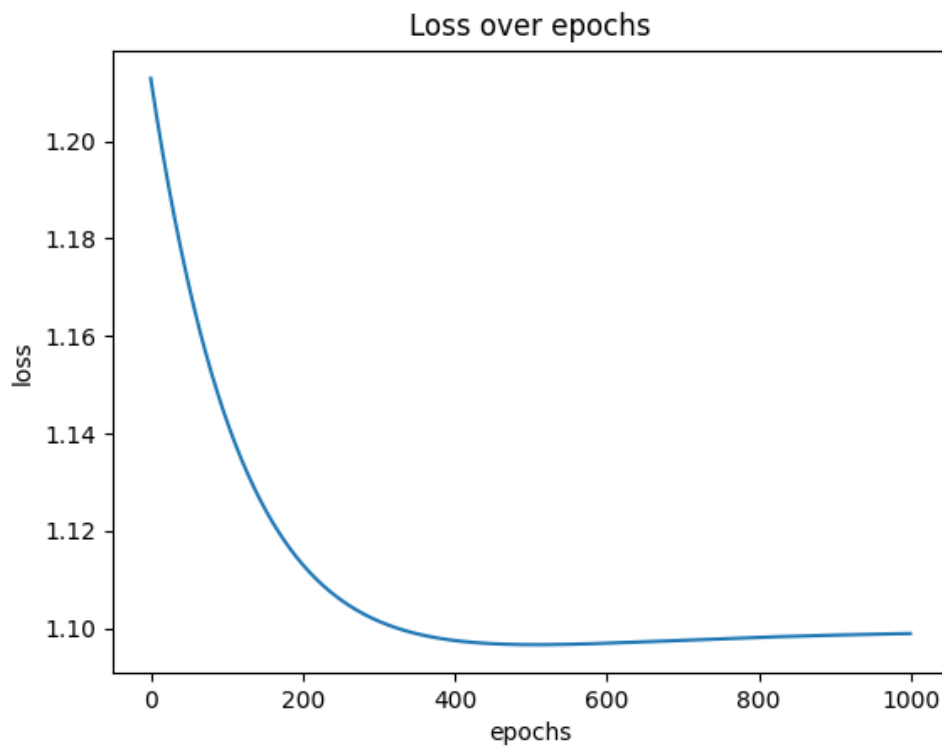


Optimal Parameters(weights and bias) can be viewed in parametersq1b.txt

Number of epochs = 1000, learning rate = 0.005, number of units in hidden layer = 25.

The final accuracy is: 54.66%

Since, it is impossible to separate the given classes using straight lines, the model classifies many labels incorrectly. Hence, the accuracy is only 54.66%.

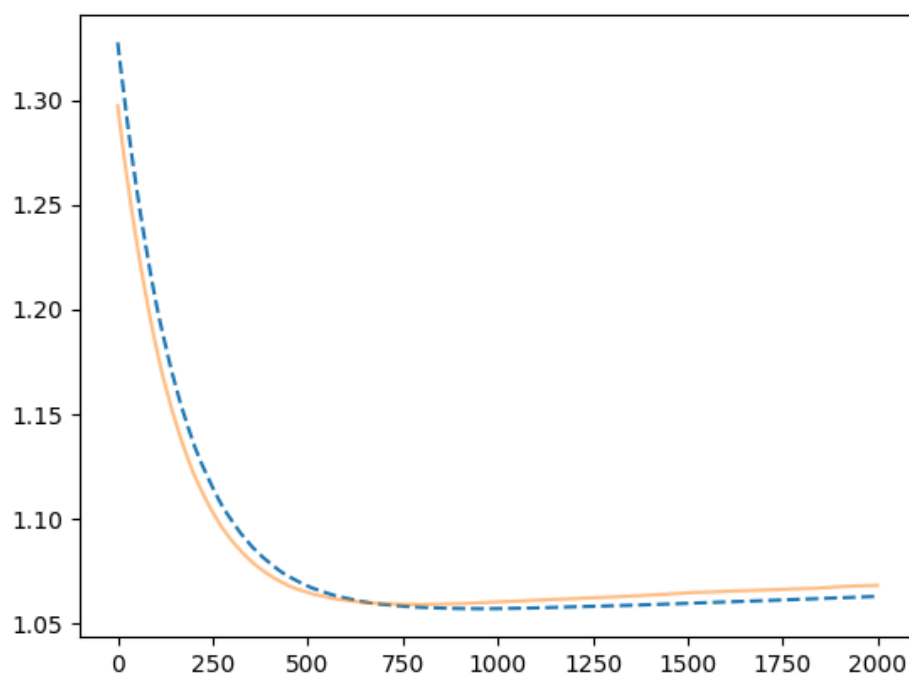


When this model's performance is compared with that of the maximum entropy model you fit in the last homework, it is a little better. The accuracy achieved for the previous model was 50%, but accuracy achieved for this model is 54.77%. The loss curves are similar for both of them.

Problem #1c: IRIS Revisited

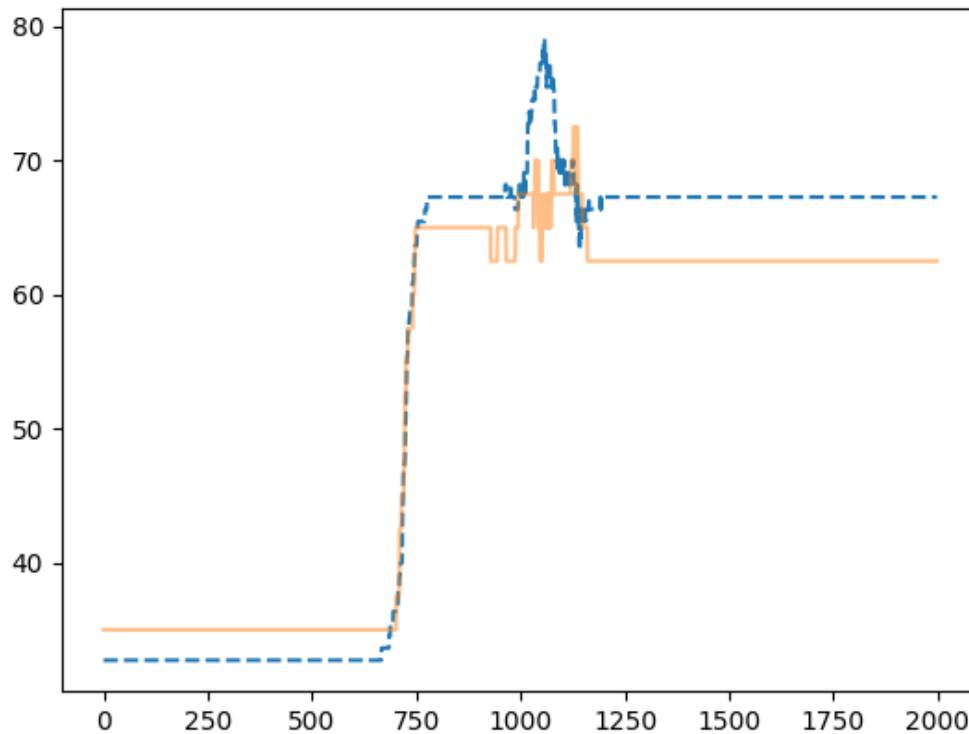
Training without regularization:

The loss over epoch plot for training and validation datasets is as follows:



Number of epochs = 1000, learning rate = 0.001, batchSize = 25, lamb = 1, number of units in hidden layer = 20

The accuracy over epoch for training and validation datasets is as follows:



The blue dashed line represents the training loss and the orange line represents the validation loss. The losses for both, training and validating, datasets are similar for most values of epochs.

As you can see, the accuracy drops after around 1000th iteration. The reason for this drop in accuracy may be that the size of training or validation data is very small or it is over-fitting the model. To reduce over-fitting, we can use weight regularizations.

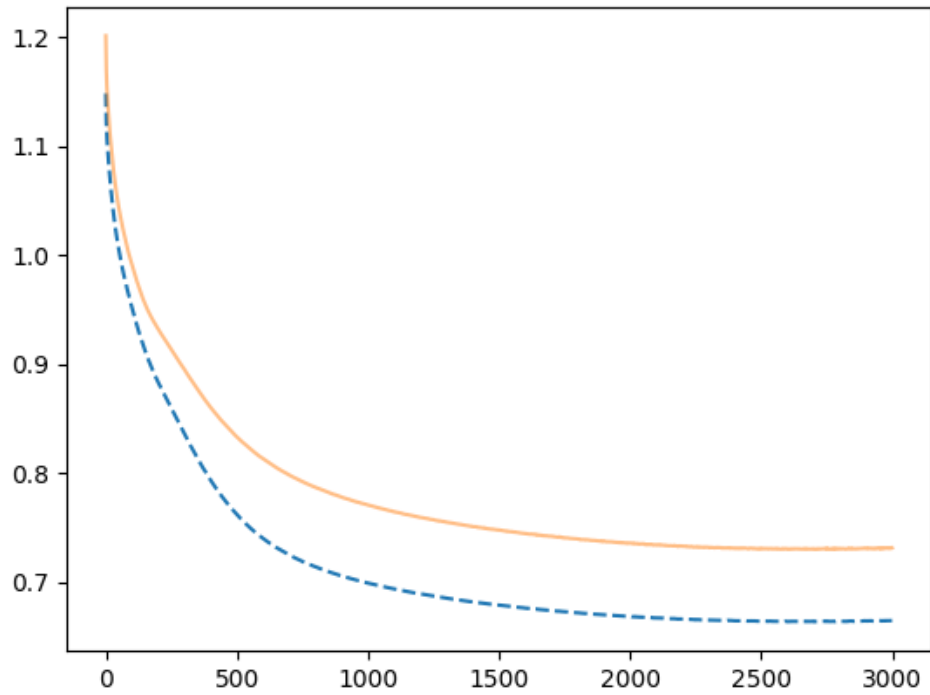
Final Accuracy at iteration 1000: 64.8 %

Training without regularization:

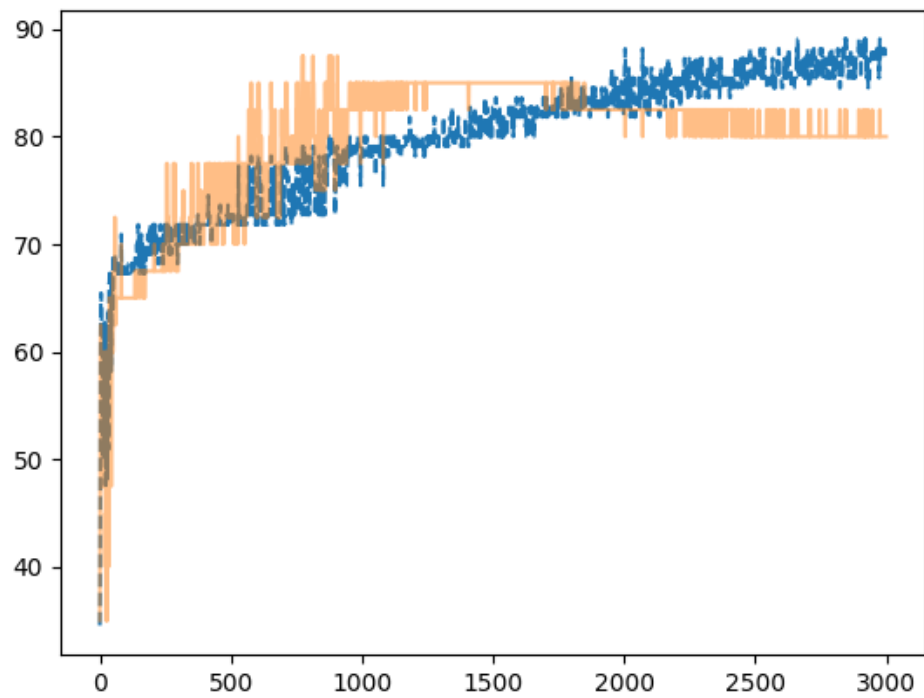
Optimal Parameters(weights and bias) can be viewed in parametersq1c.txt

Number of epochs = 3000, learning rate = 0.05, batch size = 10, lamb = 0.1, number of units in hidden layer = 20

The loss over epoch plot for training and validation datasets is as follows:



The accuracy over epoch for training and validation datasets is as follows:



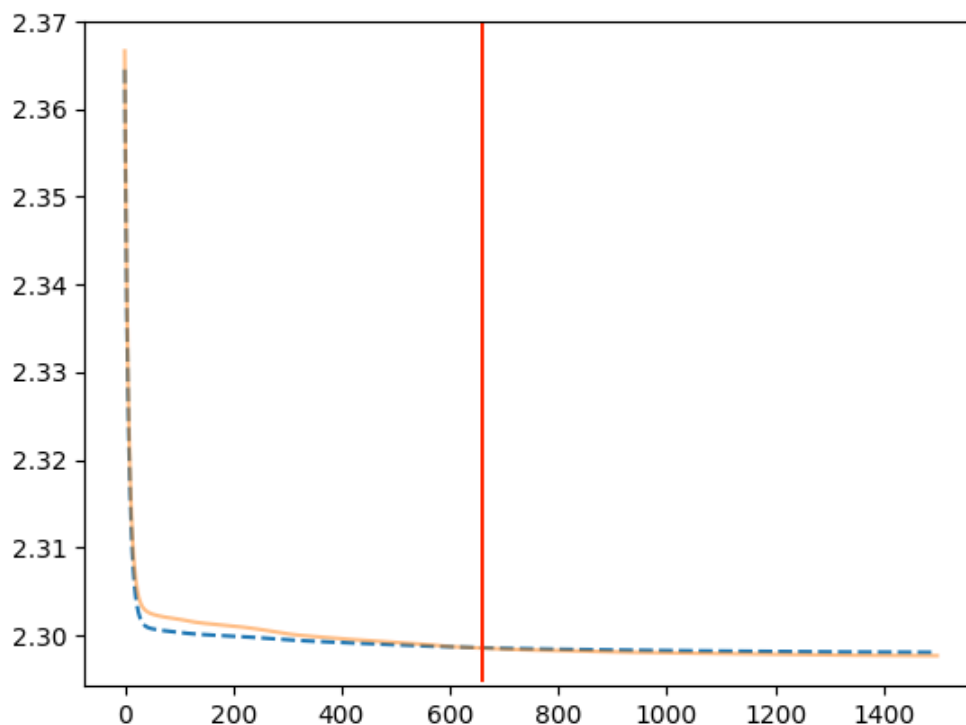
The blue dashed line represents the training loss and the orange line represents the validation loss. The losses for both, training and validating, datasets are similar for most values of epochs.

Around iteration 1700, the accuracy starts to drop for validation loss while accuracy continues to increase for training set. To keep the model from overfitting, we stop training at iteration 1700.

Final Accuracy at 1700 iteration: 82.5 %

Problem #2: Image Categorization with MLPs

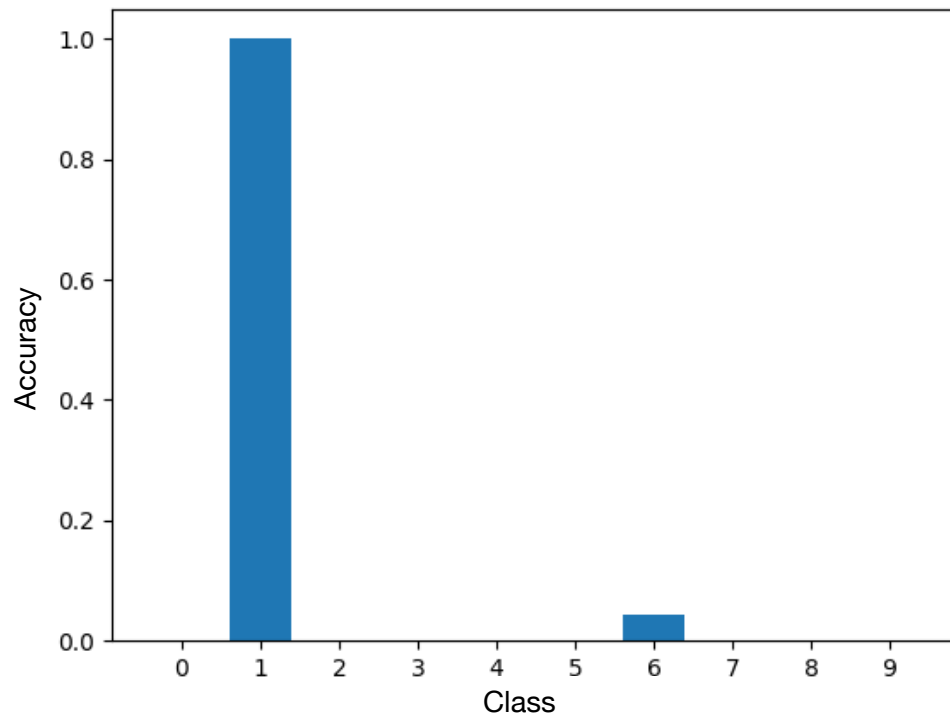
Learning curve plot showing the cross-entropy vs. epoch for both the training set and the validation set:














Number of epochs = 600, learning rate = 0.0001, batch size = 10, lambda = 0.0, number of units in hidden layer = 20

The red line shows the epoch where overfitting occurs, which is around 650. For learning rate less than 0.0001, the loss is not reaching a constant value for 600 epochs. For learning rate more than 0.0001, the loss is reaching a constant value for 600 very soon.

Histogram showing the accuracy for all classes, and for each individual digit class :

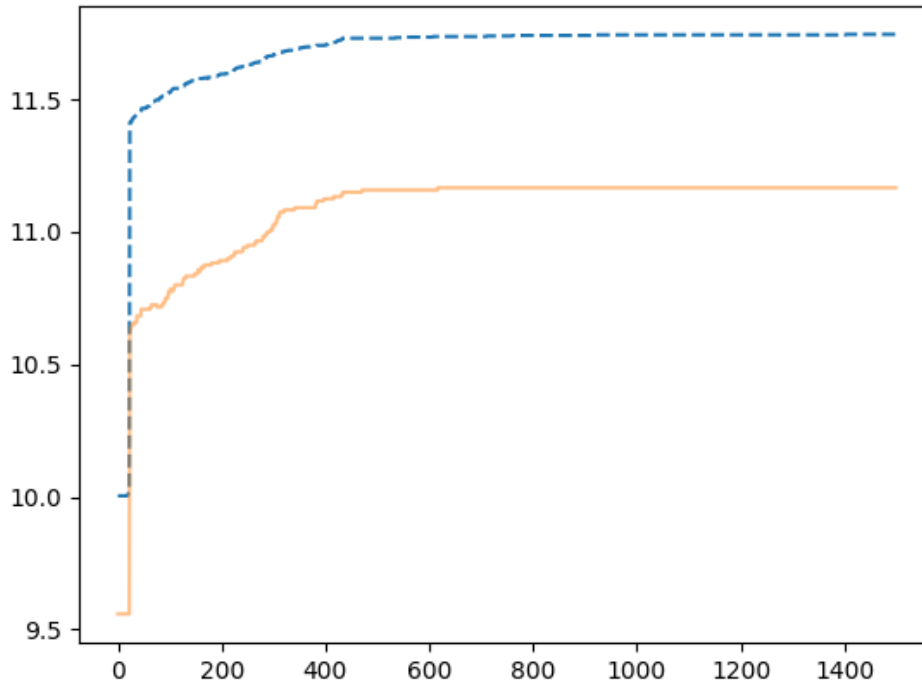


Example of one correct and one incorrect test sample from each digit class:

Label	Correct sample	Incorrect sample	Incorrect sample predicted as
0			1
1			
2			1
3			1
4			1
5			1
6			1
7			1
8			1
9			1

As we can see, most of the samples are predicted as label 1. This is probably because the parameters are not efficient enough.

Accuracy:



The accuracy becomes constant around iteration 650.

Optimal Parameters(weights and bias) can be viewed in parametersqlb.txt