

ML HW2

B02902041 徐朝駿

1.(1%) Logistic regression function.

```
for i in range(numberofiter):
    # Hypothesis of answer
    hypothesis = np.dot(x, theta) + bias

    #fake loss
    loss = (1.0 / (1.0 + np.exp(-1.0*hypothesis)) ) - y.reshape((m, 1))

    #cost = -(y[i]*math.log(sigmoid(hypothesis))+(1-y[i])*math.log(sigmoid(1-hypothesis)))

    # Gradient
    gradient = sum(x* loss).reshape((57,1))
    grad_bias = sum(loss)

    # Adagrad
    ada += gradient**2
    ada_bias += grad_bias**2

    # Updating theta and bias
    theta = theta - alpha*gradient/(ada**0.5)
    bias = bias - alpha*grad_bias/(ada_bias**0.5)
```

使用 Adagrad，初始 Learning rate =0.1，loss 經過偏微分後，還沒乘上 w_i 的部份。原先是使用 sgd + regularization 的方法，但是發現他的速度還有收斂的情況較不穩定，最重要的是在把做 sgd 時的 for loop 拿掉之後，單純使用現在的矩陣運算，可以使得的運算速度快上 5~10 倍。

2.(1%) Describe your another method, and which one is best.

我使用的是上課時老師提到的 generative model。對於分類問題有不錯的成效，他的好處在於不需要做 training。只要擁有 label data 就可以很快的做出 model。但是正確率來說還是地一種的 logistic regression 較高。