

機器學習 REPORT

Logistic regression:

我 logistic regression 用三個方法:

第一個：把所有的 train data 下去 train，建立一個 w 和 b 的 model，在 while 迴圈裡放 iteration，所有的 train data(4001 筆) train 完才更新一次我的 w 跟 b 。

第二個：拿所有的 train data 下去 train，建立一個 w 和 b 的 model，在 while 迴圈裡放 iteration，一筆 train data train 完就更新 w 和 b 。

第三個：把所有的 train data 先分三筆，分別建立三個 w 和 b 的 model，在 train 完之後，拿這三個 w 跟 b 去 test 另外兩個的 data，最後選 error 最小的 w 和 b model。

PLA (Perceptrons learning algorithm) :

將答案分成 1 跟 -1 兩類，找出一個 model 的 w 和 b 可以使 y (答案) 跟 $z(z = w*x+b)$ 相乘後同號。所以當 $y*z$ 異號時，我更新 w 讓他成為 $w = w + y[i]*x[i]$ ，並去算這組 w 跟 b 的 error，如果比舊的 error 低，就用這組 w ，反之就把它丟掉。但我在做的時候發現 PLA 的 error 一直降不下去，於是就讓他每一次判斷 $y*z$ 開始的地方不一樣。讓他隨機出發，避免 local minimum 的問題。

在 train 的過程中，處理 data 的地方比起 HW1 簡單，但是如果有效的分類卻不容易。這次沒有調整 learning rate 而是在資料的處理上做比較多的版本。發現每個版本的 learning rate 需要的不一樣，而且在跑的時候也會比一樣快。另外發現 $(w*x[i]).sum()$ 的速度比起 $sum(w*x[i])$ 還要快很多。