Assignment 5

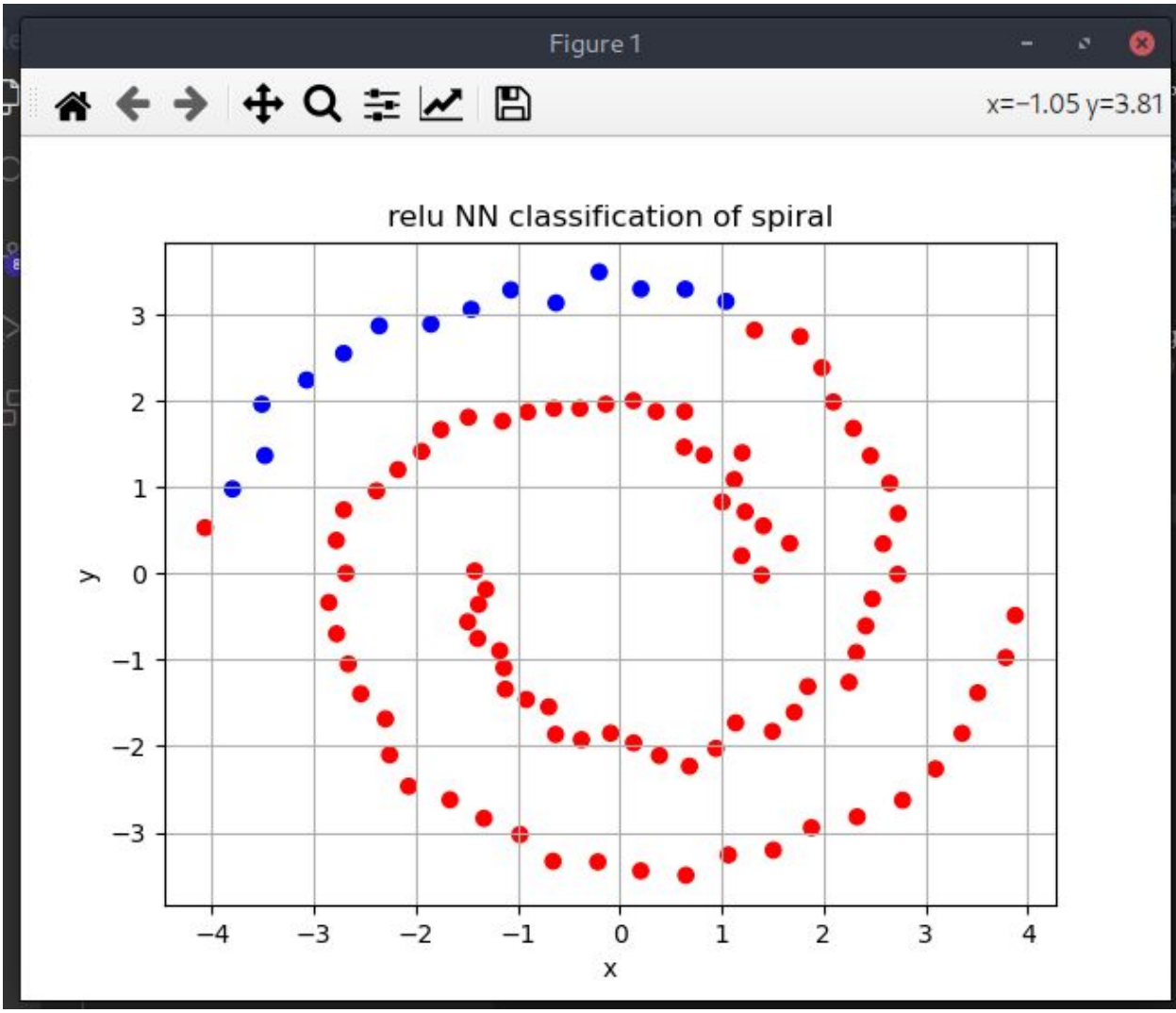
1.
   a. The output of the following code tells us that **50%** of the observations have SpectralCluster = 1

```
25
26  total = 0
27  cluster1 = 0
28  for c in df['SpectralCluster']:
29      if c == 1 or c == '1':
30          cluster1 += 1
31      total += 1
32
33  print('Opservations with SpectralCluster=1 : %s%%' % (100 * cluster1 / total)) # 50%
34
```

   b. See attached table.txt for complete list of tests performed

Activation Funtion	Hidden Layers	Neurons per layer	Iterations	Loss	Misclassification
identity	1	1	8	0.6662895167926476	0.25
logistic	3	3	1782	0.2426169017150364	0.35
relu	1	1	27	0.5847125309814049	0.14
tanh	1	2	35	0.5441035363834196	0.19

   c. Every single model used the ‘logistic’ output activation function
   d. Based on the table given in B it seems that the relu activation function gives the lowest misclassification rate and the logistic function gives the lowest loss value. I believe that the misclassification rate is more important so I’d argue that the best network to use is the relu activation function with one neuron and one layer with 27 iterations or row 3 in the table
   e. Here you can see that the classifier definitely isn’t perfectly accurate and the others don’t seem to be much better



2.

```
[tate@archbook assignment5]$ python q2.py
Mean Accuracy =  0.5
                   id         x         y  SpectralCluster
_PredictedClass_
0                60.16 -0.110362 -1.672481              0.5
1                40.84  0.076604  1.654779              0.5
Intercept =  [0.003345]
Coefficients =  [[0.05333512 0.32868383]]
```

a. Using the above information, you can see that the Coefficients are 0.05335 and 0.32868 and the intercept is 0.003345. Using these values you can construct the equation for the linear SVM

b. The mean accuracy was about 50%

c. The below plot was produced from the attached code