Q5

Pen Data:

Max: 0.9053744997141223
Average: 0.900800457404231
SD: 0.004938324272541829

Car Data:

Max: 0.99
Average: 0.984
SD: 0.00489897948556636

This data is pretty self explanatory. The multiclass classification problem of Pen Data results in a .08 less accurate neural net than binary classification problem of Car Data.

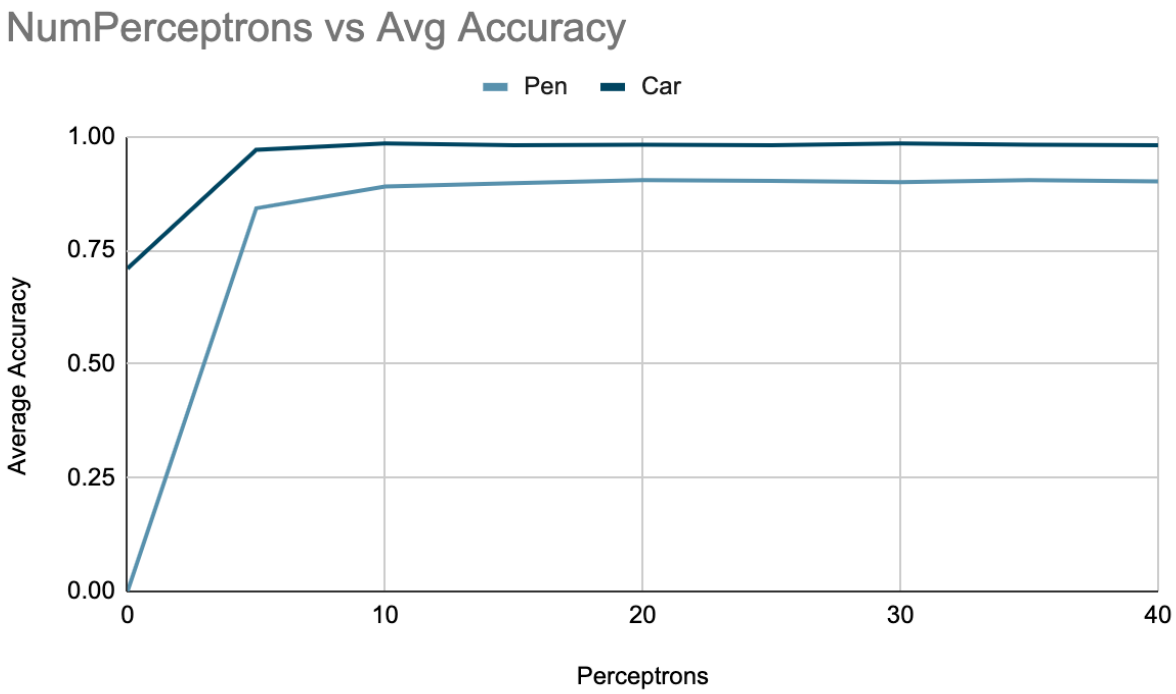
Q6

Pen Full Data:

Perceptron Count	Max	Average	Standard Deviation
0	0	0	0
5	0.8536306460834763,	0.8424242424242424	0.00852548296939468
10	0.9039451114922813	0.8903373356203546	0.007638017249649742
15	0.9036592338479131	0.8973699256718124	0.007862414847771308
20	0.906232132647227	0.9043453401943967	0.001711449919622179
25	0.9053744997141223	0.9025728987993139	0.002157571442437183
30	0.9039451114922813	0.899828473413379	0.006824277217318821
35	0.9079473985134362	0.9045168667810177	0.0021393123995276982
40	0.9065180102915952	0.9014865637507148	0.00673021793919367

Car Full Data:

Perceptron Count	Max	Average	Standard Deviation
0	0.71	0.71	0
5	0.98	0.9710000000000001	0.0058309518948453055
10	0.99	0.9850000000000001	0.006324555320336764
15	0.985	0.9810000000000001	0.0058309518948453055
20	0.99	0.982	0.00509901951359279
25	0.985	0.9809999999999999	0.0020000000000000018
30	0.99	0.9850000000000001	0.003162277660168382
35	0.99	0.982	0.00509901951359279
40	0.99	0.9809999999999999	0.008602325267042634



Through my pen data, I found that for 10 or more perceptrons the average accuracy of my neural net remains around .90. Meanwhile, 0 perceptrons provide an accuracy of 0 as 0 will never be an output for this pen data. On the other hand, 0 perceptrons provides an accuracy of .71 for car data as this is a binary classification and thus, 0 is an answer some of the time. 5 perceptrons for pen data produces a decent accuracy, but still .05 worse than 10+. Car data is a bit different with 5 perceptrons resulting in .971 average, only .01 behind the average of 10+. The general summary of this data is that any more perceptrons than 10 in a layer will not increase our accuracy.

Q7

XOR Data:

{'Perceptron Count: 0': 0.5,
'Perceptron Count: 1': 0.75,
'Perceptron Count: 2': 0.9416666666666664,
'Perceptron Count: 3': 0.9983333333333333,
'Perceptron Count: 4': 1.0}

Above is the averaged data from 30 entire training sessions, each with 5 runs of 10000 iterations. I use a learning rate of 0.6 and weight change threshold of .0000001. Through this data, I found that 4 perceptrons within the hidden layer will provide a 100% consistent neural network for Xor. 3 perceptrons is very close in accuracy, but am not sure if a 99.8% accuracy is "neural net that works well" by this project's standards. The accuracy from 0 perceptrons matches my expectations as there are only 2 outputs (0, 1) so 0 will be correct 50% of the time. Similarly, 1 perceptron accuracy is correct as Xor is non-linearly separable, thus 1 perceptron will not provide high accuracy. However, the accuracy of 2 and 3 perceptrons does not match my expectations. From my understanding, only 2 perceptrons and hence 2 classification lines are needed to reach a consistent neural net. 94% accuracy is arguably consistent, but I imagined 2 perceptrons would be closer to perfect. 3 perceptrons reaches closer to perfect but finally 4 perceptrons gives us a perfectly consistent neural network for Xor.

Q8 - Caesarian Data

1 session:

Max: 0.5
Average: 0.3875
SD: 0.0728868986856

10 sessions:

Max accuracy: 0.6875
Overall average: 0.521726190476

Above is the max, average, and standard deviation of my neural network after 5 runs of 5000 iterations. Following that is the max and averaged accuracy from 10 entire training sessions, each with 5 runs of 5000 iterations. These both utilize 20 perceptrons. As we increase the total sessions, our average accuracy moves closer to 50%. This average makes sense as my data set has a binary output whether the patient receives a caesarian section or not. From this information, I find that my neural network cannot be correctly used on this data set due to lack of data and binary output which result in its brutal 50% accuracy.