CS 5350/6350: Machine Learining Fall 2017

Homework 2

Handed out: 12 September, 2017
Due date: 26 September, 2017

1 Warm up: Linear Classifiers and Boolean Functions

Solution:

1. $x_1 - x_2 + x_3 - 1 \geq 0$. Thus, w = [1, -1, 1] and b = -1.
2. $x_1 + 2x_2 + x_3 - 2 \geq 0$. Thus, w = [1, 2, 1] and b = -2.
3. $x_1 - 2x_2 + 3x_3 - 1 \geq 0$. Thus, w = [1, -2, 3] and b = -1.
4. This boolean function is not linearly separable.
5. $-x_1 + x_2 - x_3 - 3 \geq 0$. Thus, w = [-1, 1, -1] and b = -3.

2 Mistake Bound Model of Learning

Solution:

1.

   (a) Since both x_1 and x_2 is integer, $|\mathcal{C}|$ is the number of integers in the range of l which 80.

   (b) $f_l(x_1^t, x_2^t) \neq y^t$

   (c) If $f_l(x_1^t, x_2^t) = +1 \quad and \quad f_l(x_1^t, x_2^t) \neq y^t$, then remove all l that are bigger than the biggest absolute value of $x_1^t \quad and \quad x_2^t$
   If $f_l(x_1^t, x_2^t) = -1 \quad and \quad f_l(x_1^t, x_2^t) \neq y^t$, then remove all l that are smaller than the biggest absolute value of $x_1^t \quad and \quad x_2^t$

   (d) It will make at most 80 - 1 = 79 mistakes.
   Pscudocode:
   Initialize C = the set of all l
   When an example x arrives:
      Randomly choose a l to predict the result

if $f_l(x_1^t, x_2^t) \neq y^t$:

Remove that l from C

Break if there is only one l left in C.

2. The algorithm will stop when there are M + (M - 1) elements left since the majority of the function is always right and no more elements will be removed. Thus, the mistake bound is $O(log(N) - log(2M - 1)) = O(log(\frac{N}{2M-1}) = O(log(\frac{N}{M})$.

3 The Perceptron Algorithm and its Variants

Solution:

1. I used MATLAB to implement this experiment. I have a function to build up formalized table with the data file by putting zeros in omitted features and filling a matrix with cells. Each type of perceptrons will have its own function to train data. A error reporting function is used to report errors with a trained perceptron and test data set. Since there is 67 features, I create weight vector w with [w = 0.02 * rand(1,67) - 0.01] to initialize its value. And when I need to times it with x, I use [w * x'] to flip the dimensions of x.

2. Majority baseline accuracy on test set is 57.3082%.
   Majority baseline accuracy on development set is 54.9204%.

3.

   Simple Perceptron:

   (a) $\eta = 1$

   (b) 90.1942%

   (c) 15623 updates in 20 epochs

   (d) 90.521%

   (e) 92.1852%

   (f)

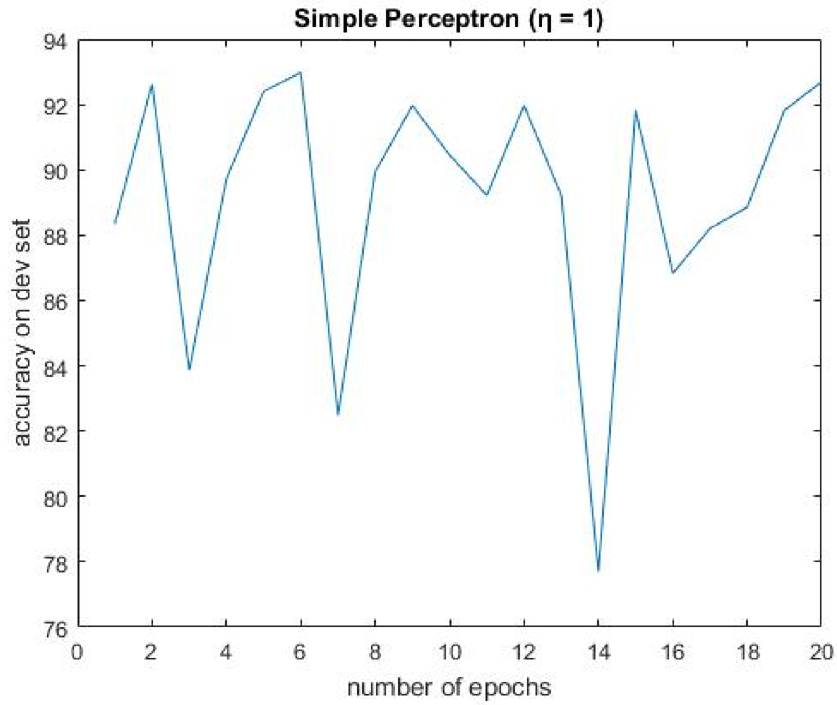

Figure 1: Simple Perceptron with $\eta = 1$

Perceptron with dynamic learning rate: Update learning rate on each example

(a) $\eta = 1$

(b) 84.8269%

(c) 21408 updates in 20 epochs

(d) 85.6006%

(e) 86.4689%

(f)

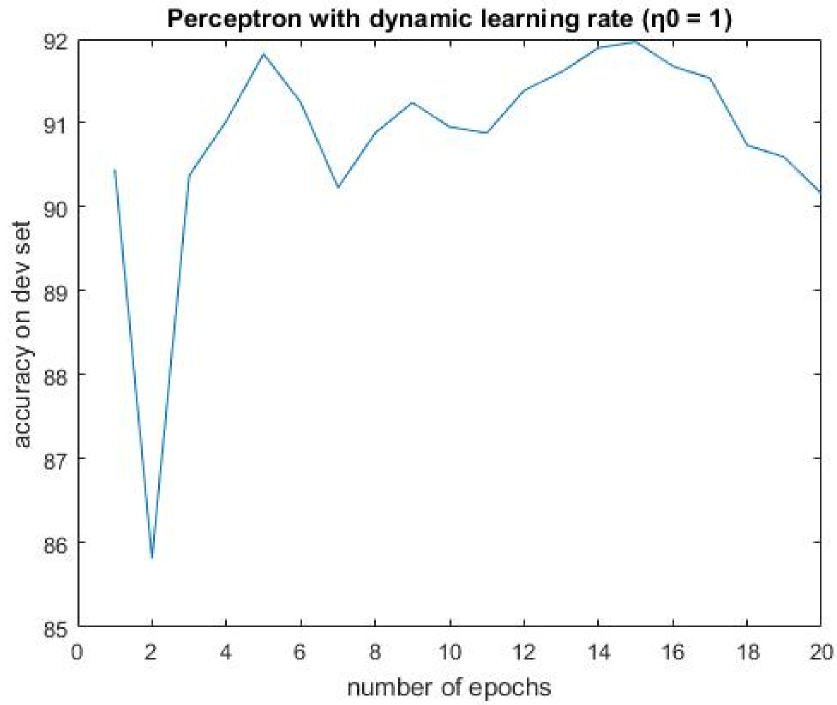

Figure 2: Dynamic learning rate perceptron with $\eta_0 = 1$

Margin Perceptron: Update learning rate on each example

(a) $\eta = 1$ and margin $= 2$

(b) 87.6854%

(c) 73624 updates in 20 epochs

(d) 89.3632%

(e) 90.2315%

(f)

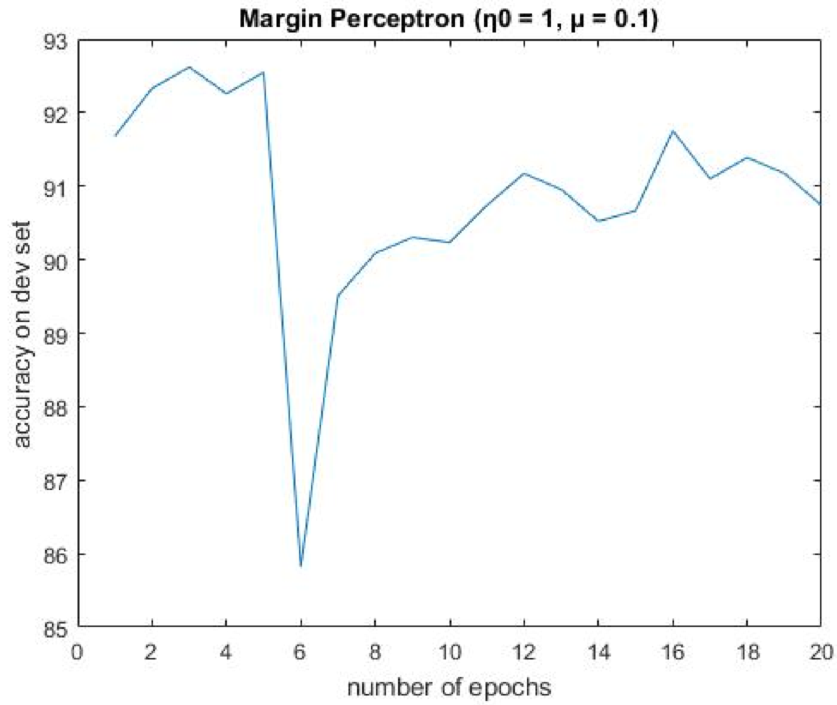

Figure 3: Margin Perceptron with $\eta_0 = 1$ and $\mu = 0.1$

Averaged Perceptron

(a) $\eta = 1$

(b) 91.8466%

(c) 15610 updates in 20 epochs

(d) 92.547%

(e) 92.9812%

(f)

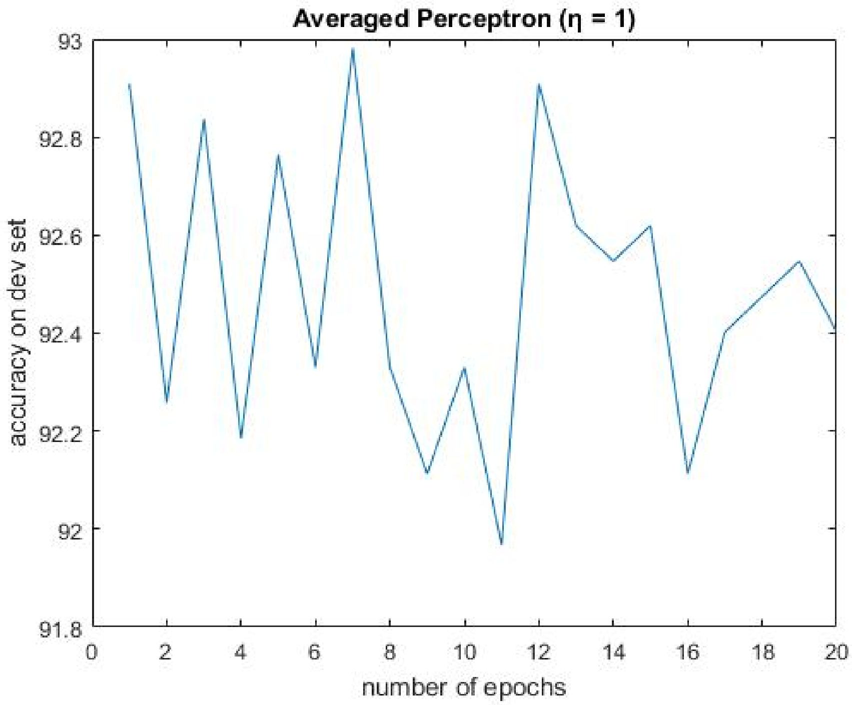

Figure 4: Averaged Perceptron with $\eta = 1$