Machine Learning 2020 Spring - HW2 Report

學號:B07902064 系級:資工二 姓名:蔡銘軒

1. (2%) 請比較實作的 generative model 及 logistic regression 的準確率，何者較佳？請解釋為何有這種情況？

	generative model	logistic regression
Accuracy	87.944	89.378

為了使比較有相同的基準，測量方式為將所有資料作為training set，並在public data set上探討準確率。從結果來看logistic regression的表現較好。

課堂影片李宏毅教授有提到在training data數量較多時，通常logistic regression的表現會比generative model要好。在這次的比較中，training data有兩萬筆左右，數量是足夠讓logistic regression發揮的。而generative model因為自己預設了distribution的前提、條件，可能使model與真正的data distribution有所差異，所以在這次資料量夠大時，表現比logistic regression要差。

2. (2%) 請實作 logistic regression 的正規化 (regularization)，並討論其對於你的模型準確率的影響。接著嘗試對正規項使用不同的權重 (lambda)，並討論其影響。

lambda/Accuracy	Testing set	Validation set(20%)
0	0.8854166666666666	0.8869
0.5	0.8852584960231381	0.8871
0.1	0.8852359002169198	0.8869
0.05	0.8854392624728851	0.8869

用於本次實驗的model為所有510個features取一次logistic regression進行3000次iterations。從結果來看，有沒有使用regularization對這個model而言並沒有太大的影響。當λ較大時，例如0.5，testing set上的表現略差於沒有regularization，但在validation set上則有小幅度的提升。當$\lambda = 0.1$時，testing set上表現又更差一點，且validation set上也沒有進步。而當$\lambda = 0.05$時，表現與沒有regularization沒有太明顯的差別。

Regularization主要的用途是避免較複雜的model發生overfitting，但這實驗使用的model本身的VC dimension相對於training data的數量而言本就不大，不太需要擔心overfitting的問題。對於簡單的model加入更多的限制，有時反而會阻礙training的過程，沒辦法得到更好的結果。

3. (1%) 請說明你實作的 best model，其訓練方式和準確率為何？

我的best model使用的訓練方式是gradient descent搭配Nadam，進行15000次的iterations。

在feature方面，通常實務上只要訓練的資料量大於model的VC dimension的10倍，就不會有很嚴重的overfitting。而這次提供的training data有兩萬筆左右，相對於總共510個features的線性model的VC dimension，資料量是很足夠的，因此我試著提高model的複雜度。

我觀察到510個features當中有7個是連續的，例如age, wage per hour, capital gains等等。我用trial and error的方式在model裡調整這些feature的次方，選出在validation set中表現最好的model。

	Accuracy	**Loss**
Testing set	0.8882863340563991	0.257687140685735
Validation set(20%)	0.8896	0.26826581981202857

4. (1%) 請實作輸入特徵標準化 (feature normalization)，並比較是否應用此技巧，會對於你的模型有何影響。

   此題的model為助教在colab上提供的logistic regression model。因為在我的best model中我將一些features的次方提高，在沒有feature normalization之下overflow的情形非常嚴重，準確率大幅度下降。

Testing set/Validation set(20%)	**With normalization**	**Without normalization**
Accuracy	0.8836166291214418/0.8733873940287504	0.7654925250870367/0.7640987836343531
Loss	0.27135543524640593/0.2896359675026287	4.081222904078705/4.103426184419094

   從以上的結果可以看出有feature normalization的情況下，不管在training set還是validation set上都有較好的結果。

   理論上有沒有做feature normalization對$Y = WX + b$形式的regression並沒有影響，因為對training data X的調整都可以被W以及b吸收。但有feature normalization的情況下，numerical stability會比較高。在這次的測試中，沒有進行feature normalization的model容易在sigmoid function的運算中發生overflow，可能是導致表現較差的原因。另外加上feature normalization後，運算速度也有所提升。在本機上我將兩個程式同時運作，有feature normalization的較快得到結果。

Collaborator: b07902047 羅啟帆