For Q1 and Q2, you need to use IRIS dataset available at - https://archive.ics.uci.edu/ml/datasets/iris

Q1. Implement two perceptron model to perform classification on iris database from scratch. You should not be using any inbuilt function for this implementation (except reading the data).

(a) vary the learning rate and show the best learning rate value when your run it for 50 epochs. **lr = 0.1** `correct: 95, incorrect: 5`

(b) vary the number of epochs from 10 to 100 in a step of 10 and show the loss value curve (using the best learning rate obtained from (a))

```
EPOCH NUMBER: 10
correct: 52
incorrect: 48
[5.2, 4.1, 1.5, 0.1, 0.0]
EPOCH NUMBER: 20
correct: 90
incorrect: 10
[5.2, 4.1, 1.5, 0.1, 0.0]
EPOCH NUMBER: 30
correct: 97
incorrect: 3
[5.2, 4.1, 1.5, 0.1, 0.0]
EPOCH NUMBER: 40
correct: 98
incorrect: 2
[5.2, 4.1, 1.5, 0.1, 0.0]
EPOCH NUMBER: 50
correct: 99
incorrect: 1
[5.2, 4.1, 1.5, 0.1, 0.0]
EPOCH NUMBER: 60
correct: 99
```

```
incorrect: 1
[5.2, 4.1, 1.5, 0.1, 0.0]
EPOCH NUMBER: 70
correct: 99
incorrect: 1
[5.2, 4.1, 1.5, 0.1, 0.0]
EPOCH NUMBER: 80
correct: 100
incorrect: 0
[5.2, 4.1, 1.5, 0.1, 0.0]
EPOCH NUMBER: 90
correct: 100
incorrect: 0
[5.2, 4.1, 1.5, 0.1, 0.0]
EPOCH NUMBER: 100
correct: 100
incorrect: 0
[5.2, 4.1, 1.5, 0.1, 0.0]
```

Q2. Implement a 3-class backpropagation NNet on your own to classify iris data, i.e. from scratch. You should not be using any inbuilt function for this implementation (except reading the data).

(a) vary the learning rate and show the best learning rate value when your run it for 50 epochs.

EPOCHS = 50

ETA = 0.01 **(fixed when giving best accuracy)**

Calculated accuracy : 97.32268998830814

EPOCHS = 50

ETA = 0.001 **(fixed when giving best accuracy)**

calculated accuracy = 97.32376017678327

(b) vary the number of epochs from 10 to 100 in a step of 10 and show the loss value curve (using the best learning rate obtained from (a)) (c) add L2 regularization - show the comparisons with and without this regularization and analyze your results.

epoch_number 10

Overall Accuracy38.956008277885815

epoch_number 20

Overall Accuracy45.86043377627047

epoch_number 30

Overall Accuracy51.878865588825974

epoch_number 40

Overall Accuracy54.63512274668732

epoch_number 50

Overall Accuracy55.7723636783644

epoch_number 60

Overall Accuracy56.19781157146795

epoch_number 70

Overall Accuracy56.369467047439414

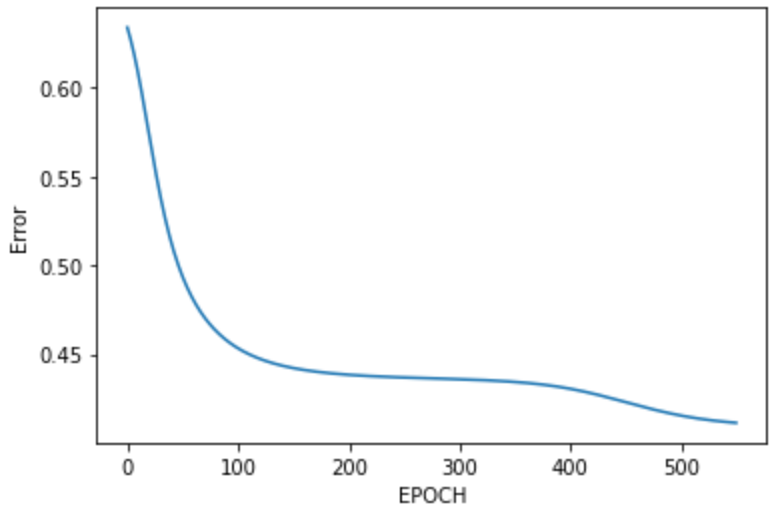
epoch_number 80

Overall Accuracy56.609013922077025

epoch_number 90

Overall Accuracy57.69299964246666

epoch_number 100



Overall Accuracy58.863831669270205

```
EPOCH WISE:
[38.956008277885815,
 45.86043377627047,
 51.878865588825974,
 54.63512274668732,
 55.7723636783644,
 56.19781157146795,
 56.369467047439414,
 56.609013922077025,
 57.69299964246666,
 58.863831669270205]
```

Q3. Use any toolbox in python and implement RBF NNet to solve one of the problems/databases (of your choice from the UCI ML database Repo). Analyze your results with respect to varying learning rate and epochs. You are not allowed to use someone's code available online. UCI databases: **The accuracy is 0.9736842105263158**

Q4. Using MNIST database, code Autoencoder model with three encoding and three decoding layers. Show the visualization of the feature maps. On the features, add a classifier to perform 10-class classification and show the training loss curve and test accuracy. **Show in notebook**