

# Machine Learning HW6

學號：r06921081 系級：電機碩一 姓名：張邵瑀

## 1. (1 %)請比較有無normalize的差別。並說明如何normalize.

normalize方法： $(y - \text{mean}_y) / \text{標準差}_y$

訓練參數：

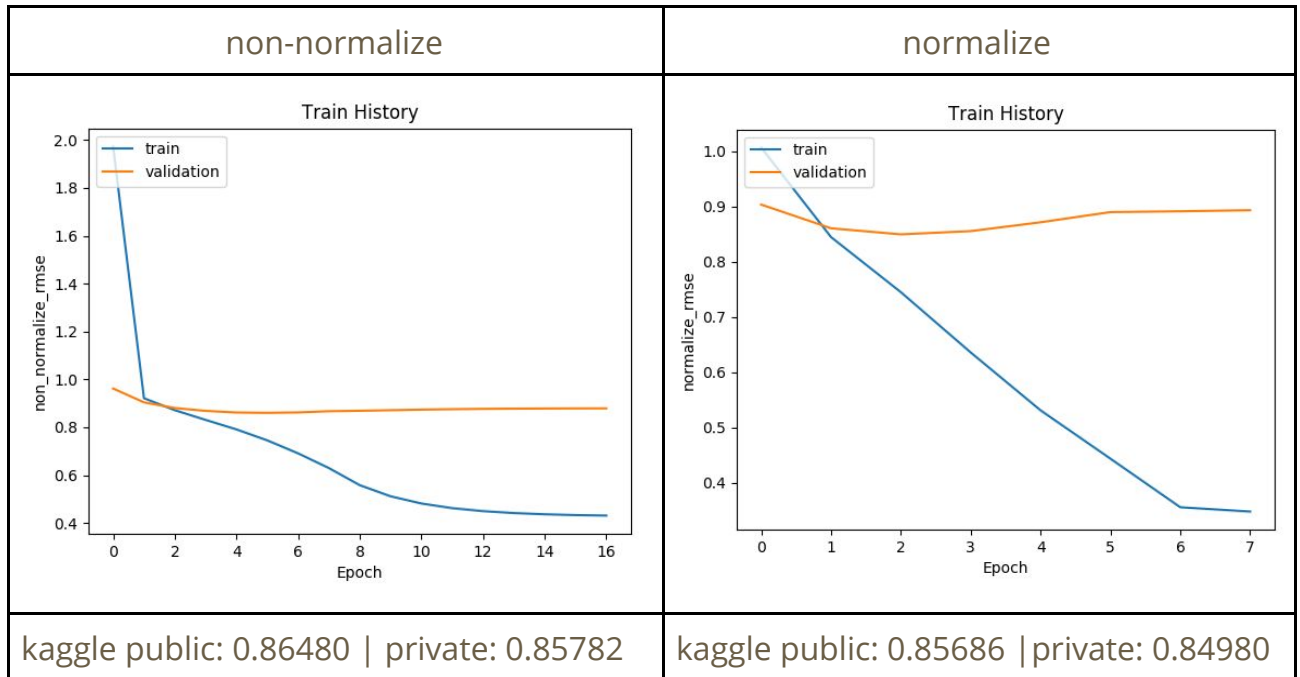
batch size: 512

embedding dimension: 256

optimizer: Adam(lr=0.00064)

validation rate: 0.1

首先先減掉平均之後除上標準差，可以發現有標準化過的model在前幾個epoch的loss很早就已經降到1左右了，所以normalize過的過程會是收斂比較快速的，大約在第2,3個epoch就已經到最低點了，而沒有normalize的則要大概到4~6個epoch才到最低點。



## 2. (1 %)比較不同的embedding dimension的結果。

邊我是取各個dimension做10次左右rmse在validation accuracy最好的一次結果上傳到KAGGLE的分數，可以看到為度不夠高的時候會有比較差的結果，大概在512為的時候會有很不錯的結果再往上加到1024結果反而變差，但這也有可能只是variance造成的誤差，但至少知道不會再更好了。

| dim            | 32      | 64      | 128     | 256     | 512     | 1024    |
|----------------|---------|---------|---------|---------|---------|---------|
| kaggle public  | 0.86303 | 0.86011 | 0.85940 | 0.85686 | 0.85642 | 0.85849 |
| kaggle private | 0.85576 | 0.85357 | 0.85077 | 0.84980 | 0.84800 | 0.85186 |

## 3. (1 %)比較有無bias的結果。

我比較bias時也有對該model做normalize，normalize方法： $(y - \text{mean}_y) / \text{標準差}_y$

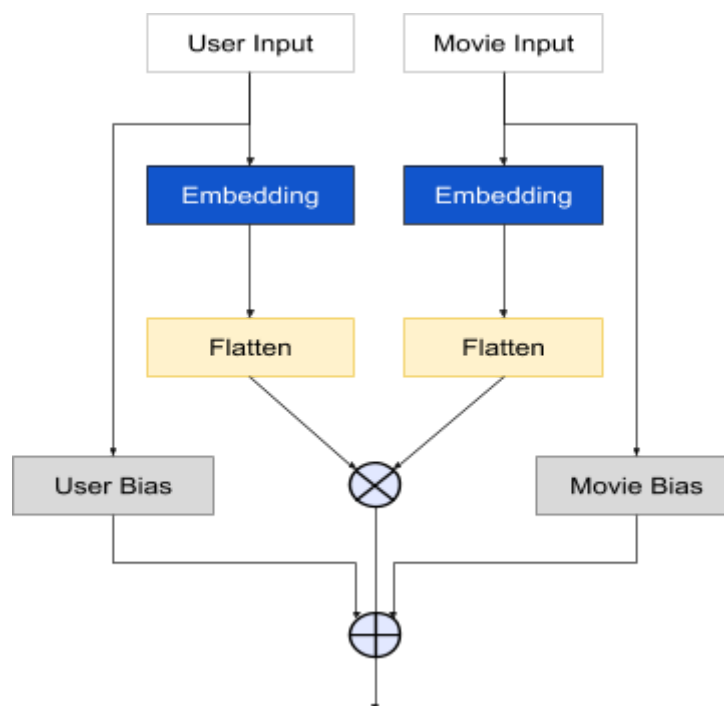
訓練參數：

batch size: 512

embedding dimension: 256

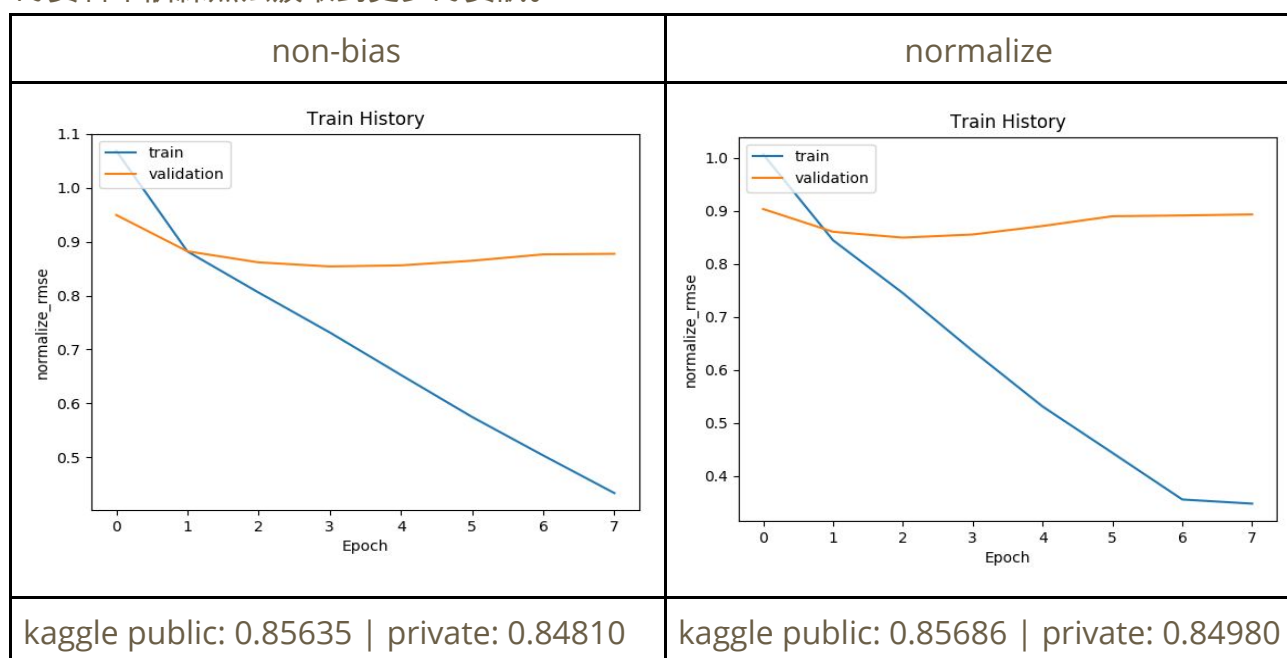
optimizer: Adam(lr=0.00064)

validation rate: 0.1

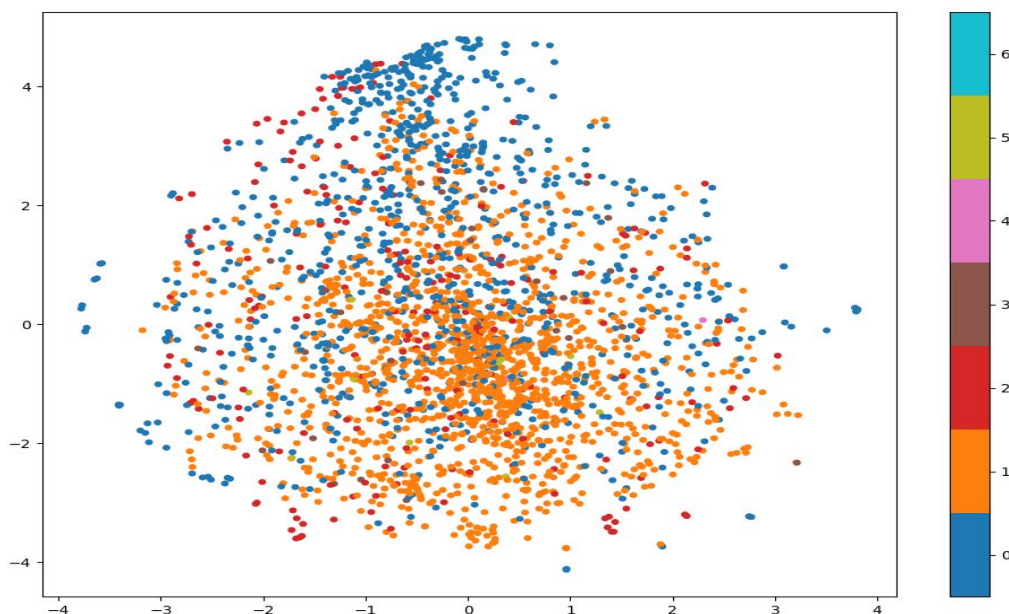


架構：

沒有加bias的結果甚至比我加入bias的結果還要來的好，我想應該是實驗的次數做的不夠多的關係，理論上擁有較多資訊的model會做的比較好，所以只是單純增加bias應該是無法起到太大的作用，抑或是我取的embedding維度已經足夠高到以未處理過的資料來訓練無法汲取到更多的資訊。



- (1 %)請試著將movie的embedding用tsne降維後，將movie category當作label來作圖。



上圖是用以下分組做出的圖

0: Action, Adventure, Horror

2: Animation, Children's, Sci-Fi

1: Crime, Documentary, Drama, Film-Noir, Musical, Thriller, Western

3: Romance

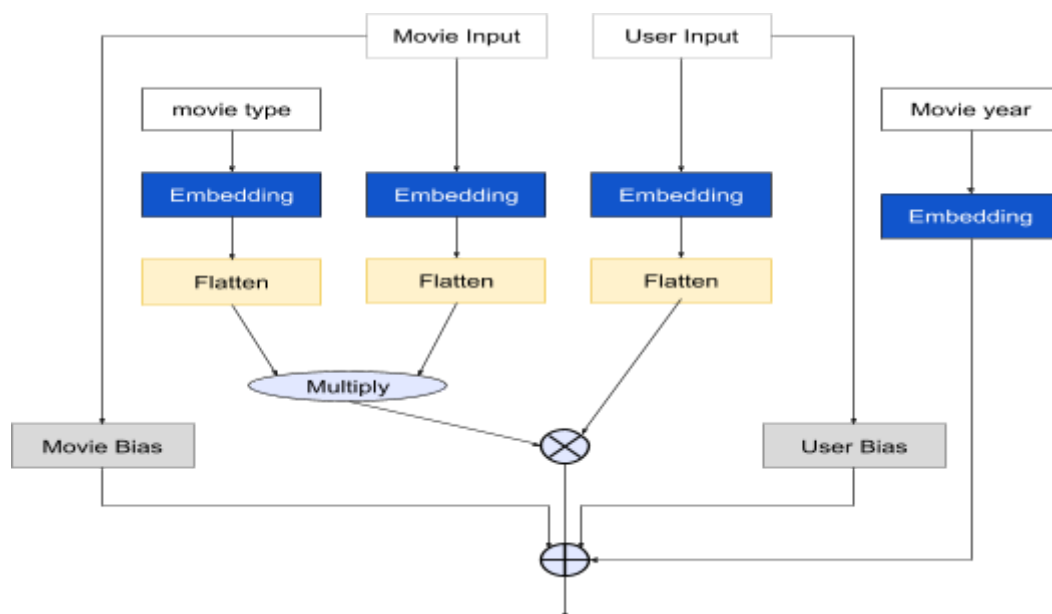
4: Fantasy

5: War

6: 其他分佈不明顯的(不顯示)

在18類下我觀察Action, Adventure, Horror三個分佈相近比較明顯，而Animation, Children's, Sci-Fi也是顏色分佈比較相近的一群，而分類1則是我抓出在中間分佈比較相近但卻跟第0類分很開的群體可以看出來都是劇情類的電影，跟動作片上有一定的區隔性，而紅色則是觀察到分佈較為靠近的一群又跟其他兩大類分的較開。

5. (1 %)試著使用除了rating以外的feature, 並說明你的作法和結果，結果好壞不會影響評分。



架構：

| kaggle          |                  |
|-----------------|------------------|
| public: 0.85449 | private: 0.84690 |

我的作法是把主要的movie type跟當作權重與movie embedding做點積相乘，再跟user embedding做內積，接著除了bias在Add層再把年份做1維的embedding加上，結果有比單純做MF好一點點，大概好0.002~0.003，原本也有把user資訊也如法泡製但結果更差就沒有採用了。