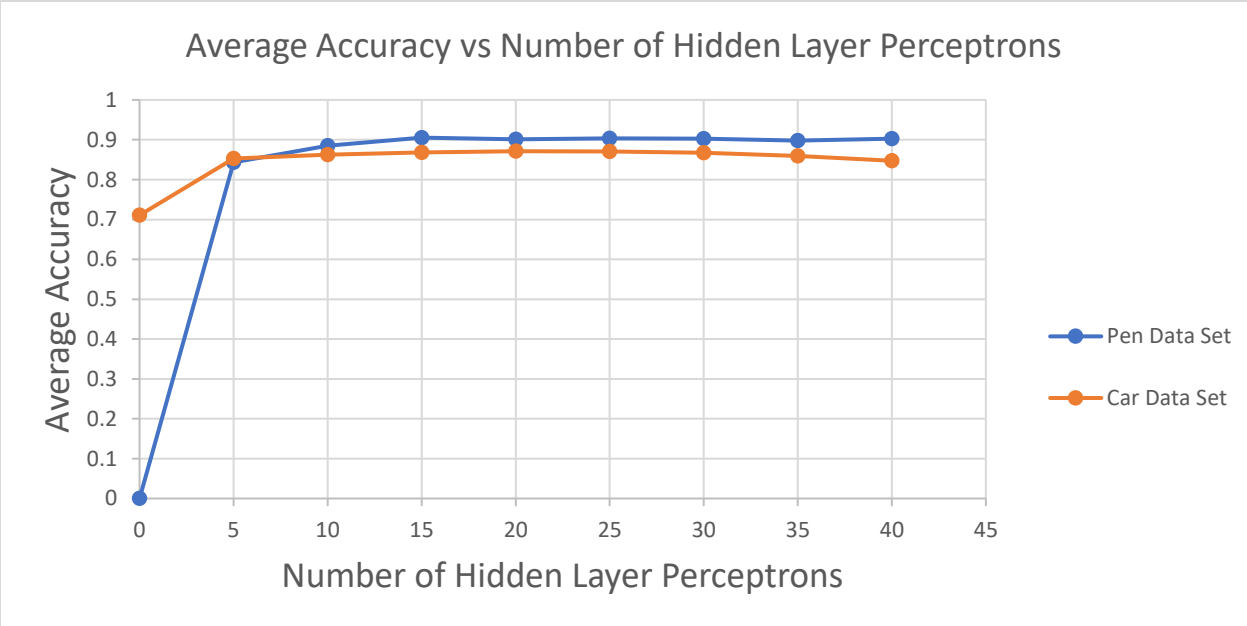
Question 5

Dataset Test	Max	Average	Standard Deviation
Pen	0.90623213	0.90354488	0.00248960
Car	0.86780105	0.86112565	0.00532322

Question 6

	Pen Dataset Test Accuracy Statistics			Car Dataset Test Accuracy Statistics		
Number of Perceptrons	Max	Average	Standard Deviation	Max	Average	Standard Deviation
0	0	0	0	0.710732984	0.710732984	0
5	0.857632933	0.843682104	0.01268395	0.867801047	0.852617801	0.00976338
10	0.891366495	0.885191538	0.004318929	0.877617801	0.862434555	0.013098164
15	0.91023442	0.905317324	0.003613385	0.885471204	0.868586387	0.009754603
20	0.905660377	0.900743282	0.006768482	0.878926702	0.871204188	0.007268819
25	0.909090909	0.903430532	0.003256496	0.886125654	0.870811518	0.008294754
30	0.905660377	0.902401372	0.00382948	0.878926702	0.867146597	0.007012092
35	0.9053745	0.898284734	0.008153078	0.876308901	0.859685864	0.0093254
40	0.906803888	0.902458548	0.007130685	0.880890052	0.847643979	0.016820263



Zero perceptrons in the hidden layer produces poor results for both datasets (mediocre in the car dataset). The jump in average accuracy from zero to five perceptrons in the hidden layer is drastic, as hidden layers allow for non-linearities and more flexibility in the expression. After five, increasing the number of perceptrons barely increases the average accuracy of the neural network. The car data set even shows a slight decrease in average accuracy beyond 20 perceptrons. Increasing the number of hidden layer perceptrons in the neural network, therefore increasing time and compute power needed to train, after 5 or 10 perceptrons shows diminishing returns with regard to the average accuracy of the tests.

Question 7

When there is no hidden layer, the average accuracy is 20% for 5 tests. The average accuracy of the neural net slowly increases as the number of perceptrons increases. Finally, when the neural net has **49 perceptrons** in the hidden layer, the accuracy reaches 100%. The results are what I expected, as increasing the number of perceptrons in the hidden layer allows for more connections. Though I knew a hidden layer would be needed to calculate a non-linear expression, I did not expect to need 49 perceptrons. This is the case for when the learning rate (alpha) is 0.1.

When the learning rate is increased, the number of perceptrons in the hidden layer drastically reduces. When alpha is 1.0, the number of hidden layer perceptrons required drops to only 3. The learning rate can be increased in this case because there is no noise in the dataset, as the XOR function is theoretical data (vs real data with noise). The higher learning rate allows for the weights to be adjusted much more so the output can be found with much fewer neurons.