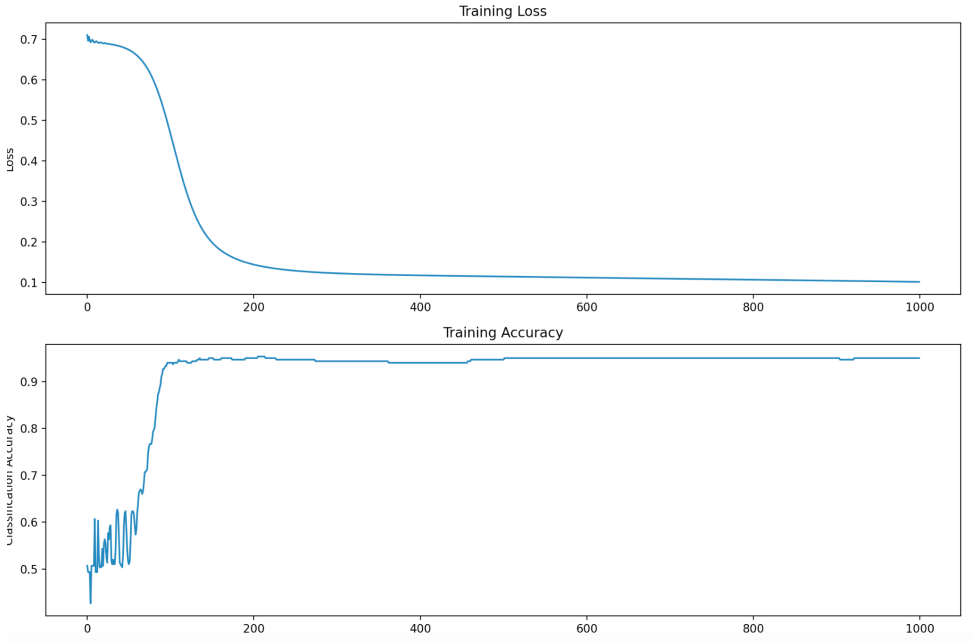
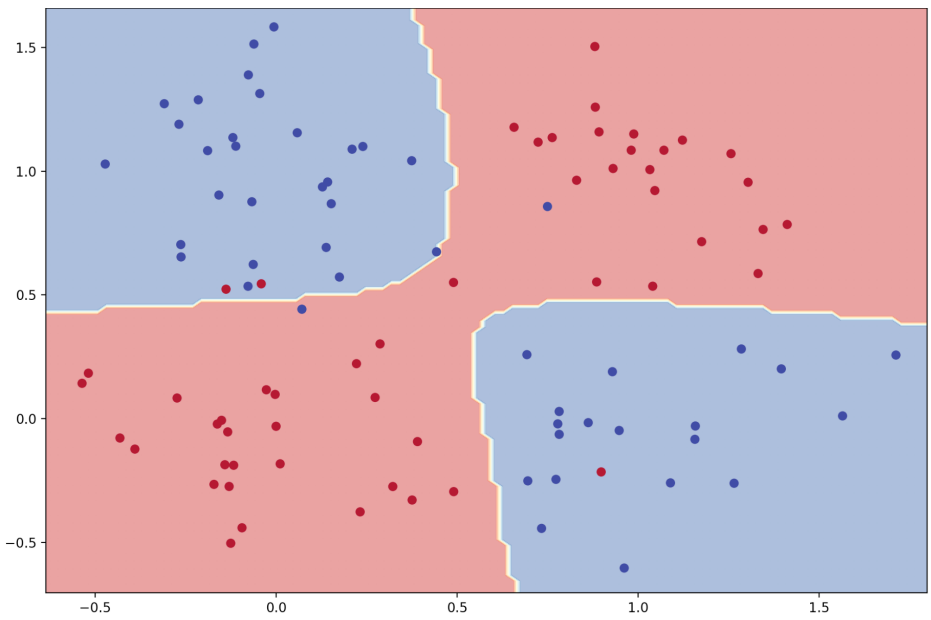
Erika Fox & Peining Yang
IDS 703: Natural Language Processing

IDS 703 HW07: Feed-Forward Neural Network

Question 1

After adding a second hidden layer with 100 nodes to the function, on average our testing accuracy is 95.5% after running the code 10 times. In general, our testing accuracy is always above 90%. The training process curve and decision surface with overload training data is shown below, the accuracy here was 95%.





Question 2

Our from-scratch neural network consistently got the same results as the pytorch network from the previous question. Here are screenshots of the terminal from various runs that demonstrate this, where “Test Accuracy” describes the pytorch network and “The accuracy of my neural network is” describes our numpy network.

```
The accuracy of my neural network is 95.0%
(base) MacBook-Pro-9:nlp3 erikafox$ /Users/erikafox/opt/miniconda3/bin/python /Users/erikafox/Desktop/ffnn/ffnn.py
Test Accuracy 87.00%
The accuracy of my neural network is 87.0%
(base) MacBook-Pro-9:nlp3 erikafox$ /Users/erikafox/opt/miniconda3/bin/python /Users/erikafox/Desktop/ffnn/ffnn.py
Test Accuracy 96.00%
The accuracy of my neural network is 96.0%
(base) MacBook-Pro-9:nlp3 erikafox$ /Users/erikafox/opt/miniconda3/bin/python /Users/erikafox/Desktop/ffnn/ffnn.py
Test Accuracy 97.00%
The accuracy of my neural network is 97.0%
(base) MacBook-Pro-9:nlp3 erikafox$ /Users/erikafox/opt/miniconda3/bin/python /Users/erikafox/Desktop/ffnn/ffnn.py
Test Accuracy 98.00%
The accuracy of my neural network is 98.0%
```

In order to achieve this, we went through the steps of a neural network using numpy operations instead of pytorch operations to build *yhat*, a numpy array of predicted probabilities for y-values. Namely, we built *yhat* by taking the layers and biases generated in question one and implementing the appropriate matrix multiplications and sigmoid functions. We are simply doing the same thing the pytorch network is doing, so we can expect the same results every time.

To complete the network once *yhat* was filled, we then defined *yval*, where when *yhat*[i] is greater than 0.5, *yval*[i] is 1, and 0 otherwise. From this, we could calculate the new accuracy by taking the total times *yhat*[i] is equal to *yval*[i] divided by the total length of Y_test, multiplied by 100.