

Charlie Hill
Artificial Intelligence
Pablo Rivas
11 October 2017

Homework 3

1 a) Running the pocket algorithm starting from $w = 0$ takes the most computation to solve. The perceptron has no idea where to begin so it just starts working. Using the two blob data set it either found a solution in some iterations or it could never find a solution. Often times the solution that was found was not “good”, as in it was very close to one blob or the other. I recorded the weights in an array and printed them after. The best weight was chosen if the solution was not found in a given number of iterations.

These are the weights from when the weight started at 0. (Sorry for the size, the .png is on my Github.)

```
charlie@charlie-VirtualBox:~/Documents/ArtificialIntelligence/Hill/hill-03$ sudo python pocketPLA2.py
[0.5, 0.3, 0.12, 0.07, 0.03, 0.03, 0.03, 0.02, 0.04, 0.03, 0.02, 0.04, 0.04, 0.03, 0.01, 0.03, 0.02, 0.04, 0.04, 0.04, 0.01, 0.01, 0.03, 0.04, 0.02, 0.04, 0.03, 0.0]
[1.0, 0.5, 0.3, 0.12, 0.07, 0.03, 0.03, 0.03, 0.02, 0.02, 0.02, 0.02, 0.02, 0.02, 0.02, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01]
28
charlie@charlie-VirtualBox:~/Documents/ArtificialIntelligence/Hill/hill-03$
```

b) The linear regression process is the fastest. It find the solution immediately and seems to use little to no computation. Linear regression also doesn't care if it misclassifies a few points if it can find the best solution. This means it does not get stuck if there is no possible linear separation between the blobs.

c) The Pocket algorithm tends to work well if it starts from the linear regression line. If the data is linearly separable the solution is found within one iteration. However, if the blobs are not linearly separable, the perceptron just continues until it the maximum number of iterations is reached. The saved weights tend to be better when starting from the linear regression line.

These are the weights from when the weight starts from the linear regression line. (Sorry for the size, the .png is on my Github.)

```
charlie@charlie-VirtualBox:~/Documents/ArtificialIntelligence/Hill/hill-03$ sudo python pocketPLA2.py
[0.5, 0.43, 0.16, 0.47, 0.23, 0.43, 0.33, 0.1, 0.25, 0.18, 0.06, 0.12, 0.12, 0.04, 0.21, 0.12, 0.03, 0.26, 0.12, 0.03, 0.2, 0.12, 0.05, 0.05, 0.12, 0.01, 0.25, 0.16, 0.01, 0.06, 0.11]
[0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01]
31
charlie@charlie-VirtualBox:~/Documents/ArtificialIntelligence/Hill/hill-03$
```

Adding outliers to the set tends to skew the linear regression line. It does not seem to affect the perceptron as much because they perceptron only checks whether the point was misclassified or not.