

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

(Collaborators:)

答：經過統計，得標準差約為 1.1169、平均值約為 3.5817。

因此在 training 前，先把 Rating 進行標準化。

Training 結束後，在 predict 後，將得到的結果，乘上標準化再加平均值。

而有無進行標準化所得到的 Error 分別為

Without Normalize	1.65507/1.64468
With Normalize	1.50879/1.51313

可以看出 Normalize 確實是對預測的準確度有所一定的幫助

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

(Collaborators:)

答：

dimesion	3	16	120	256
loss	1.65507/1.64468	1.93649/1.93592	1.83852/1.82838	1.81972/1.80642

在本項測試中，loss 並沒有隨著 dimension 數量呈線性變化。

因此，未必 dimension 開越大就越好。

實際上仍需要進行測試、相互比較才能達到較佳的結果。

3. (1%) 比較有無 bias 的結果。

(Collaborators:)

答：

Without Bias	1.65507/1.64468
With Bias	1.63236/1.63753

由此可看出，加入 bias 對整體表現沒有太大影響。

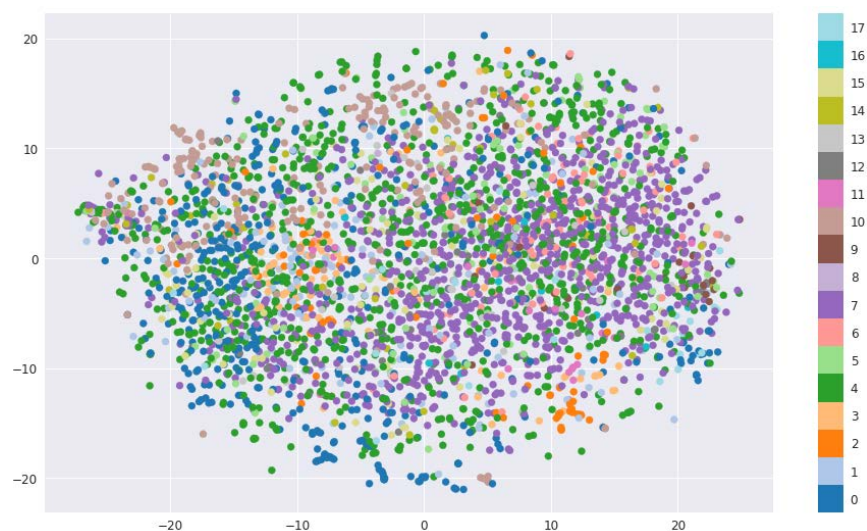
即便如此，加入 bias 能微幅提升整體表現也是不爭的事實。

因此，雖然加入 bias 並不能當作提升 performance 的關鍵策略，但在沒有其他辦法時，也不失為一個拉升 performance 的 trick

4. (1%) 請試著將 movie 的 embedding 用 tsne 降維後，將 movie caegoray 當作 label 來作圖。

(Collaborators:)

答：



由此圖可看出:

綠色(Comedy)幾乎遍布整張圖；

而藍色(Action)與紫色(Drama)則分別佔據圖的左右。

	class	genres
0	0	Action
1	1	Adventure
2	2	Animation
3	3	Children's
4	4	Comedy
5	5	Crime
6	6	Documentary
7	7	Drama
8	8	Fantasy
9	9	Film-Noir
10	10	Horror
11	11	Musical
12	12	Mystery
13	13	Romance
14	14	Sci-Fi
15	15	Thriller
16	16	War
17	17	Western

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

(Collaborators:)

答：

Reference:

<https://nipunbatra.github.io/blog/2017/recommend-keras.html>

<https://github.com/bradleypallen/keras-movielens-cf>

<https://github.com/thtang/ML2017FALL/>