# Intro to Machine Learning

## Q1a.

For gradient checking not all the hyperparameters passed the test and the following was the output:-

```
Gradient check did not pass
Correct
Gradient check did not pass
Gradient check did not pass
Correct
Gradient check did not pass
Gradient check did not pass
Correct
Gradient check did not pass
Gradient check did not pass
Correct
Gradient check did not pass
Gradient check did not pass
Gradient check did not pass
Gradient check did not pass
Gradient check did not pass
Gradient check did not pass
Gradient check did not pass
Gradient check did not pass
Gradient check did not pass
Correct
Correct
Correct
Correct
Correct
Correct
Correct
Correct
Correct
```

Correct

Correct

Correct

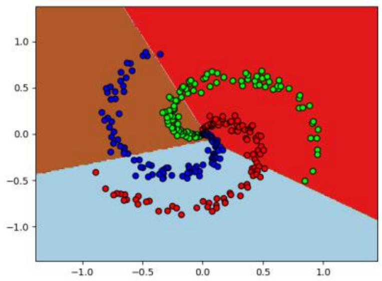
Accuracy = 75.0(Highest accuracy achieved)

Final loss = 0.6931757845690542

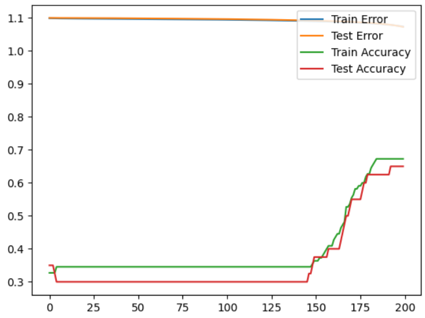
Q1b.

Training Accuracy = 57.333333333333336

Initially with only 3 to 8 hidden layers the model would only predict output for one class correctly, but with increase in the non-linearity of the model and increasing the number of epochs, the accuracy increased and the following decision boundary for the dataset was plotted by the model



Q1c.



Highest Accuracy achieved in Train = 67.27272727272726

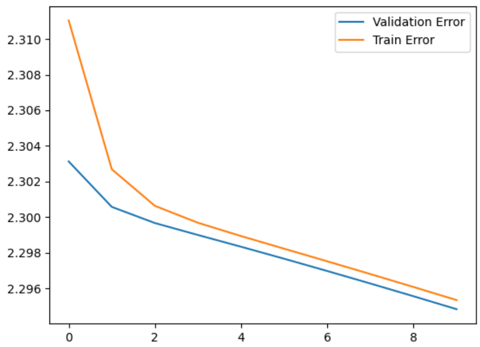
Highest Accuracy achieved in Test = 65

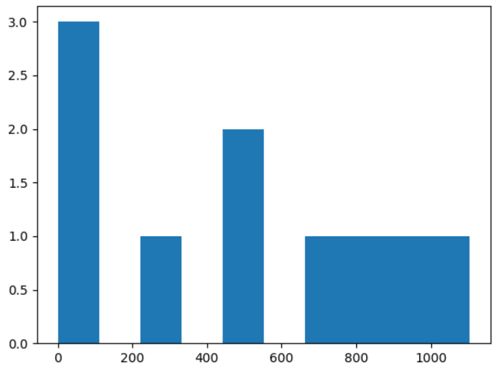
Since I used sigmoid function the model suffers with the problem of vanishing gradients hence when I increase the number of epochs the gradients vanish.

One other problem it faces is the problem of overfitting, since the amount of data is very low and if you increase the epochs, learning-rate and even the neurons in the hidden layer the model will suffer from overfitting.

With after applying L2 weight decay the gradient descent slows down and a smoother gradient descent is observed and this improves generalization.

Q2.





| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Predicted digits | [881 | 1129 | 517 | 766 | 77 | 0 | 434 | 962 | 220 | 0] |
| Actual number | [980 | 1135 | 1032 | 1010 | 982 | 892 | 958 | 1028 | 974 | 1009] |

Accuracy = 49.86

Initially with lesser number of epochs and hidden layers the accuracy was only 10%-12%, the loss was extremely high and barely reduced. If the number of neurons in the hidden layers, learning rate or epochs are too high then the model starts to overfit where the validation loss starts to increase, hence to stop overfitting early stopping was applied where if the loss increases beyond the given threshold the model will stop updating the hyperparameters like weights and biases.

With sufficient number of neurons in the hidden layer, and appropriate learning rate you can get a decent performance out of the model. The above-mentioned accuracy was the maximum accuracy I could achieve, the predicted classes are presented corresponding to the digits they belonged to along with the actual number of images in those classes. With some more tuning like increase epoch and decreasing learning rate and increasing the L2 penalty we might be able to get the performance of the model up to 70%-80% maybe more.