CS440 : Introduction to Artificial Intelligence **Spring 2014**

Problem Set 2 - Part 2 Problem 3

Dan McQuillan *Handed In: February 18, 2014*

1. **Solution to problem 3**

   (a) Find and report the list MAX for Dataset 2
   $(70.0, 1.0, 4.0, 170.0, 407.0, 1.0, 2.0, 194.0, 1.0, 4.0, 3.0, 3.0, 7.0)$

   (b) Repeat Problem 1

Training Dataset 1x

The training took 14 epochs.
$Threshhold : 5.0$
$\gamma \rightarrow 0.0320963696907362$
$w_1 \rightarrow -1.2368421052631573$
$w_2 \rightarrow 3.0$
$w_3 \rightarrow 5.5$
$w_4 \rightarrow -0.2899999999999997$
$w_5 \rightarrow -0.13829787234042568$
$w_6 \rightarrow 0.0$
$w_7 \rightarrow 0.5$
$w_8 \rightarrow -13.317204301075275$
$w_9 \rightarrow 3.0$
$w_{10} \rightarrow 5.645161290322578$
$w_{11} \rightarrow 2.9999999999999987$
$w_{12} \rightarrow 5.0$
$w_{13} \rightarrow 5.7142857142857135$

Test Dataset 1x

Confusion Matrix:

54	0
0	63

Total loss: 0.0

Test Dataset 2x

False Positives:
Index: 7
Confusion matrix:

13	0
1	19

Total loss: 0.010623925503195387

Application Dataset 3x

1 → 0.0
2 → 1.0
3 → 0.0
4 → 1.0
5 → 1.0
6 → 0.0
7 → 0.0
8 → 0.0
9 → 0.0
10 → 1.0
11 → 1.0
12 → 0.0
13 → 1.0
14 → 0.0
15 → 0.0
16 → 0.0
17 → 1.0
18 → 0.0
19 → 0.0
20 → 0.0
21 → 0.0

(c) Explanation

The results above show that the number of epoch were quite a bit less. This is a result of the perceptron having to oscillate because of the larger difference between input vectors. Instead it is more likely to increase with a smaller Δw and have a lesser chance to have to use a negative Δw to correct a movement when it moves past the margin of a "good" perceptron.

(d) Test new perceptron on DataSet1

When testing data set 1 it resulted in a complete failure of classifying the input vectors from dataset one. This is a result of normalizing the input vectors before training which caused the weight vector to be much smaller. Although we normalized the input vectors that does not correlate to the weight vector itself being normalized.

(e) **Is is possible to recover the problem 1 perceptron**

It is not possible to recover the perceptron from problem 1 from this weight vector since the resultant weight vectors are dependent on the values chosen to be the initial training weights. It will not also be able to be recovered since the test on DataSet1 failed which means that the resultant vector from part d is not the weight vector normalized but instead it has a different direction entirely. Since it lacks this corollary it is not possible to recover the perceptron from problem 1. In addition, even if we used the resultant weight vector as the initialization of the perceptron, then trained the weight vector we would still be left with a different perceptron.