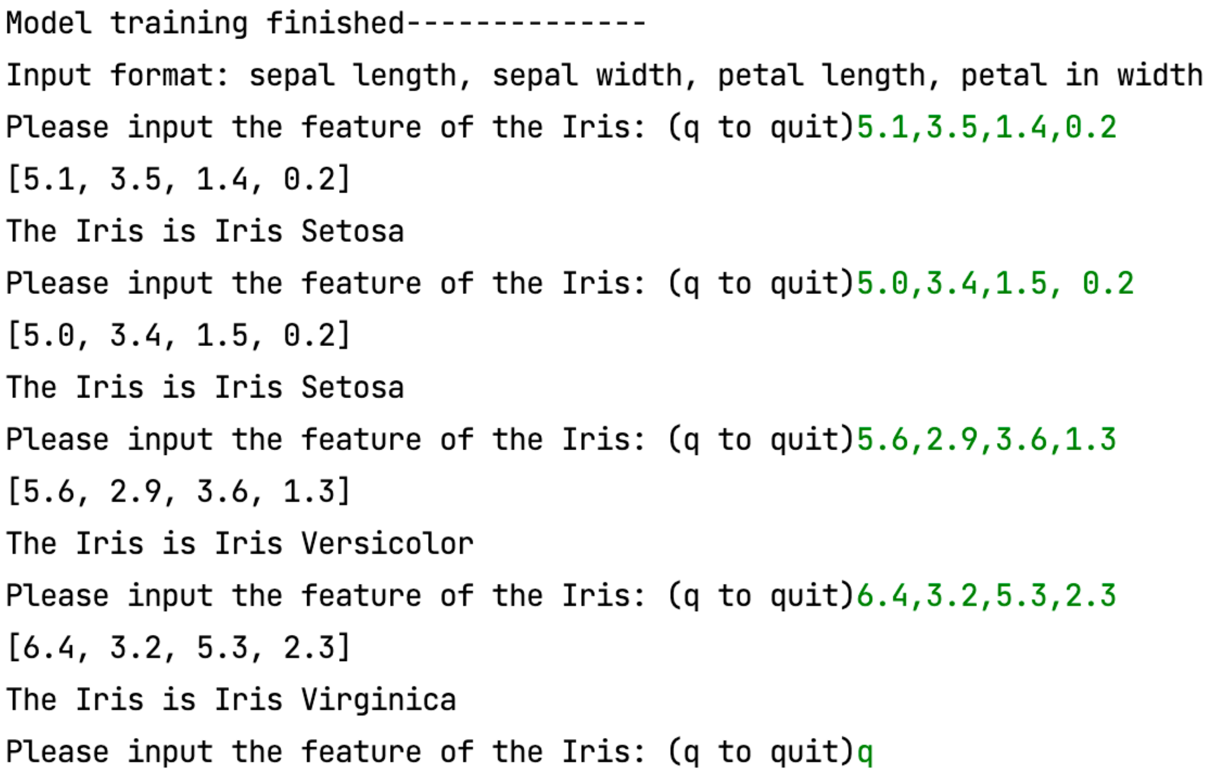
1 Environments and instructions

1. Python version **3.8** required
2. Run with `python neural_network.py`, and input the features of Iris according to the intructions.
3. The terminal will show prediction of the category to which the iris belongs.

2 Implementation Details

In my implementation, I use ***forward propagation*** and ***backward propagation*** to train the model and perform predictions. To build a neural network classifier that can adapt to large data sets, I perform hyperparameter tuning by cross-validation to avoid underfitting or overfitting. Also, by using the ***Sigmoid*** function as the activation function, the calculation of the associated gradient becomes simpler.



```
Model training finished--------------
Input format: sepal length, sepal width, petal length, petal in width
Please input the feature of the Iris: (q to quit)5.1,3.5,1.4,0.2
[5.1, 3.5, 1.4, 0.2]
The Iris is Iris Setosa
Please input the feature of the Iris: (q to quit)5.0,3.4,1.5, 0.2
[5.0, 3.4, 1.5, 0.2]
The Iris is Iris Setosa
Please input the feature of the Iris: (q to quit)5.6,2.9,3.6,1.3
[5.6, 2.9, 3.6, 1.3]
The Iris is Iris Versicolor
Please input the feature of the Iris: (q to quit)6.4,3.2,5.3,2.3
[6.4, 3.2, 5.3, 2.3]
The Iris is Iris Virginica
Please input the feature of the Iris: (q to quit)q
```

First I processed the data, especially the labels, into a form suitable for neural network processing. Then the model was trained continuously according to a preset number of iterations, and it was found that the model worked well enough when the number of iterations was around ***200***, so this number was subsequently used as a parameter to train the model. When I input the data from the original dataset to see how well my model works, I can find that the model has high accuracy for prediction as figure above shows.