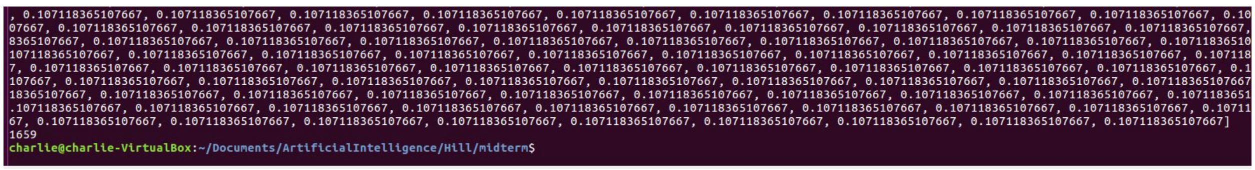
Charles Hill
Artificial Intelligence
Professor Rivas
12 October 2016

Midterm

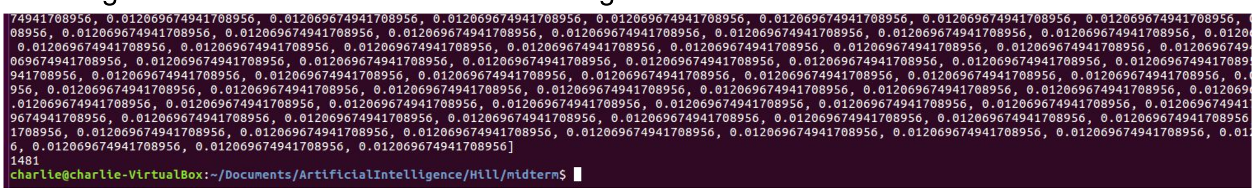
1.a.) Running the pocket perceptron on this dataset took an incredible amount of time. This is because there were 7291 data points. The data was not linearly separable, therefore the perceptron would run forever and not find a solution.The best error after 1000 of the weights are the same consecutively is 0.107118365. Changing the dataset does not cause a noticeable difference other than the data points being different colors. The best error from the second set after 1000 of the weights are the same consecutively is 0.012069675.

These are the weights from the first set where the weight is zero. The pocket has to have the same weight for 1000 times before ending. (All these .pngs are on my Github as well).



```
, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.10
07667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667,
8365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.10711836510
107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118
7, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.1
107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667
18365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.1071183651
.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.10711
67, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667, 0.107118365107667]
1659
charlie@charlie-VirtualBox:~/Documents/ArtificialIntelligence/Hill/midterm$
```

The weights from 1000 consecutive saved weights. This is from the second set.

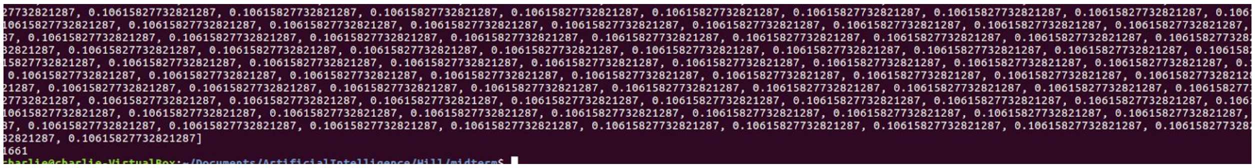


```
74941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0
08956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.0120
 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.0120696749
069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.0120696749417089
941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.
956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069
.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.01206967494
9674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956
1708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956, 0.01
6, 0.012069674941708956, 0.012069674941708956, 0.012069674941708956]
1481
charlie@charlie-VirtualBox:~/Documents/ArtificialIntelligence/Hill/midterm$
```

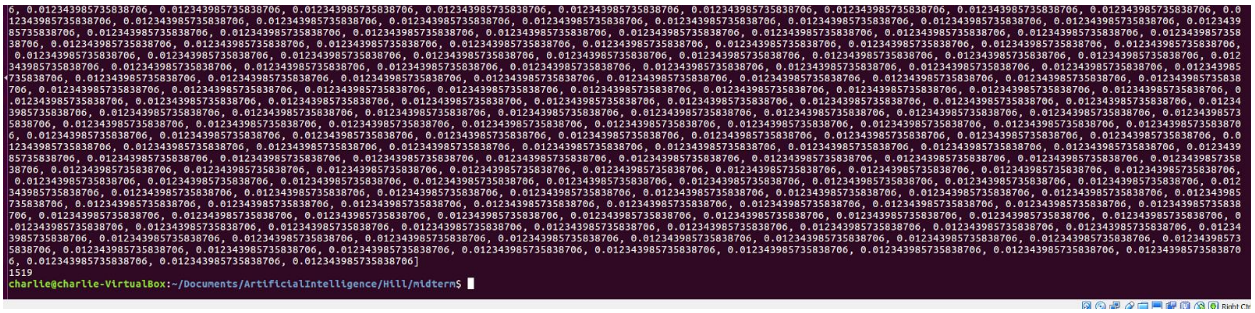
b.) The linear regression worked well on my VM. The line was created with the first iteration and goes right through the datapoints. Linear regression is strong here because it does not care if the data is not linearly separable. Changing the points sign altered the linear regression line. The line is based mathematically off the points so therefore it still finds the best possible solution despite the data not being linearly separable.

c.) Having the pocket algorithm start from the linear regression made a little bit difference from the weight starting at zero. The error starts off smaller but doesn't seem to get to much smaller from that point. This is because the data is not linearly separable and therefore the pocket algorithm will iterate indefinitely until it is told to break. The best error after 1000 of the weights are the same consecutively is 0.106158277. Changing the dataset does not cause a noticeable difference other than the data points being different colors. The best error from the second set after 1000 of the weights are the same consecutively is 0.0123439857.

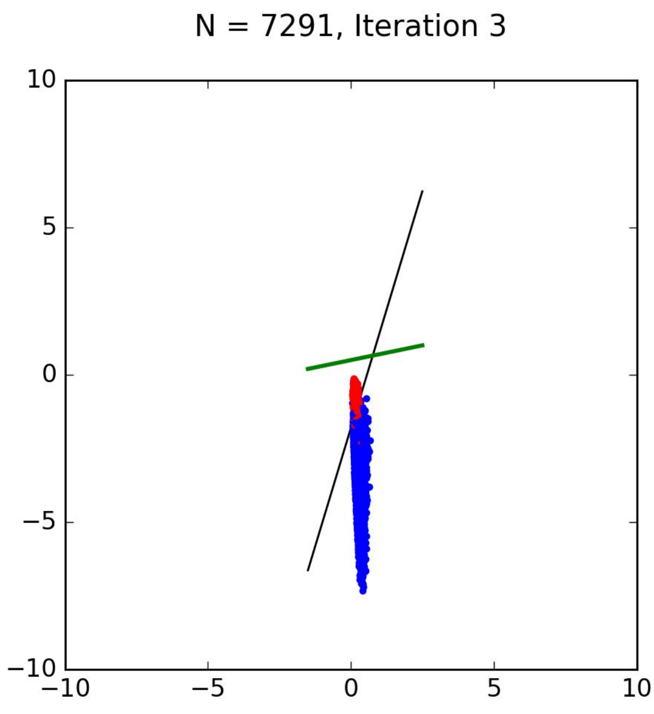
These are the weights from the first set where the weight starts from the linear regression. The pocket has to have the same weight for 1000 times before ending.



These are the weights from the second set where the weight starts from the linear regression. The pocket has to have the same weight for 1000 times before ending.



This is the pocket algorithm running with the altered dataset and the weight starting from zero. It made it to the third iteration before crashing. Plotting the 7291 data points over and over again crashes my VM rather quickly.



2.) After 10 iterations the answer to Problem 2.12 is about 452957 samples that are needed. This was found using my midtermpt2.py file. This program runs the equation 10 times. Each time it takes the newly found N and solves the equation with it. After about 10 iterations the difference in iterations is not significant.

```
charlie@charlie-VirtualBox:~/Documents/ArtificialIntelligence/Hill/midterm$ sud
o python midtermpt2.py
257251.363936
434853.08159
451651.627315
452864.520629
452950.340234
452956.403785
452956.832159
452956.862423
452956.864561
452956.864712
Answer after 10 iterations: 452956.864712
charlie@charlie-VirtualBox:~/Documents/ArtificialIntelligence/Hill/midterm$
```