COMP307 Assignment 2: ANN
Rohan Gajadhar 300562245

Part 1)

a) *Report the output and predicted class of the first instance in the dataset using the provided weights.*

```
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

epoch = 0
Initial Hidden layer weights:
[-0.28, -0.22]
[0.08, 0.2]
[-0.3, 0.32]
[0.1, 0.01]
[-0.02, -0.2]

Initial Output layer weights:
[-0.29, 0.03, 0.21]
[0.08, 0.13, -0.36]
[-0.33, 0.26, 0.06]

Training accuracy = 0.0
```

b) *Report the updated weights of the network after applying a single back-propagation based on only the first instance in the dataset.*

```
Weights after performing BP for first instance only:
Hidden layer weights:
[-0.28052064067333937, -0.219718347073908]
[0.07839682135470441, 0.200867277528664]
[-0.30115025265040085, 0.3206222564646219]
[0.09937879128267824, 0.010336057595779677]
[-0.023329678724844684, -0.19819873128662094]

Output layer weights:
[-0.27752533507224475, 0.017579233592740794, 0.19865828140016373]
[0.09419939713573042, 0.11586195332955135, -0.37290981100762555]
[-0.30387256101445353, 0.2339854482393872, 0.03624544928263795]
```

c) *Report the final weights and accuracy on the test set after 100 epochs. Analyse the test accuracy and discuss your thoughts.*

```
epoch = 99
Training accuracy = 0.8283582089552238

After training:
Test accuracy: 0.8153846153846154
Finished!

Final weights
Hidden layer weights:
[0.9330005251044311, -9.812388946972462]
[-7.290279730413504, 5.203416161081893]
[2.3893887252132866, -1.4061671732881285]
[2.471480908006519, 1.4300475252498666]
[-10.38768963341311, 9.25263290305635]

Output layer weights:
[-9.672638785504889, -2.4448641684415735, 3.242127685441499]
[4.907975842091944, -2.8737073926728973, -11.648327456078443]
[-62.71587793726595, 81.54218476562394, -55.071524719567954]
```

The test accuracy is 81% after 100 epochs. This is an appropriate test accuracy. The training accuracy after 99 epochs is 82%. As the training and test accuracies are in close proximity to each other, I am inclined to say that the model is performing well. A disparity between training and testing accuracies would signify overfitting, underfitting or in extreme situations - both.

d) *Discuss how your network performed compared to what you expected. Did it converge quickly? Do you think it is overfitting, underfitting or neither?*
The network exceeded my expectations for performance. The network converged quickly with a training accuracy of 51% on epoch 0, and a training accuracy of 79% on epoch 7. It then settles and slowly increases training accuracy to 82% on epoch 99.

After 100 epochs, the testing accuracy was 81%. This **does not** appear to signify overfitting or underfitting. If the model was overfitting then the training accuracy would be significantly higher than the testing accuracy. If the model was underfitting then the training accuracy would be significantly lower than the testing accuracy.

Part 1.1)

a) *Reported test accuracy with bias.*

```
epoch = 99
Training accuracy = 0.9925373134328358

After training:
Test accuracy: 1.0
Finished!

Final weights
Hidden layer weights:
[-1.4112514916433252, -11.167712402394596]
[-6.1961745489218885, 5.366538324449307]
[4.203701222306227, -0.8357352108606549]
[4.653345223701814, 2.5153360448519573]
[-1.3561086593772536, 0.13615236100081476]

Output layer weights:
[-2.7157512445797782, -7.036719980471046, 7.551222654831535]
[9.250986865160149, -8.182727938954367, -5.4842032686676285]
[-3.7458004388271706, 3.7834457669226054, -2.7802333817152034]
```

b) *Compare the accuracy achieved to that of the original network, and discuss possible reasons for any performance differences.*
The training accuracy reaches 99% after 99 epochs and the testing accuracy is 100%. After implementing the biases the testing accuracy is perfect, and the training accuracy is **near** perfect.
A bias helps the neural network better fit the training data. It is a measure of how much the predicted values differ from the actual values. By introducing a bias, we are effectively shifting the activation function of a given neuron to the left or right. In turn, this makes it harder, or easier, for the neuron to fire.
During training, the weights and biases are adjusted to minimise the error between the predicted output and the actual output. The bias allows the neuron to adjust its output independently of the inputs. For example, if all the inputs are zero and the predicted output is 1, the bias will help the neuron output a non-zero value.
This is why we see a significant increase in accuracy where we have implemented biases, and a decrease in accuracy where we have not.