106062212 郭蕙綺

ML HW3 Report

HW3.1：

首先利用open()、readlines()來整理資料，在這題是使用有w0的演算法，所以除了將[w1~w5] append到trainingSet這個list外，也多append一個1(w0)到trainingSet。利用for-loop確認每筆資料的label並存到C陣列。在這題我define了兩個function，分別為CH、Weight。

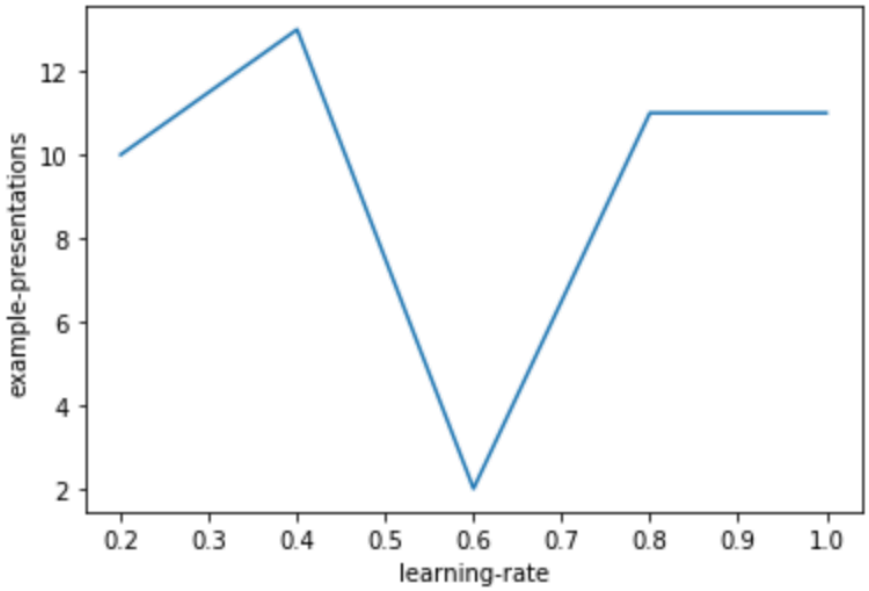
CH：這個function是在做公式裡c(x)-h(x)的部分，傳入pos(第幾個example)及data(example的資料)，利用for-loop確認h(x)的值（做四捨五入確保正確性）後return h-C[pos]的值。

Weight：這個function在做weight更改的動作，依據講義的公式為原本的weight加上learning rate*(c(x)-h(x))*example的attribute值，我在這個地方四捨五入到小數點下第一位，因為python常會有一些小數點上的問題會造成這題的誤差，所以做了四捨五入。

接下來就是實作的部分，以learning rate=0.2來說明，剩下的做法均與0.2的作法相同。

先用一個for-loop將weight的初始值設為0.2，將done、count、rund初始化為0，done用來判斷是否做完，count用來算已經連續對了幾個（c(x)=h(x)），對20個表示完成，rund用來算epoch數。用while-loop done!=1來做，每次都將rund+1，在這個loop裡面跑一個for-loop（做20個examples），call CH function 回傳值為0則count+1，回傳值不為0則call Weight function來更改weight的值並將count歸零，最後判斷當count=20表示完成，將done設為1。

用同樣的方法將0.4~1.0都做完後，利用python的plt來畫出圖表：



看此圖表可以發現在learning rate = 0.6時epoch數較少，表示太過極端的learning rate會造成浪費（epoch數較多）。

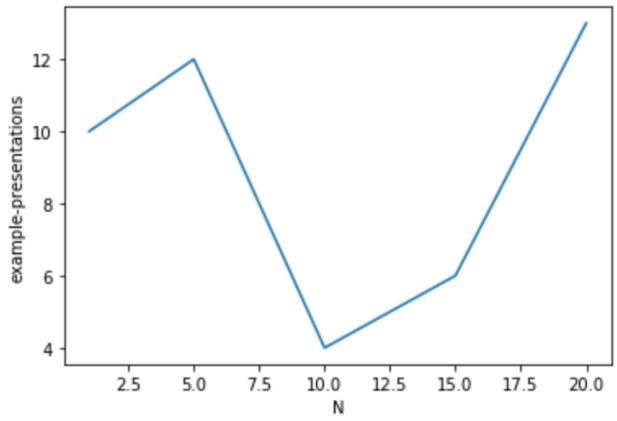
106062212 郭蕙綺

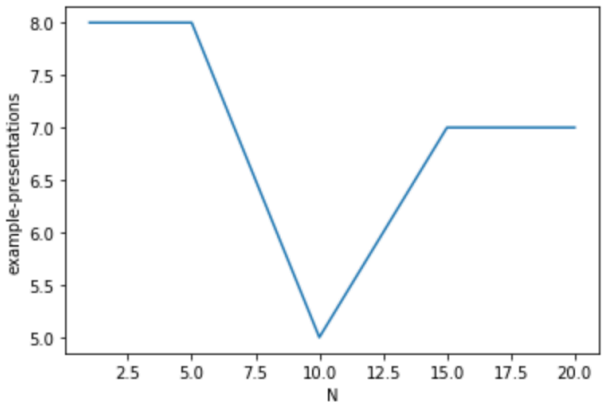
HW3.2：
讀資料及整理資料方法與HW3.1相同，也define了與HW3.1相似的兩個function (CH、Weight)微調了function裡的for-loop次數（加上N）。

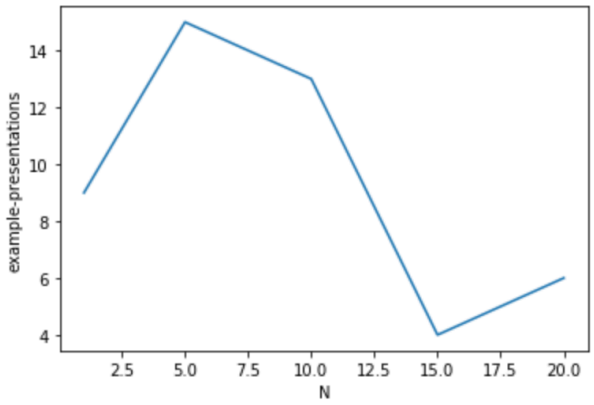
題目要求實作N=1,5,10,15,20，這裡以N=1來做說明，剩下四個做法與N=1相同。

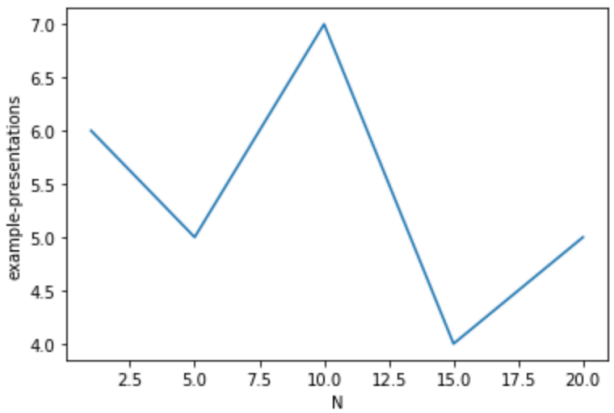
設一個tSet將[w0~w5, randint(0,1)]（N=5則多5個randint(0,1)）append到tSet，weight=[0.2]*7（因為一筆資料有7個attributes），接下來while-loop演算法與HW3.1相似，不同的地方在於呼叫CH、Weight function時要傳入N。

最後同樣使用python的plt來畫出圖表：









以上是跑了四次的圖表，由於attribute是random產生所以狀況不太一定，像是上面兩張圖N=10時epoch數較少，下面兩張則相反。