

# Report for COMP307 Assignment2

## Part1

**Wenyu Wang**

**300438451**

## Q1

```
First instance has label Adelie, which is [0] as an integer, and [1. 0. 0.] as a list of outputs.  
Predicted label for the first instance is: ['Chinstrap']
```

The output and predicted class of the first instance in the dataset using the provided weights are Chinstrap and 1 respectively.

## Q2

The updated weight after single BP

```
epoch = 0  
Hidden layer weights  
[[-0.28005354 -0.22016336]  
 [ 0.07983514  0.19949696]  
 [-0.30011828  0.31963908]  
 [ 0.09993612  0.00980508]]  
Output layer weights  
[[-0.27752534  0.04140892  0.22247679]  
 [ 0.0941994   0.1429863  -0.34579818]]
```

## Q3 & Q4

```
After training:  
Hidden layer weights:  
[[ 0.93300053 -9.81238895]  
 [-7.29027973  5.20341616]  
 [ 2.38938873 -1.40616717]  
 [ 2.47148091  1.43004753]]  
Output layer weights:  
[[ -9.67263879 -2.44486417  3.24212769]  
 [  4.90797584 -2.87370739 -11.64832746]]  
  
After testing:  
Test_accuracy = 0.8153846153846154
```

After 100 epochs training with accuracy at 82%, the test accuracy is about 81% and the final weights are showed above. In addition, I trained the algorithm 1000

epochs and the results was

```
epoch = 999
Hidden layer weights
[[ 3.04594562 -12.16876231]
 [-9.39229382  6.93323616]
 [ 1.94315446 -2.30467572]
 [ 2.06906095  2.22224285]]
Output layer weights
[[-24.55986409 -2.66758226  3.65290999]
 [ 6.15109814 -3.07907527 -28.52890961]]
acc = 0.9067164179104478

After training:
Hidden layer weights:
[[ 3.04594562 -12.16876231]
 [-9.39229382  6.93323616]
 [ 1.94315446 -2.30467572]
 [ 2.06906095  2.22224285]]
Output layer weights:
[[-24.55986409 -2.66758226  3.65290999]
 [ 6.15109814 -3.07907527 -28.52890961]]

After testing:
Test_accuracy = 0.8923076923076924
```

Training accuracy was 90% and test accuracy is 89%. That means more epochs can have better accuracy.

Then I added bias to the network, here is the result after trained 100 epochs with bias:

```
epoch = 99
Hidden layer weights
[[ -2.26419396 -10.8864365 ]
 [-8.15086237  5.65253654]
 [ 3.13357325 -1.29208228]
 [ 3.94607355  1.95545857]]
Output layer weights
[[-6.1699875  -3.97682384  4.29961169]
 [ 4.73281889 -3.63457323 -11.27487329]]
acc = 0.917910447761194
```

```

After training:
Hidden layer weights:
[[ -2.26419396 -10.8864365 ]
 [ -8.15086237  5.65253654]
 [  3.13357325 -1.29208228]
 [  3.94607355  1.95545857]]
Output layer weights:
[[ -6.1699875  -3.97682384  4.29961169]
 [  4.73281889 -3.63457323 -11.27487329]]

After testing:
Test_accuracy =  0.9076923076923077

```

The training accuracy is 91% and the test accuracy is 90%. Also I did a 1000 epochs training.

```

epoch = 999
Hidden layer weights
[[ -6.21915584 -14.21085361]
 [-14.06817531  7.68974885]
 [  6.41368741 -1.71455083]
 [  8.61202855  3.46286158]]
Output layer weights
[[ -6.37061745  -5.30148738  10.13402138]
 [  4.01781234 -3.66649131 -26.84034958]]
acc =  0.9589552238805971

```

```

After training:
Hidden layer weights:
[[ -6.21915584 -14.21085361]
 [-14.06817531  7.68974885]
 [  6.41368741 -1.71455083]
 [  8.61202855  3.46286158]]
Output layer weights:
[[ -6.37061745  -5.30148738  10.13402138]
 [  4.01781234 -3.66649131 -26.84034958]]

After testing:
Test_accuracy =  0.9384615384615385

```

In this case the training accuracy is 95% and test is 93%. It may overfitted, because it performed well in training than test. Over all I am satisfied with the network

performance. More epochs can lead to higher accuracy but can also cause overfitted.