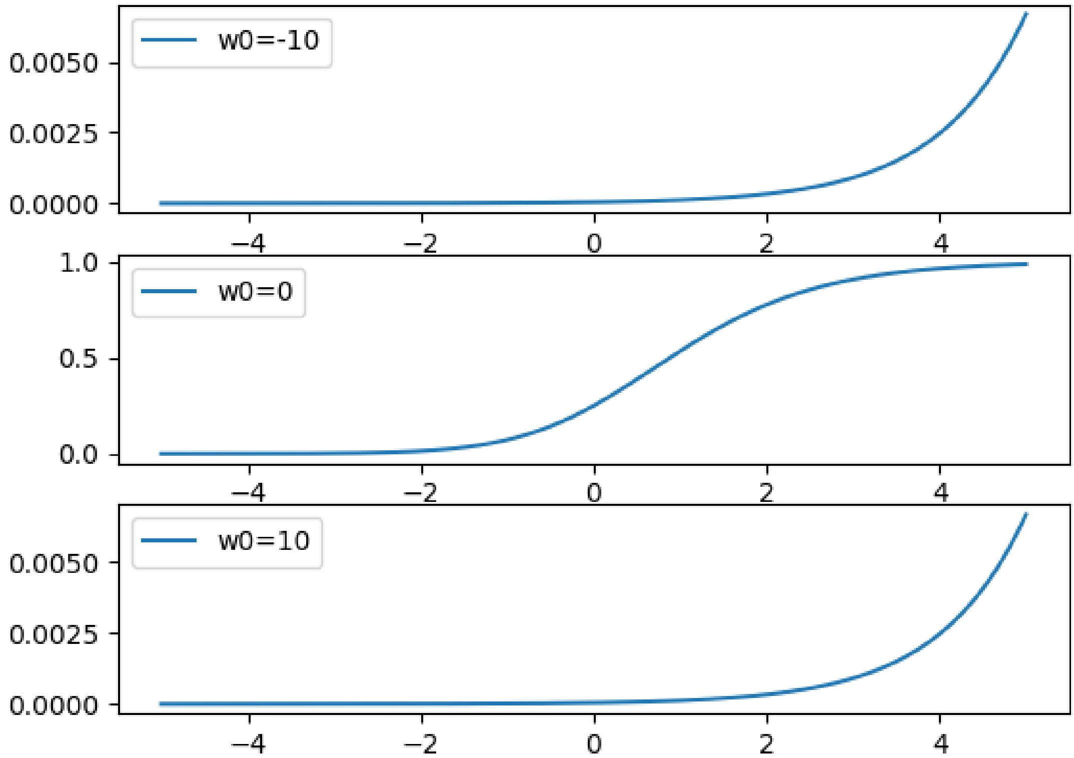
機器學習 HW2 part2
工科系 111 E94071209 林政旭

Question 3.0:

下圖橫軸為 w1，設定 w0 為定值的 likelihood 圖，由下圖可知不管 w0 為何，當 w1 越大時，likelihood function 的值皆會越大，並越來越趨近於 1，因此，不存在 unique 的一組 w0 及 w1 使得 likelihood value 有最大值。

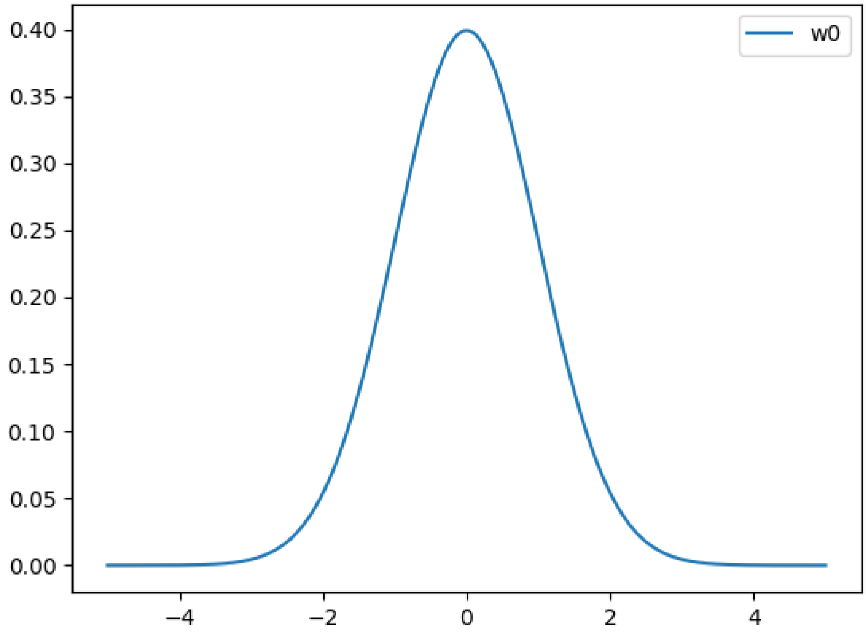


下圖為 sigmoid、likelihood、Normal weight prior 之 P[w] 與 Laplace weight prior 之 P[w] 的程式圖，並設定 x 為在-5 至 5 之間的 2000 個分布點

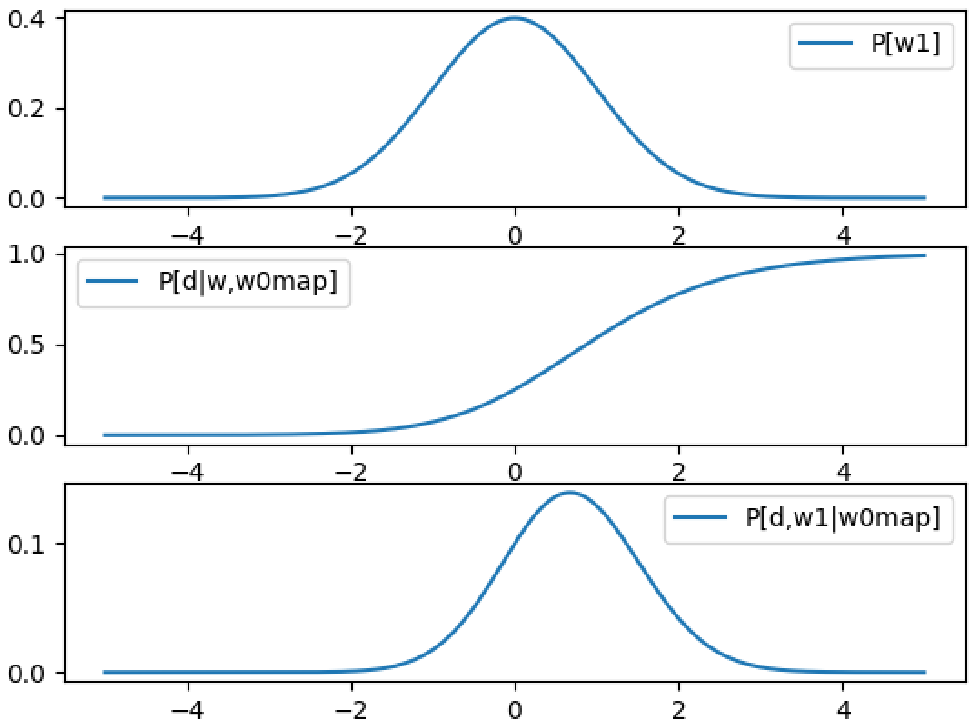
```
def sigmoid(x):
    return 1/(1+np.exp((-x)))
def likelihood(w0,w1):
    return sigmoid(w1+w0)*sigmoid(w1-w0)
def normalp(w):
    return np.exp(-np.power(w,2)/2)/np.power(2*np.pi,0.5)
def laplacep(w):
    return np.exp(-np.abs(w))/2

x = np.linspace(-5,5,2000)
```

Question 3.1

1. 下圖橫軸為 w0，縱軸為 P[w0]之圖，由圖可知最大值發生在 w0=0；對其微分可得 dP[w0]，另其=0，用程式跑發現最大值發生在-0.0025(非常接近 0)，且 P[w0]最大值=0.398941



2.



3. 在 w0=w0map=-0.002501250625313034 時，使用程式跑出來 w1 最大值發生在 0.672836418209104

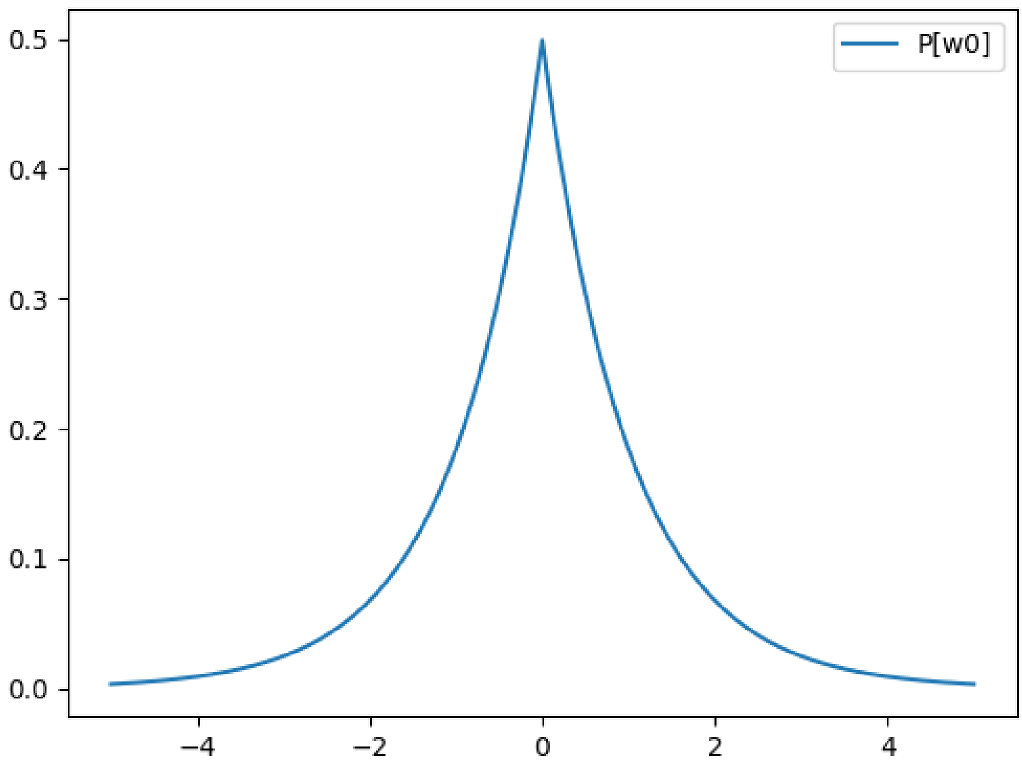
```
#3.1.3
w1max=np.argmax(likelihood(-0.002501250625313034,x)*p(x))
print('3.1.3:',x[w1max])#最大值發生在w1=0.672836418209104
```

4. 在 w0=w0map=-0.002501250625313034 時，posterior mean of w1 為 0.7042758569261673

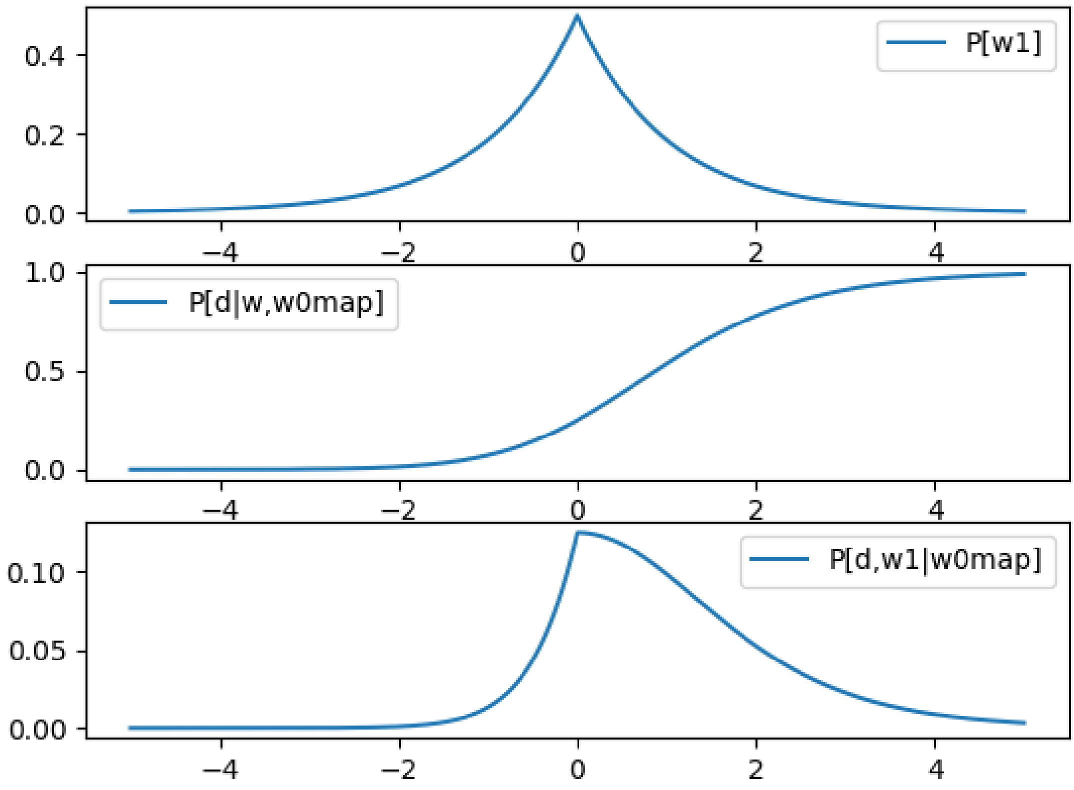
```
#3.1.4
temp=0
for i in range(len(x)):
    temp+=likelihood(-0.002501250625313034,x[i])*p(x[i])
posteriormean=0
for i in range(len(x)):
    posteriormean+=x[i]*likelihood(-0.002501250625313034,x[i])*p(x[i])
posteriormean/=temp
print('3.1.4:',posteriormean)#posterior mean為0.7042758569261673
```

Question 3.2:

1. 下圖橫軸為 w0，縱軸為 P[w0]之圖，由圖可知最大值發生在 w0=0；對其微分可得 dP[w0]，另其=0，用程式跑發現最大值發生在 0.002501250625312146 (非常接近 0)，且 P[w0]最大值=0.4987509374477932



2.



3. 以分析的角度去求最大值的 w1 的話，首先可先將 P[d,w1|w0map]拆成 w1>=0 與 w1<0 兩個部分，又因為 w0map 約等於 0，所以第一部分 w1>=0，可將式子約略化簡成$(\frac{1}{1+e^{-w1}})^2 * \frac{e^{-w1}}{2}$，接著再化簡成$\frac{0.5}{2+e^{w1}+e^{-w1}}$，因 exp(-w1)在 0~1 之間，所以主要看 exp(w1)項，而因為 w1 越大，exp(w1)越大，整個 function 值就會越小，所以當 w1 為 0 時， function 有最大值 1/8；接著看第二部分 w1<0，可將式子化簡成$(\frac{1}{1+e^{-w1}})^2 * \frac{e^{w1}}{2}$，接著進一步化簡成$\frac{0.5}{e^{-w1}+2e^{-2w1}+e^{-3w1}}$，當 w1 越來越小時，整個 function 值也會越來越小，而當 w1=0 時，function 有最大值 1/8，因此，根據夾擠定理(Squeeze theorem)，P[d,w1|w0map]的最大值為 1/8，且發生在 0 的位置。

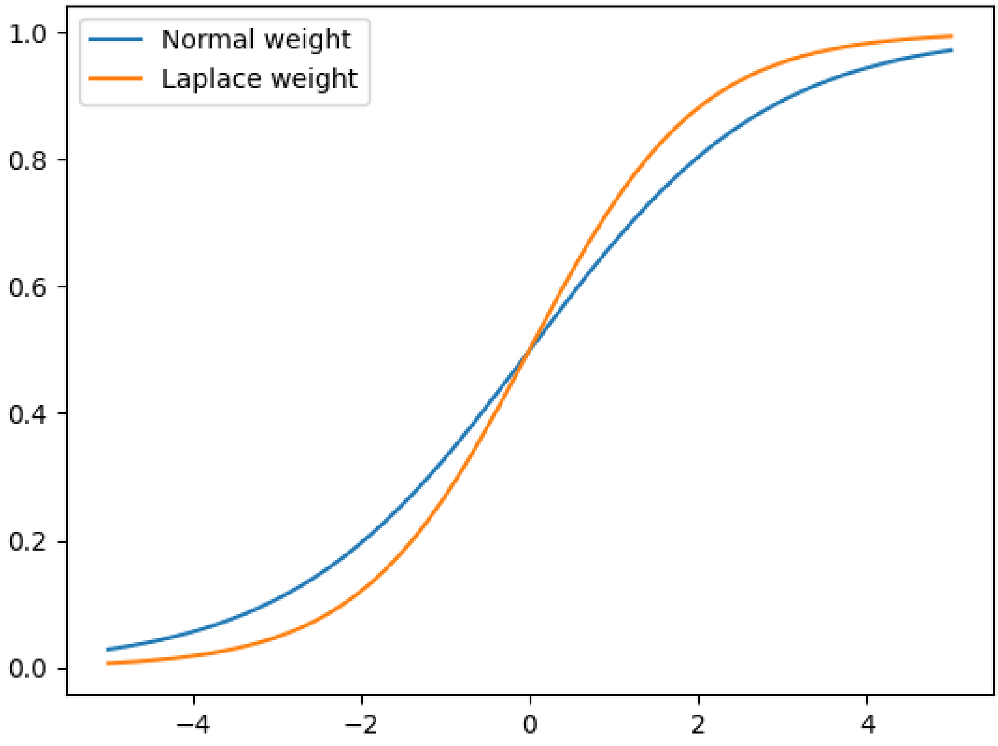
下圖為使用程式在 w0=w0map=0.002501250625312146 時， w1 最大值發生在 0.002501250625312146，且最大值為 0.12499960898510112，約等於 1/8

```
#3.2.3
w1max=np.argmax(likelihood(0.002501250625312146,x)*p(x))
print('3.2.3:',x[w1max])#最大值發生在w1=0.002501250625312146
```

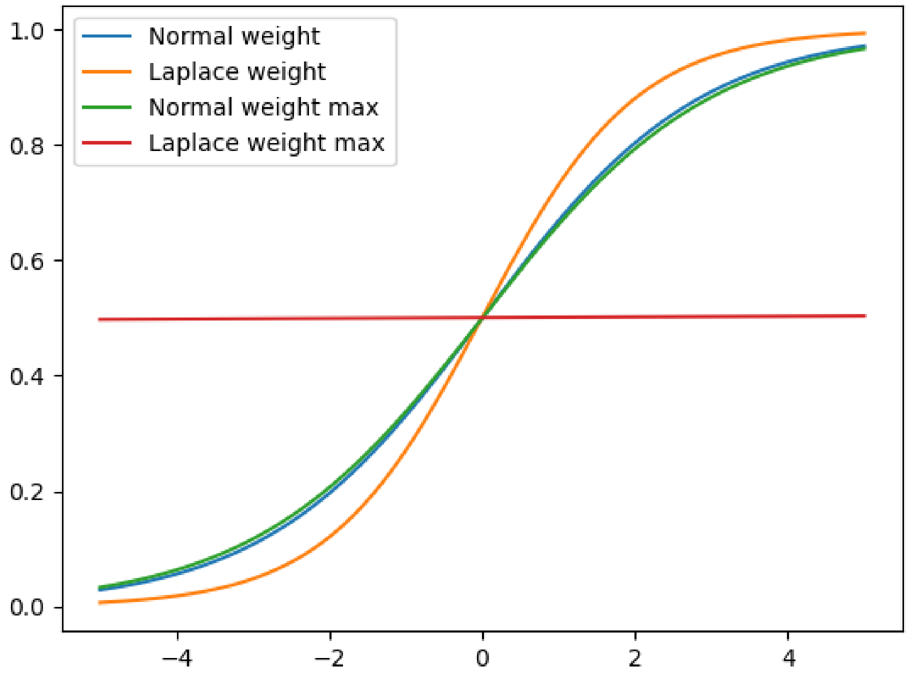
4. 在 w0=w0map=0.002501250625312146 時，posterior mean of w1 為 0.9963922458605604

```
#3.2.4
temp=0
for i in range(len(x)):
    temp+=likelihood(0.002501250625312146,x[i])*p(x[i])
posteriormean=0
for i in range(len(x)):
    posteriormean+=x[i]*likelihood(0.002501250625312146,x[i])*p(x[i])
posteriormean/=temp
print('3.2.4:',posteriormean)#posterior mean為0.9963922458605604
```

下圖為 P[class = Y | x, $w0$, $w1$]，其中 w0=w0map，w1=posterior mean，由圖中可看到，在 x = -1 時，Laplace weight prior = 0.270144218660314，Normal weight prior = 0.3303113723034824，皆小於 0.5，表示其皆可成功判斷在 x = -1 時，其有較大的概率不是 class Y，而其中，又以 Laplace weight prior 的表現比 Normal weight prior 要來的好一些。



下圖的綠色、紅色部分為用 w1=w1map 來做圖，可以看到 Laplace weight prior with w1map 已經變成一條直線了，若使用它來預測 class N 和 class Y 將會非常不準確，Normal weight prior with w1map 則是與 posterior mean 的 w1 相差不大



因為 P[w1|w0MAP,d] = c*P[d|w0MAP, w1] P[*w*1]，所以當 P[w1]為不同的 prior 時，會導致 P[w1|w0MAP,d]不同，而導致產生不同的 w1map 以及不同的 posterior mean of w1，預測的精確度也會跟著 w1 的不同而有所影響

在 small datasets 的情況下，一般而言標準差是會比較大的，若是所有抽樣的 class Y 的 datasets 的 x 值皆大於 0，那麼我們可能就會有先入為主想法去限定條件如 P[class = Y| x <0 ,w]=0，這會使 x<0 不產生任何機率，對於我們預測其結果來說是不利的，因為我們會損失掉所有 x<0 的可能性，因此，常態分布很常被使用，因為它可以集中小範圍的數據，且對於任何 x 值都保有一定的機率，永不歸零。