CA Assignment 1

Explain the Perceptron algorithm

The binary perceptron identifies the first class read as 1 and the second as -1 and calculates it, based on the calculated weights and deviations (train data) and applies them to the test data using the weights and biases from the training data.

```
PerceptronTrain(Data, MaxIter) #pseudo-code is from the lecture
    w = 0 for all i = 1,...d
    b = 0 #set weight and bias as 0
    for iter =1....MaxIter do
        random data
        for r in data:
            a = (W,X) + b #compute activation value
            if y*a <= 0 then
                w = w + y*x, for all i = 1,...,d
                b = b + y
    Return w,b
```

```
PerceptronTest(Data, Weight, Bias):#Test pseudo-code
    num_correct = 0
    a = (Weight,X) + Bias # compute activation value
    if y == sign(a) #if sign(a) equals to y, +1. otherwise +0
        num_correct = num_correct + 1
    Return num_correct/len(data) # Return accuracy
```

Result explain

```
========Classifier for class 1 and class 2========
Train accuracy: 1.00
Test accuracy: 1.00
========Classifier for class 2 and class 3========
Classifier for class 2 and class 3:
Train accuracy: 0.89
Test accuracy: 0.90
========Classifier for class 1 and class 3========
Classifier for class 1 and class 3:
Train accuracy: 0.94
Test accuracy: 0.92
```

Which pair of classes is most difficult to separate?

Following the result above, it is manifest that class 2 and class 3 are the most difficult to separate. But to be fair and ensure my answers were accurate, I eliminated the seed that guaranteed the same random result and iterated the program 2000 times.

In 1617 times, class 1 and class 3 are the most difficult to separate. In 383 times, class 2 and class 3 are the most difficult to separate.

So although in my seed is class 2 and class 3, but class 1 and class 3 should be the most difficult to separate.

One-Vs-Rest approach

One-vs-rest is a multi-class classification algorithm that splits a multiplexed dataset into a binary classification problem, setting one class to 1 and all the rest to -1. Then train it with the following results

```
========Classifier for class-1 vs the rest========
Train accuracy: 0.95
Test accuracy: 0.90
========Classifier for class-2 vs the rest========
Train accuracy: 0.91
Test accuracy: 0.92
========Classifier for class-3 vs the rest========
Train accuracy: 0.67
Test accuracy: 0.66
```

Obviously, class-3 is the most difficult to separate with the rest.

l_2 regularisation

Set the regularisation coefficient to 0.01

```
========Classifier for class-1 vs the rest========
Train accuracy(regularisation coefficient): 1.00
Test accuracy: 1.00
========Classifier for class-2 vs the rest========
Train accuracy(regularisation coefficient): 0.93
Test accuracy: 0.95
========Classifier for class-3 vs the rest========
Train accuracy(regularisation coefficient): 0.84
Test accuracy: 0.82
```

Set the regularisation coefficient to 0.1

```
========Classifier for class-1 vs the rest========
Train accuracy(regularisation coefficient): 1.00
Test accuracy: 1.00
========Classifier for class-2 vs the rest========
Train accuracy(regularisation coefficient): 0.87
Test accuracy: 0.90
========Classifier for class-3 vs the rest========
Train accuracy(regularisation coefficient): 0.94
Test accuracy: 0.95
```

Set the regularisation coefficient to 1.0

```
========Classifier for class-1 vs the rest========
Train accuracy(regularisation coefficient): 0.49
Test accuracy: 0.49
========Classifier for class-2 vs the rest========
Train accuracy(regularisation coefficient): 0.50
Test accuracy: 0.50
========Classifier for class-3 vs the rest========
Train accuracy(regularisation coefficient): 0.50
Test accuracy: 0.51
```

Set the regularisation coefficient to 10.0

```
========Classifier for class-1 vs the rest========
Train accuracy(regularisation coefficient): 0.00
Test accuracy: 0.00
========Classifier for class-2 vs the rest========
Train accuracy(regularisation coefficient): 0.12
Test accuracy: 0.10
========Classifier for class-3 vs the rest========
Train accuracy(regularisation coefficient): 0.07
Test accuracy: 0.15
```

Set the regularisation coefficient to 100.0

```
========Classifier for class-1 vs the rest========
Train accuracy(regularisation coefficient): 0.00
Test accuracy: 0.00
========Classifier for class-2 vs the rest========
Train accuracy(regularisation coefficient): 0.39
Test accuracy: 0.47
========Classifier for class-3 vs the rest========
Train accuracy(regularisation coefficient): 0.07
Test accuracy: 0.15
```

Based on these results, we can see that the regularisation coefficient significantly impacts the classifier algorithm. Furthermore, for the class-1 vs the rest and class-2 vs the rest classifier, regularisation factor of more than 0.1 leads to a rapid decrease in accuracy, while the accuracy of the first dataset goes to exactly zero at a regularisation factor of 10, which indicates that too high a regularisation factor leads Classifier for class-1 vs the rest to an underfitting situation, while the Classifier for class-3 vs the rest does not change at regularisation factors of 10 and 100, which suggests that it has less effect on this classifier.