CS 3600 Homework 4

Budi Ryan — GTID: 903266310

Question 5

Here are the results using default parameter:

For Pen:

(i) Max = 0.906803887936

(ii) Average = 0.904745568897

(iii) STD = 0.00208043515136

For Car:

(i) Max = 0.861910994764

(ii) Average = 0.850392670157

(iii) STD = 0.00692110160545

Question 6

(a) The table:

	Max Pen	Max Car	Average Pen	Average Car	STD Pen	STD Car
Num Hidden Layer						
[0]	0.000000	0.692408	0.000000	0.692408	0.000000	0.000000
[5]	0.857347	0.844241	0.840023	0.826440	0.009853	0.011877
[10]	0.897370	0.826571	0.890509	0.816885	0.004971	0.005726
[15]	0.901944	0.835733	0.892739	0.820681	0.006500	0.013444
[20]	0.908805	0.830497	0.905889	0.821073	0.002180	0.005294
[25]	0.909091	0.825262	0.903030	0.815445	0.003764	0.006725
[30]	0.905946	0.823298	0.902630	0.811649	0.002298	0.010384
[35]	0.909377	0.827225	0.903145	0.818586	0.007149	0.005258
[40]	0.905374	0.818717	0.901544	0.809817	0.003462	0.006983

Figure 1: Table of Statistics

(b) Learning Curve:

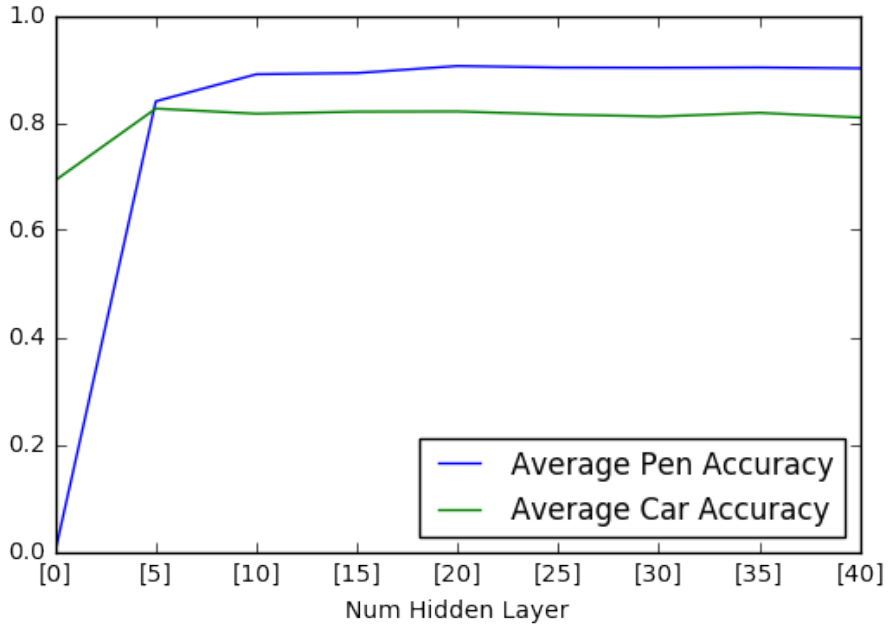

Figure 2: Table of Statistics

(c) Analysis:

As we can see from the graph above, by increasing the number of hidden layer, the average accuracy for both datasets increase. This is because as we grow the number of perceptrons in the hidden layer, the size of the weight matrix increases.

The average accuracy is also affected by the size of training data. We can observe from the graph that Pen's accuracy is higher than Car's because Pen has more training data.

Question 7

Please see the script I wrote (xor_test.py) for this particular question.
In total, I have a dataset of 500 rows in which I split into training / testing set with a ratio of 70:30.

From the experiment I did by playing with the number of the perceptron:
0 perceptron always gives $\sim 50\%$ testing accuracy.
By increasing the number of perceptrons, the accuracy gradually improves.
1 perceptron gives $\sim 75\%$ testing accuracy.
2 perceptrons is not consistent because of the randomization I applied, most of the time it gives 100% but sometimes it may drop to $\sim 72\%$.
More than or equal to 3 perceptrons consistently gives 100% accuracy.

The result of the experiment is what I have expected. XOR function is not linearly seperable. Therefore, using 0 or 1 perceptron is not enough to correctly classify the dataset.