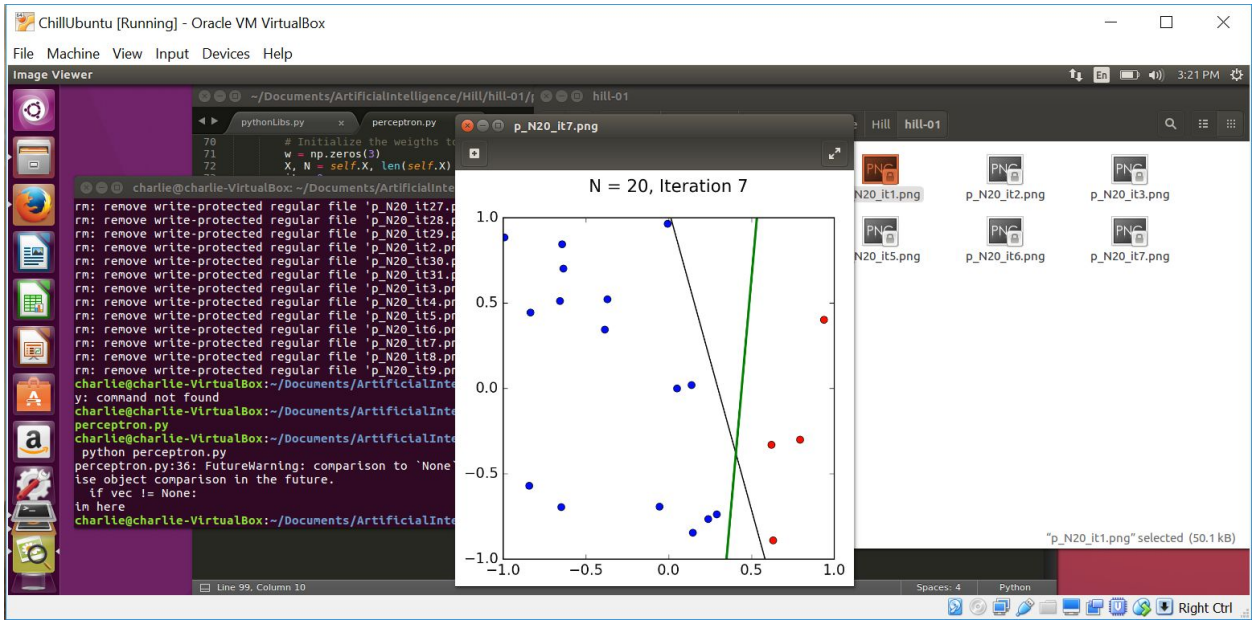
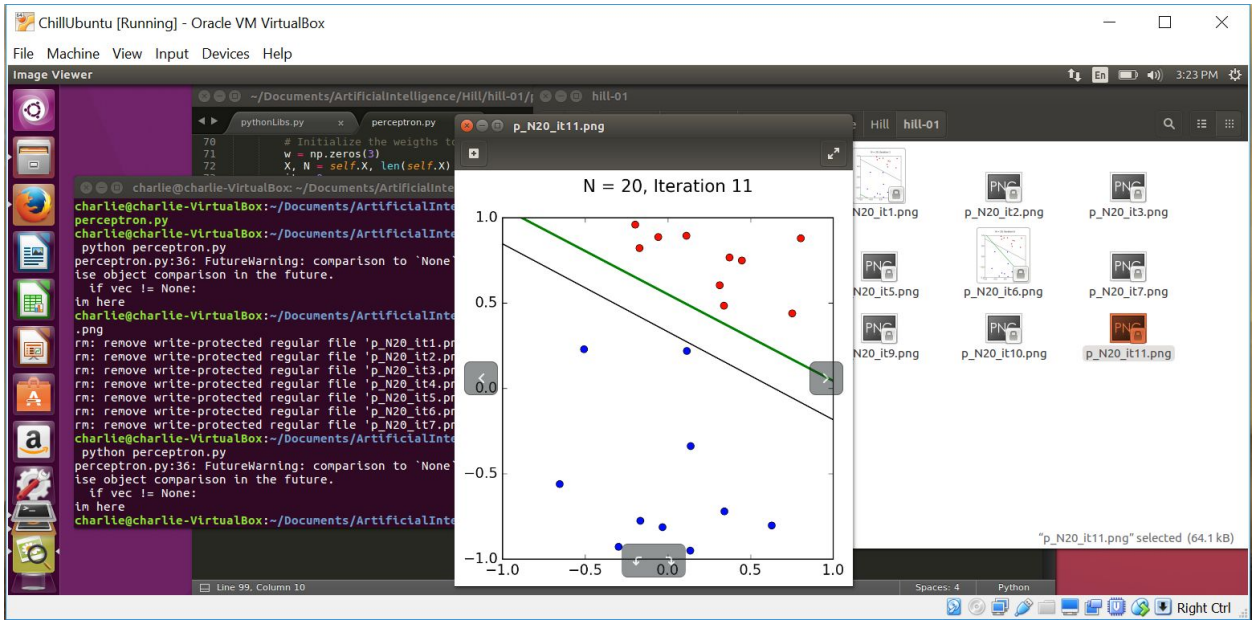
Charlie Hill
Artificial Intelligence
Pablo Rivas
19 September 2016

Homework 01

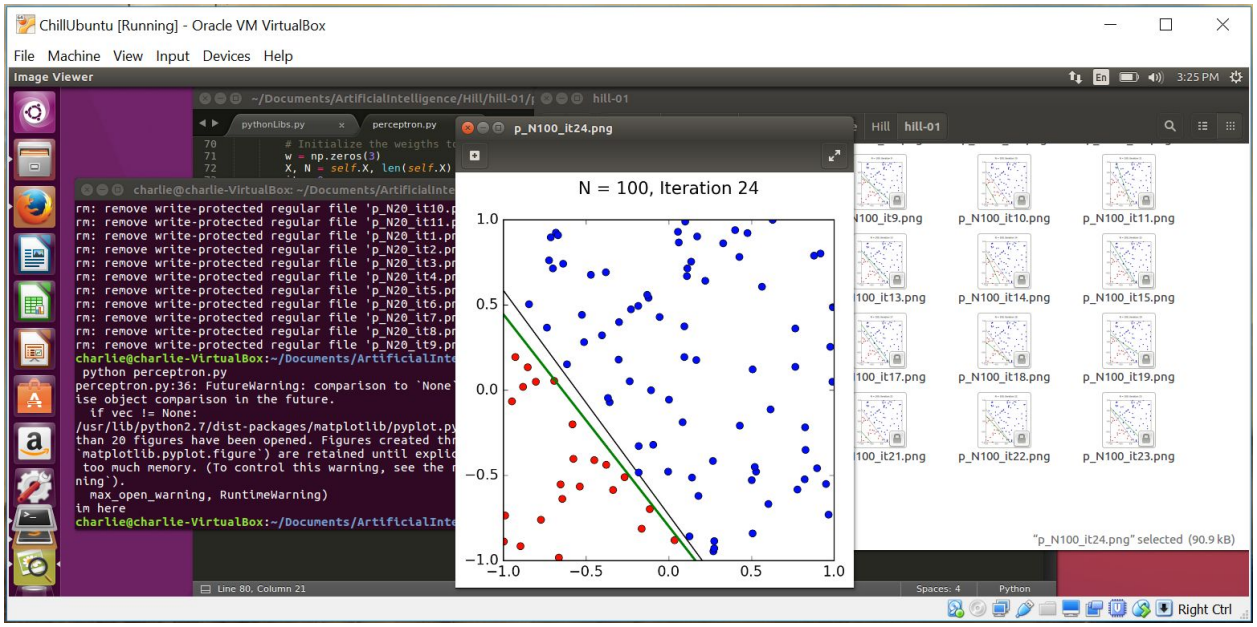
a and b) Running the perceptron with 20 data points gave me this. It took 7 iterations for the perceptron to find a solution. The perceptron does not seem close to the expected solution that is pre generated.



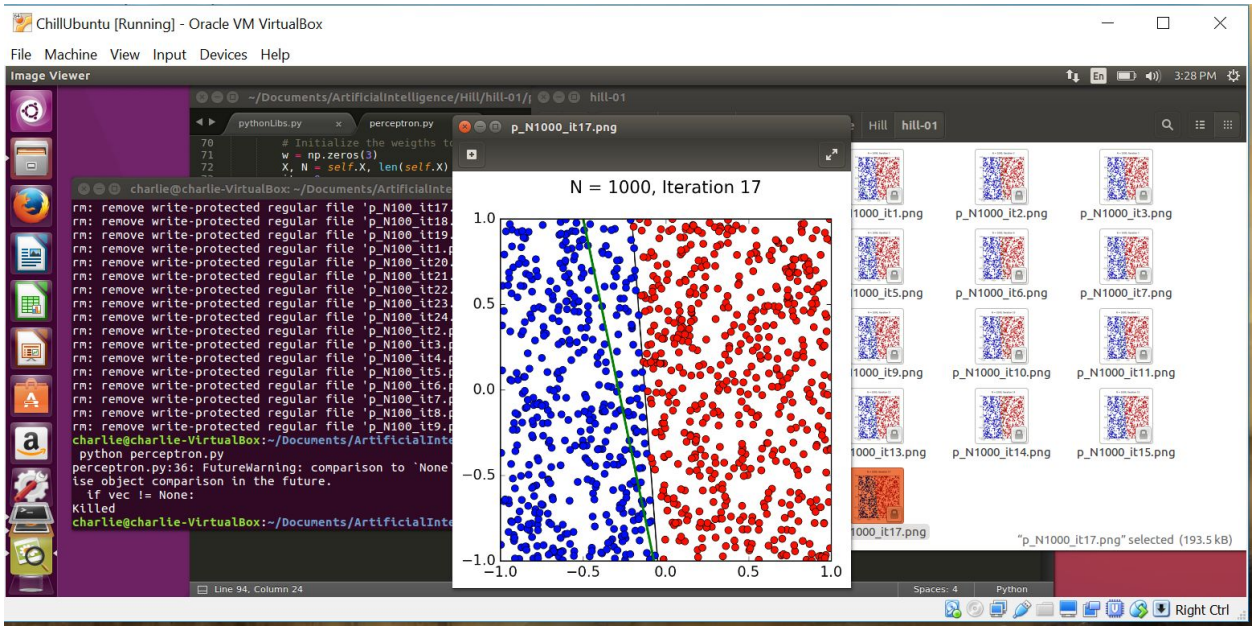
c) I ran the script again and got this.This time it took 11 iterations to find a solution. The perceptron seems much closer to the expected answer this time.



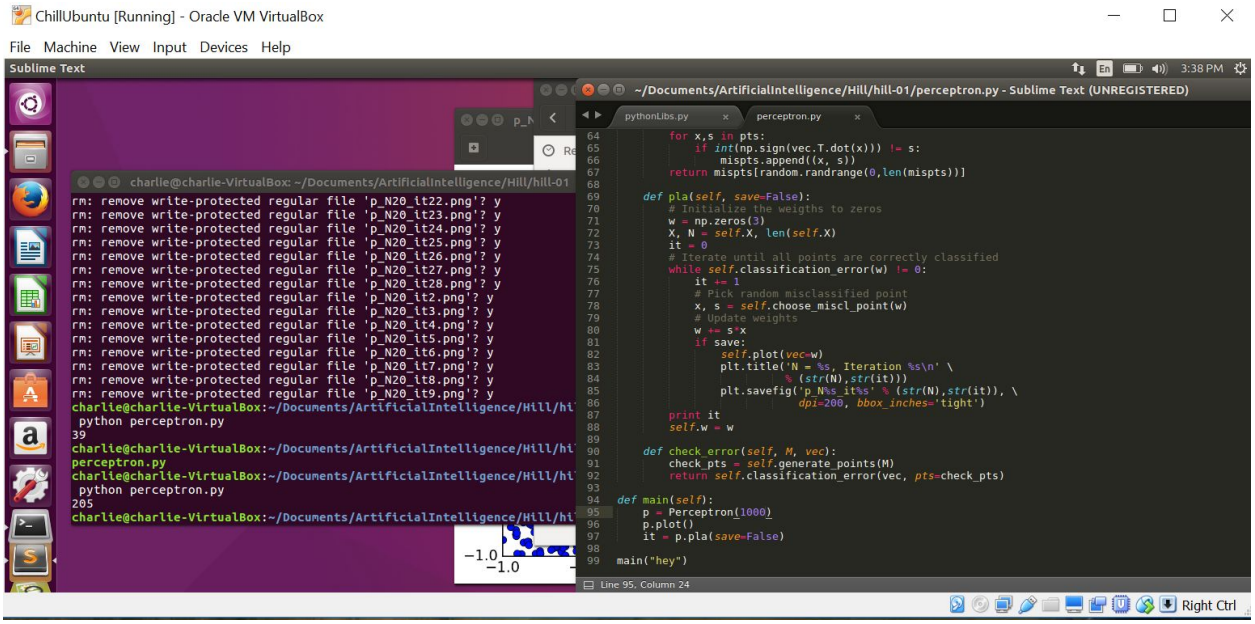
d) I changed the data points to 100 and ran the script again. This time it took 24 iterations and the perceptron is very close to the expected solution.



e) I changed the data points to 1000 and ran the script. The first time I ran it, the script was killed after a bit, most likely because it was using too much memory.

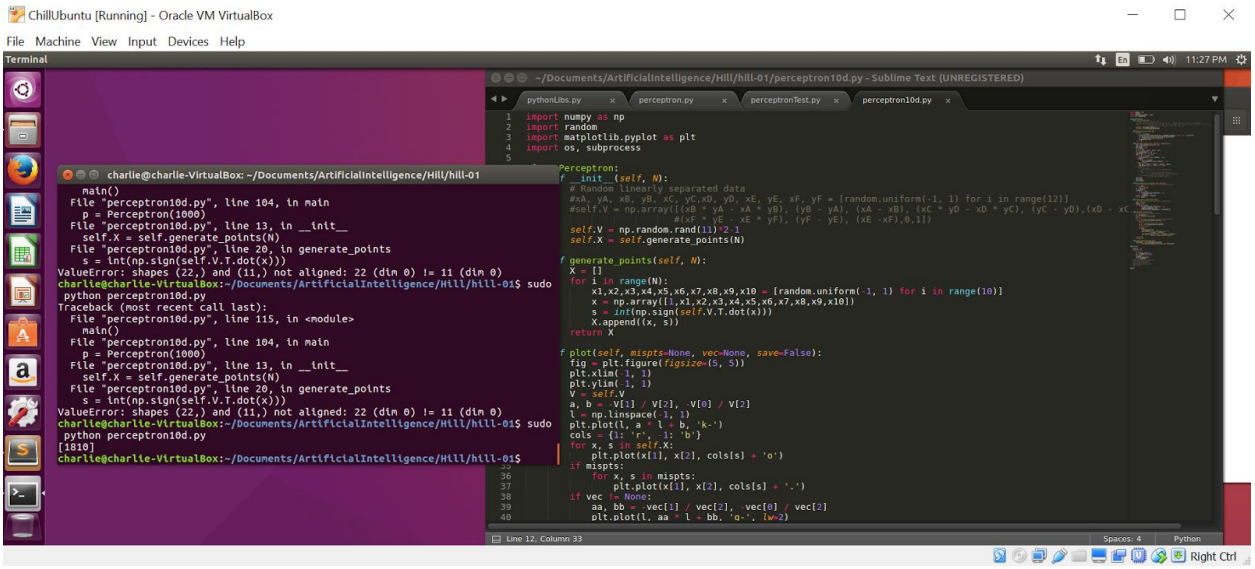


So then I decided to not save the plots so I could find out how many iterations it would actually take. It would end up taking 205 iterations to find a solution. If I had to guess I can imagine the perceptron probably looks really similar to the expected solution because with so many data points the line has to be exact because there is only a little room for solutions. The more we increase the data points the closer the perceptron will get to the expected solution.

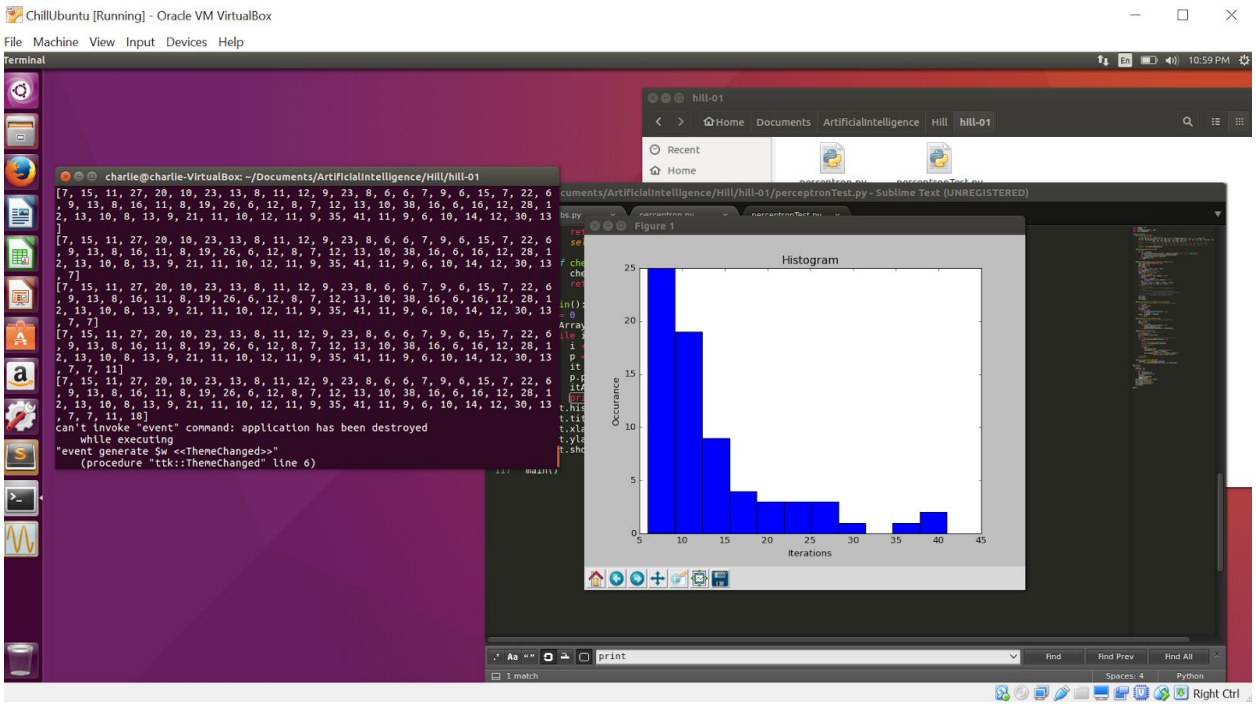


f)This was by far the hardest part of the homework to figure out in my opinion. Altering the program to allow for 10 dimensions was not easy. However, I was eventually able to figure it out and get it working. From there running the program became much slower. I imagine the perceptron has much more thinking to do in 10 dimensions.

I ran the program with 1000 data points. It took 1810 iterations to find the perceptron in this screenshot. I ran it a few times and found around 2000 to 4000 iterations seemed to be pretty common.



g) I ran the algorithm 100 times with 20 data points. By the time it got to the end the process was being killed so I had to lower the number of times it was being run. It was able to be run 70 times pretty consistently. I built the histogram and here is a screenshot of it.



It seems that the perceptron finds the solution under 15 tries generally when there are only 20 data points.

h) Both *N* and *d* seem to both have positive correlation with respect to time. As the number of data points increases, the amount of iterations the perceptron takes to find a solution increases, as seen by the 1000 data points experiments. As the number of dimensions increase the time it takes for the perceptron to calculate its next move seems to increase. This was noticeable when the 10 dimensions were used to try and calculate the 1000 datapoint perceptron. When there were only two dimensions it seemed to find the solution much faster.