Q1

For testing the system we used a k fold with 10 splits for all the tests. We varied the number of nodes using 2 layers and the number of layers using 5 nodes. After testing we found that the error rate was approximately 90% which indicates that the neural network was just guessing randomly. It was changing its accuracy while using the same testing data which also indicates that the program was changing itself. We were unable to find proper numbers of nodes or layers which would allow us to increase the accuracy. Alternatively there could be a problem with the formula we used to change the weights. Testing ran for 6 hours for each of the different arrangements

Nodes	Layers	Best accuracy (% correct
10	2	10.242
5	2	10.314
20	2	9.228
5	1	10.271
5	3	10.057

Of all the possible arrangements the 5 node 1 layer was the only one which had enough completed runs through the training data to show signs of improvement. However it only did 4 testing cycles so it could easily be a coincidence. Testing was cut off due to lack of time to run complete experiments.

Q2
For testing we used K fold with 5 splits for all test and varied the number of clusters for each test

Clusters	Best accuracy	Accuracy after each k fold
7	0.102	0.10,0.09,0.09,0.10,0.09
9	0.1	0.1,0.1,0.98,0.1,0.98
15	0.11	0.098,0.11,0.11,0.10,0.098
20	0.105	0.094,0.099,0.101,0.105,0.95

This network seemed to have a very similar problem to the first one. It was only putting out one number which was based on the original weights as set up randomly. Due to both of these networks having similar issues I suspect that our implementation of the shifting

weights is to blame. We have run out of time for further testing but these results show that while the network is capable of guessing, it is not changing itself in a significant enough way so as to improve itself.