Programming assignment 1: MNIST Classification using MLP

Arunima Sarkar,EE21S062

August 2022

1 Backpropagation from scratch

1.1 Backpropagation on Baseline model

The baseline parameters are, batch size = 64 , epoch size = 15 , learning rate $\eta = 0.01$. Due to memory constraints I have been able to implement only 30000 images on train set. Below are the results for these hyper-parameters.

Activation function : Sigmoid

$$\sigma(x) = \frac{1}{1 + \exp^{-x}} \tag{1}$$

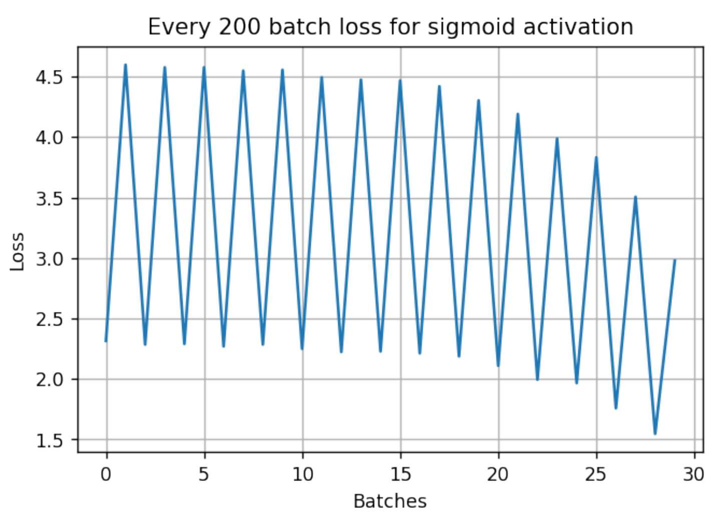


Figure 1: Batch Loss for sigmoid

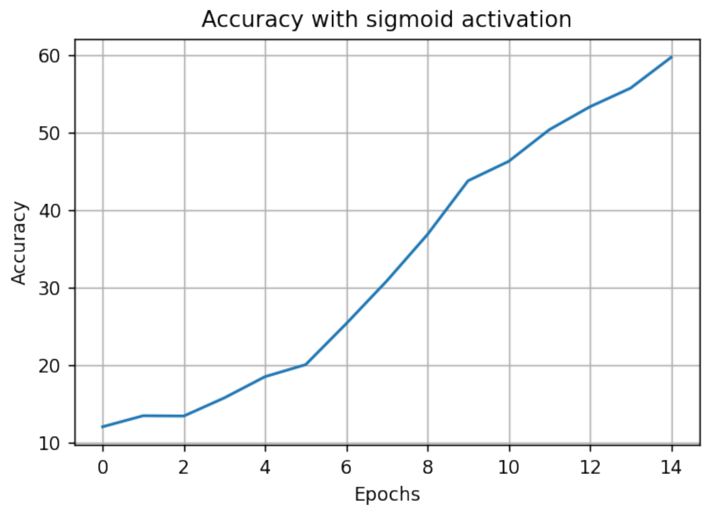


Figure 2: Accuracy on train set with sigmoid

On test set we used all of 10,000 images and the test set accuracy is achieved as 62%.

Activation function :Tanh

$$\tanh x = \frac{\exp^x - \exp^{-x}}{\exp^x + \exp^{-x}} \tag{2}$$

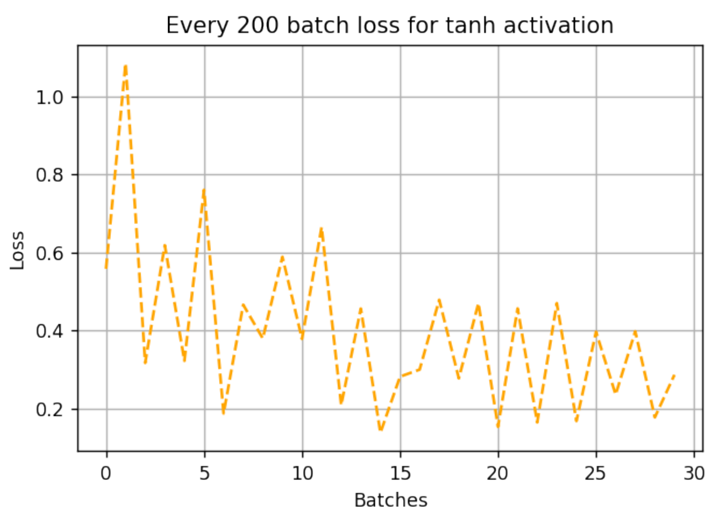


Figure 3: Batch Loss for with Tanh

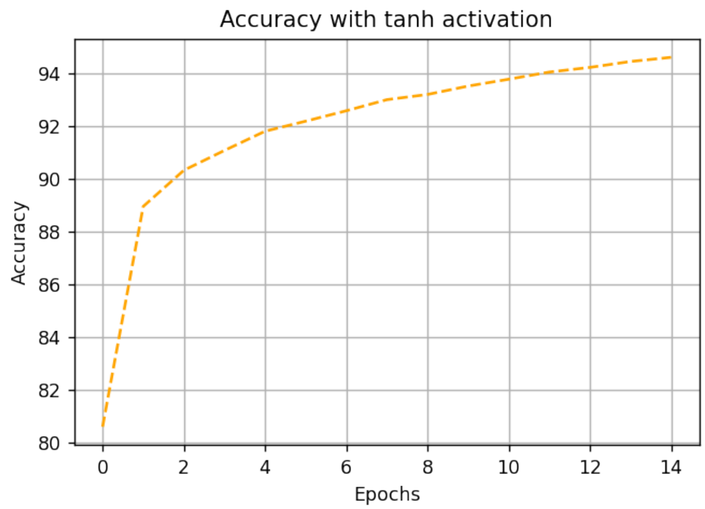


Figure 4: Accuracy on train set with Tanh

On test set the accuracy reaches high values, the test accuracy being 94.3%.

Activation function : ReLU

$$ReLU(x) = \max(0, x) \tag{3}$$

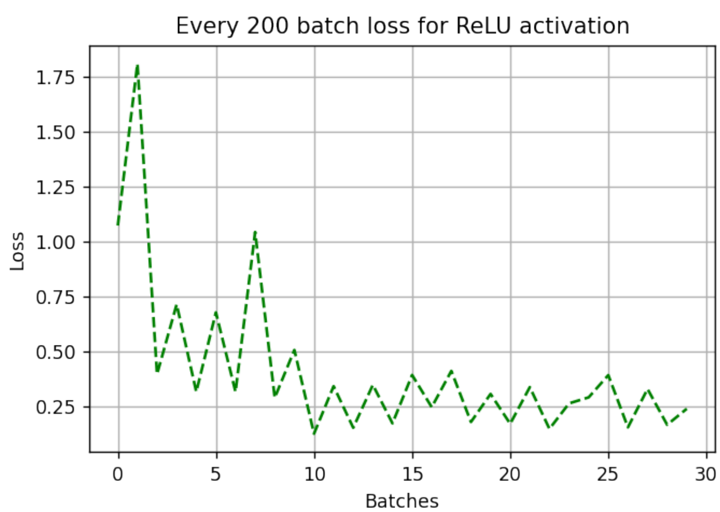


Figure 5: Batch Loss for with ReLU

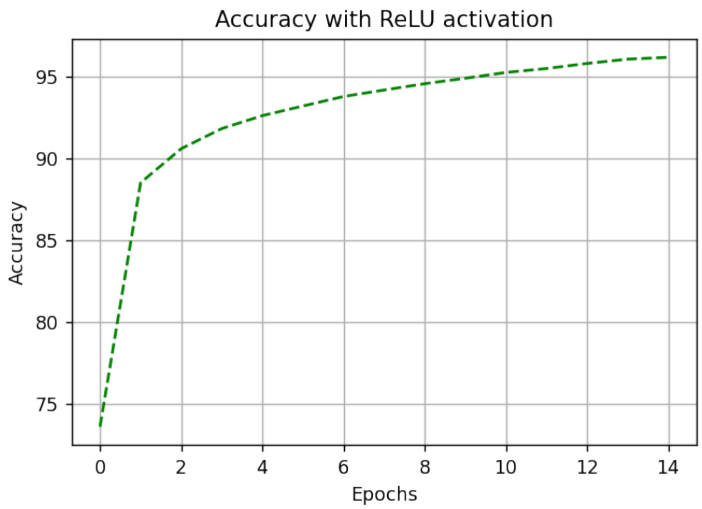


Figure 6: Accuracy on train set with ReLU

On train set the accuracy reaches 95.7%.
This proves that ReLU is the best activation for MLP classification on this dataset.

Confusion matrix:

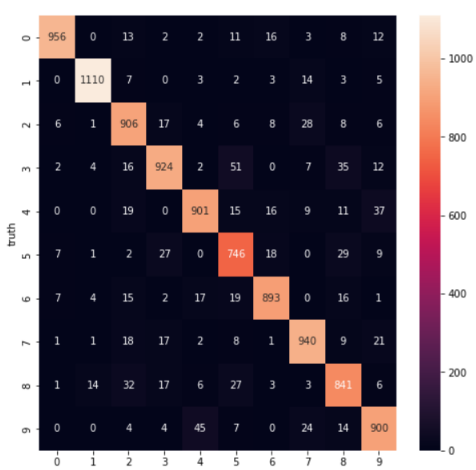
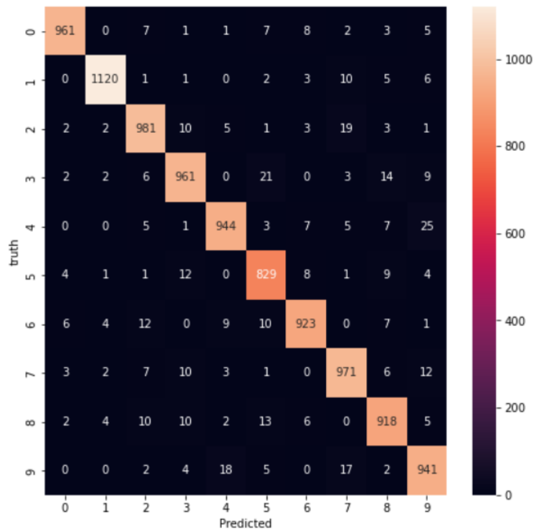
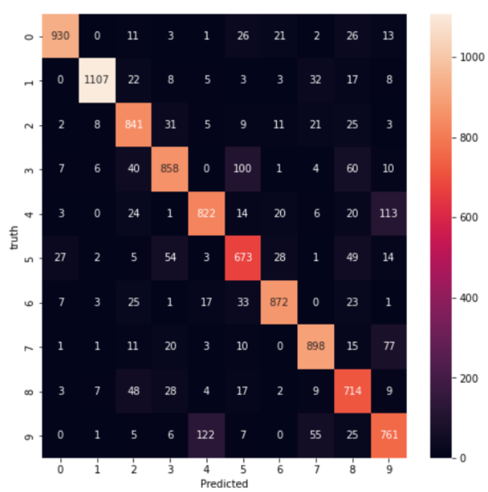


Figure 7: Confusion matrix for Relu



((a)) Confusion matrix for Tanh



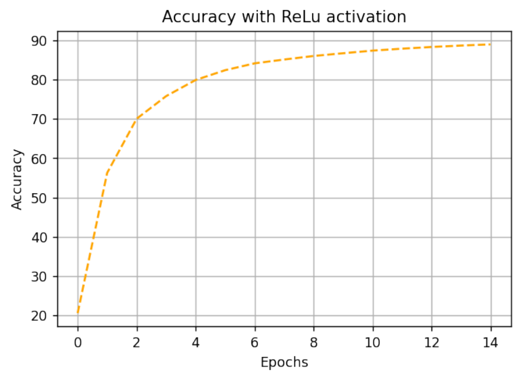
((b)) Confusion matrix for Sigmoid

Figure 8: Confusion matrix

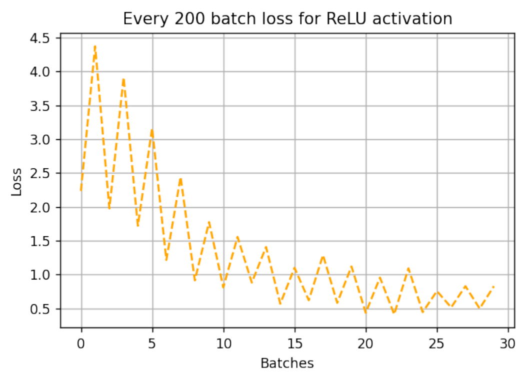
1.2 Backpropagation with two different learning rates

I have tried to train the model on all of the 3 activation functions with learning rate $\eta = 0.001$ and learning rate $\eta = 0.1$. Below are the results.

1.2.1 Learning rate: $\eta = 0.001$



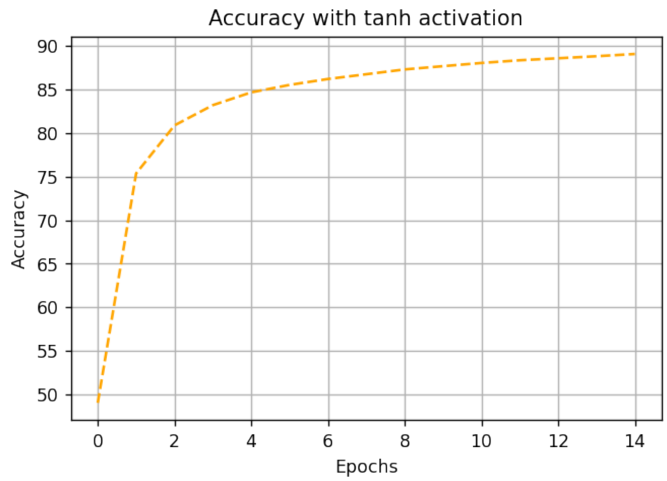
((a)) Accuracy graph for ReLU



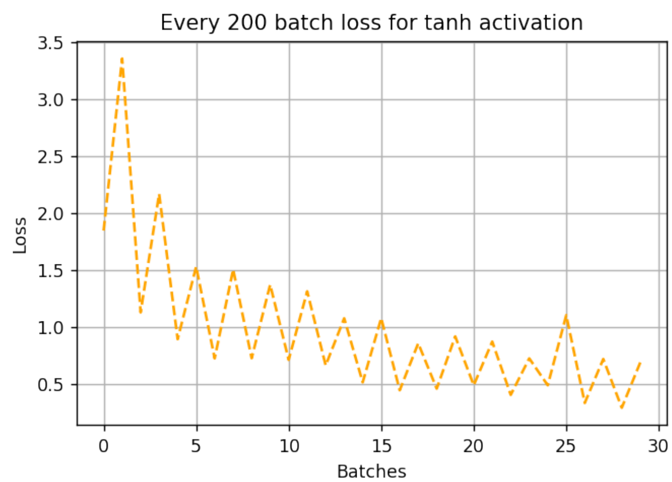
((b)) Loss graph for ReLu

Figure 9: Metrics with ReLU

The test accuracy is 90%.



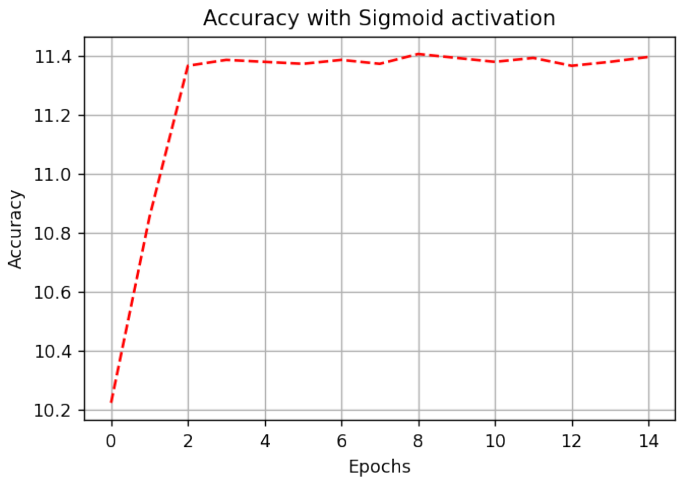
((a)) Accuracy graph for Tanh



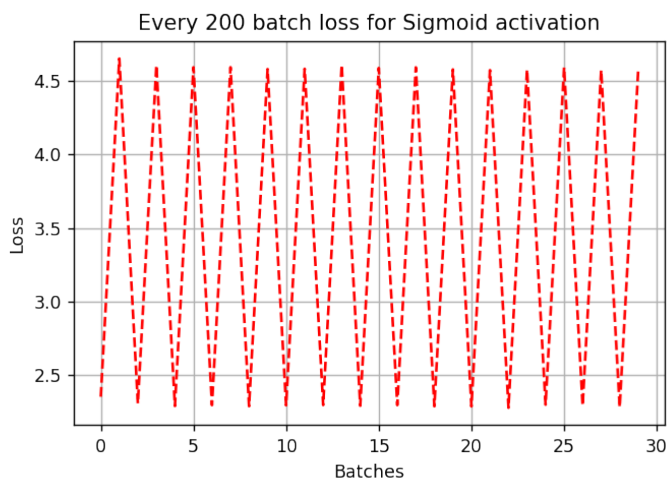
((b)) Loss graph for Tanh

Figure 10: Metrics with Tanh

The test accuracy is 89%.



((a)) Accuracy graph for Sigmoid

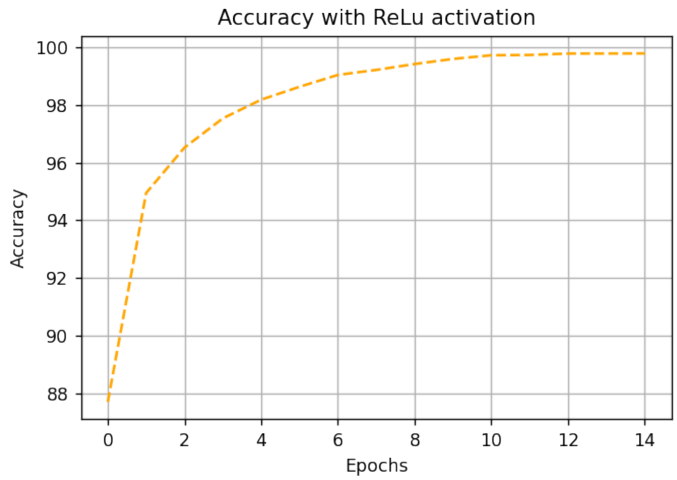


((b)) Loss graph for Sigmoid

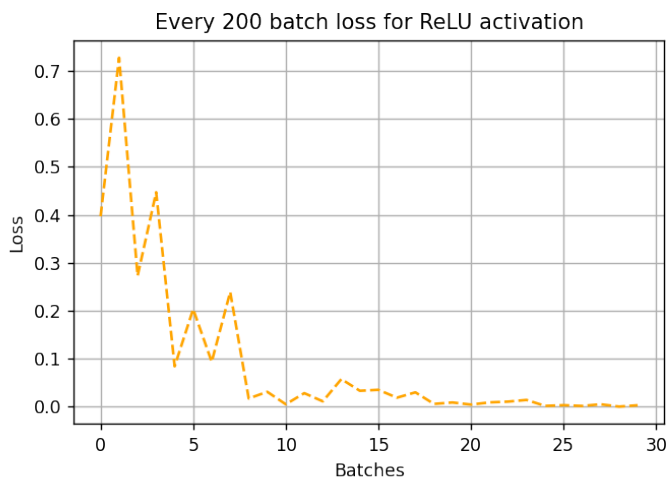
Figure 11: Metrics with Sigmoid

The test accuracy is 11.3%.
Thus we can conclude that this learning rate is not efficient for efiicient results, thus I took a higher learning rate of 0.1.

1.2.2 Learning rate: $\eta = 0.1$



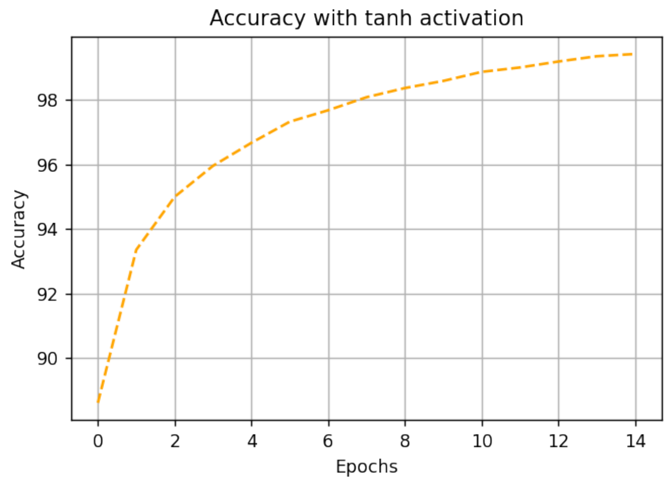
((a)) Accuracy graph for ReLU



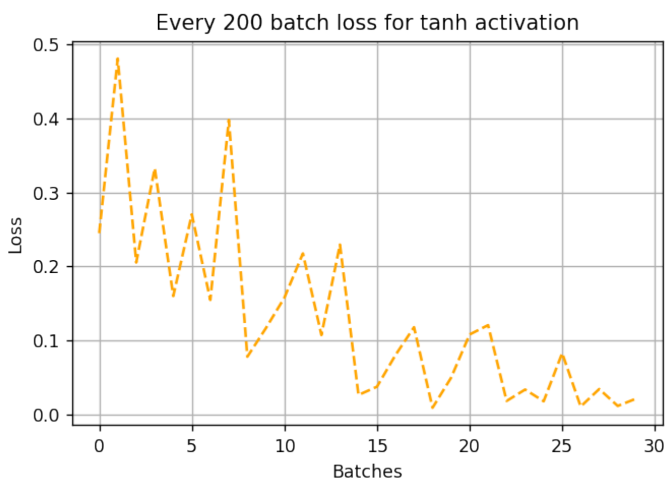
((b)) Loss graph for ReLU

Figure 12: Metrics with ReLU

The test accuracy reaches 97.8%.



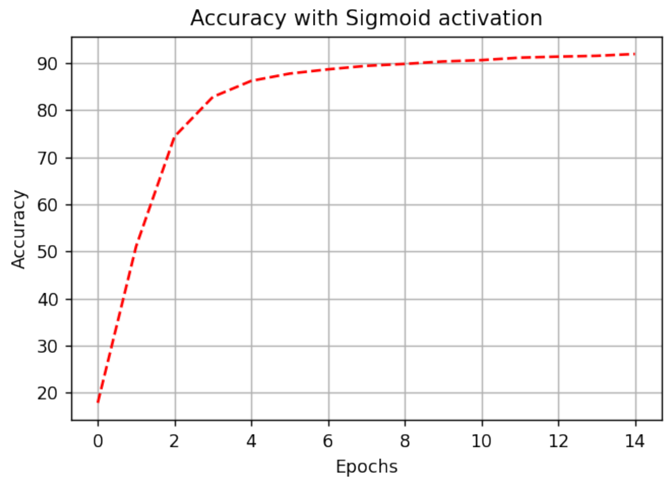
((a)) Accuracy graph for Tanh



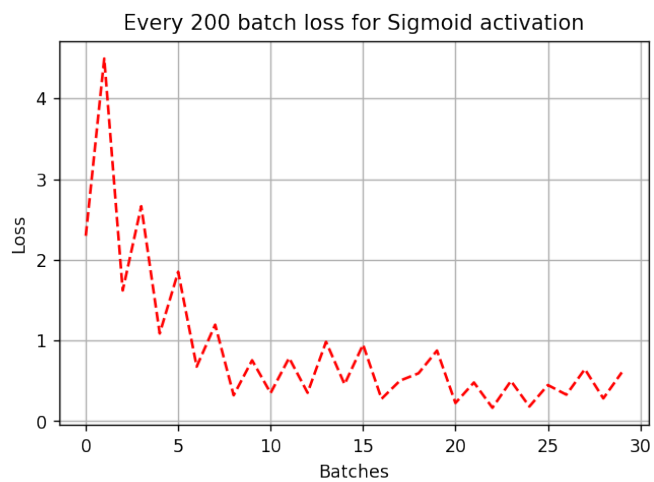
((b)) Loss graph for Tanh

Figure 13: Metrics with Tanh

The test accuracy reaches 97%.



((a)) Accuracy graph for Sigmoid



((b)) Loss graph for Sigmoid

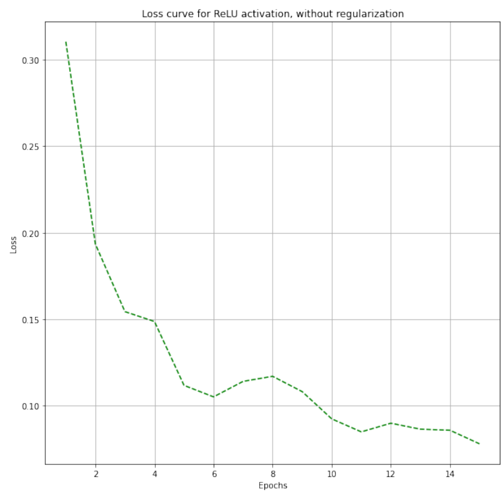
Figure 14: Metrics with Sigmoid

The test accuracy reaches 92%.
Thus a 0.1 learning rate is better in our baseline model compared to the one given.

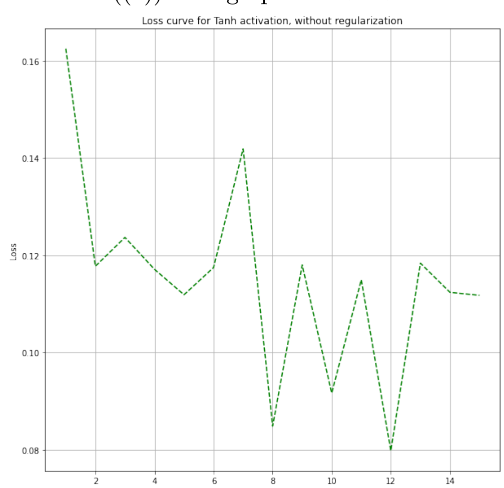
2 Package: Using Pytorch on the same baseline architecture

2.1 Using pytorch without Regularization

The loss curves for different activation functions are shown below:



((a)) Loss graph for ReLU



((b)) Loss graph for Tanh

Figure 15: Loss of Baseline model with different activation function

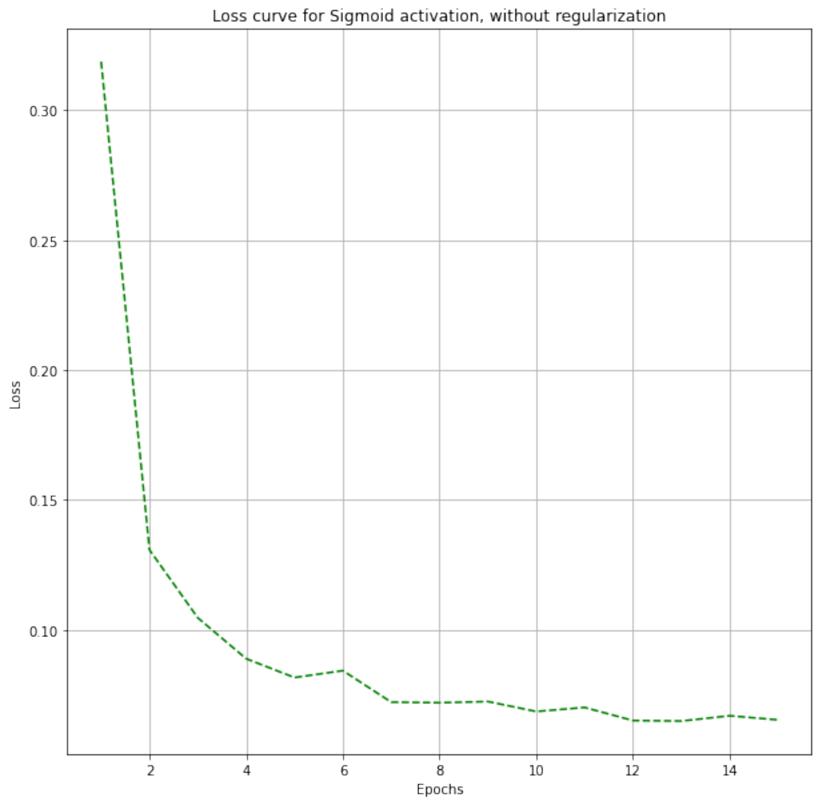


Figure 16: Loss graph for Sigmoid

The test accuracy for ReLU activation is 97%, for Tanh activation is 96% and for sigmoid is 93%, without regularization.

2.2 Using pytorch with L2 Regularization

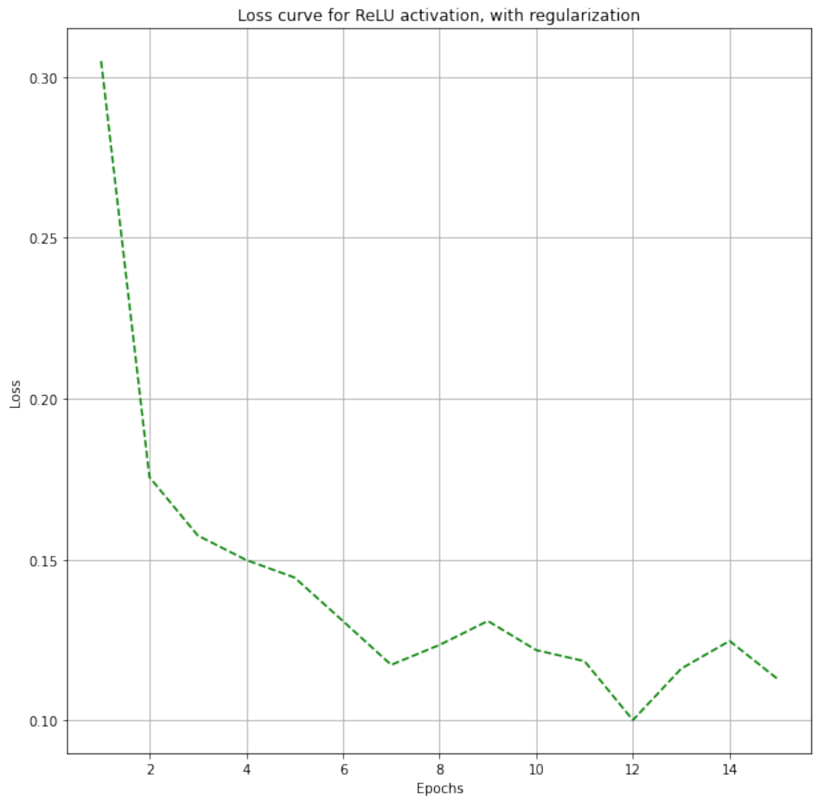


Figure 17: Loss graph for ReLU

The test accuracy was 96.3%.Here I have added a weight decay term of value 10^{-5},Larger weight decays just increases the train loss value.

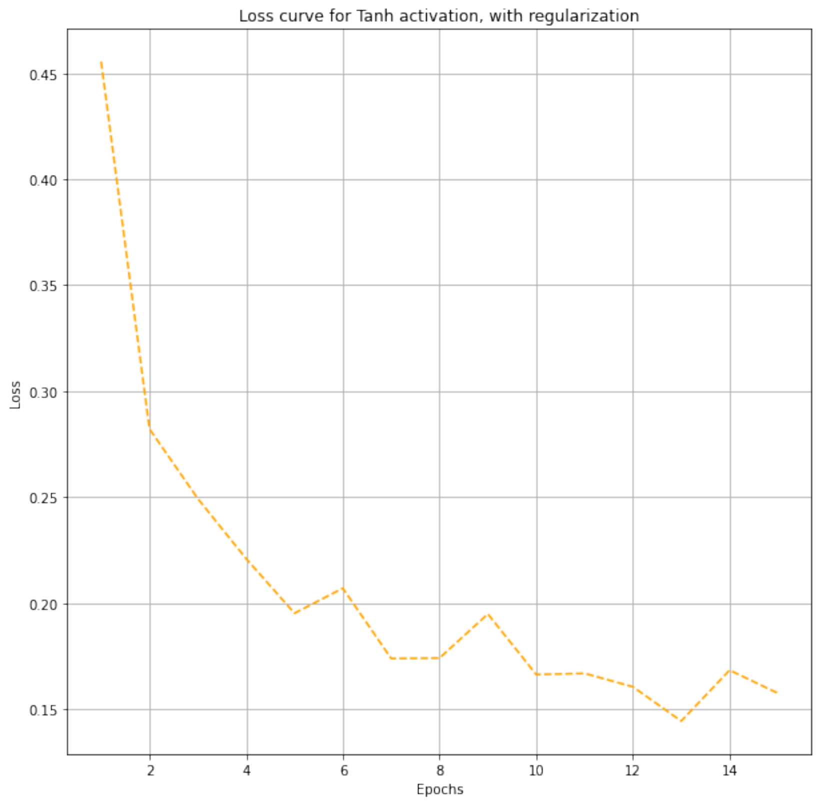


Figure 18: Loss graph for Tanh

The test accuracy is 96.3% thus model has improves over the test dataset.The weight decay paramater, i,e the λ=10^{-5} .

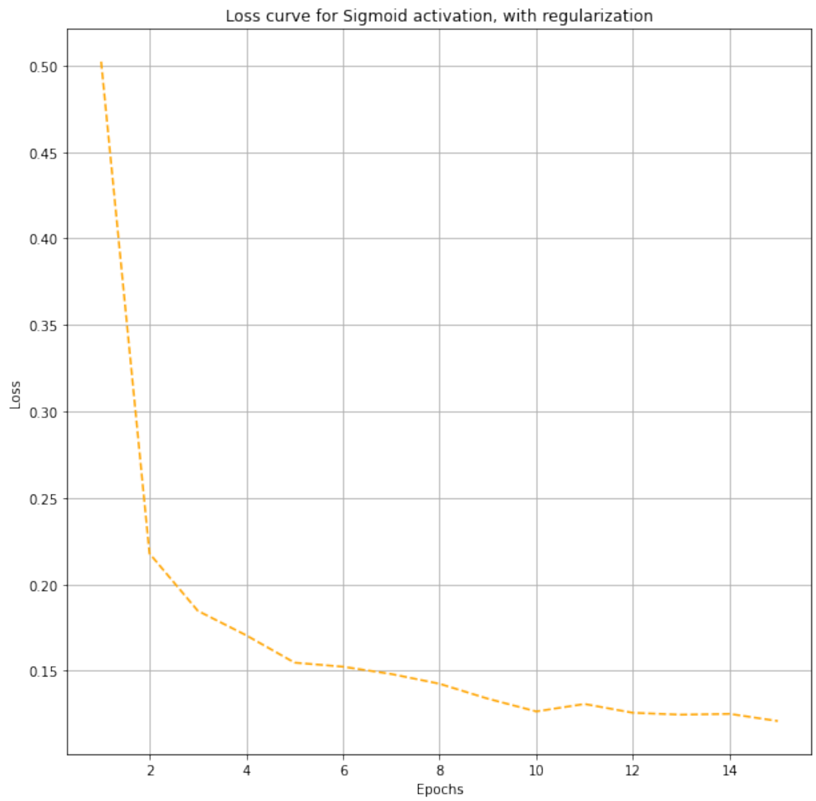


Figure 19: Loss graph for Sigmoid

The test accuracy has reached 95%.
Thus with L2 regularization there is a overall increase in test accuracy which was expected from theory.