

# Machine Learning HW6 Report

學號：B05611038 系籍：生機二 姓名：張育堂

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

(collaborator : No)

Matrix factorization model structure :

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 1)	0	
input_2 (InputLayer)	(None, 1)	0	
embedding_1 (Embedding)	(None, 1, 128)	773248	input_1[0][0]
embedding_3 (Embedding)	(None, 1, 128)	505984	input_2[0][0]
embedding_2 (Embedding)	(None, 1, 1)	6041	input_1[0][0]
embedding_4 (Embedding)	(None, 1, 1)	3953	input_2[0][0]
dropout_1 (Dropout)	(None, 1, 128)	0	embedding_1[0][0]
dropout_3 (Dropout)	(None, 1, 128)	0	embedding_3[0][0]
dropout_2 (Dropout)	(None, 1, 1)	0	embedding_2[0][0]
dropout_4 (Dropout)	(None, 1, 1)	0	embedding_4[0][0]
reshape_1 (Reshape)	(None, 1, 128)	0	dropout_1[0][0]
reshape_2 (Reshape)	(None, 1, 128)	0	dropout_3[0][0]
flatten_1 (Flatten)	(None, 1)	0	dropout_2[0][0]
flatten_2 (Flatten)	(None, 1)	0	dropout_4[0][0]
dot_1 (Dot)	(None, 1, 1)	0	reshape_1[0][0] reshape_2[0][0]
add_1 (Add)	(None, 1, 1)	0	flatten_1[0][0] flatten_2[0][0] dot_1[0][0]
flatten_3 (Flatten)	(None, 1)	0	add_1[0][0]
dense_1 (Dense)	(None, 1)	2	flatten_3[0][0]
Total params: 1,289,228			
Trainable params: 1,289,228			
Non-trainable params: 0			

	no normalize	Normalize
Public score	0.85473	0.94924

我的 normalize 做法是把 keras 中 Dot 層的參數 normalize 設為 True，而它做法是對要點積的 dimension 進行 L2 標準化，而有設置 normalize 的 model 表現比較差，我想原因是因為我們最後 output 的結果是 movie rating，而它代表的意義也只有分數類的排名，點積 normalize 後輸出所不足的地方就要由 bias 去補足，而同時也會喪失當初做 matrix factorization 的意義，所以表現才會比較差。

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

(collaborator : No)

Dimension	64	128	256	512
Public score	0.86000	0.85498	0.85661	0.85995

由分數的高低可以看出，其實並非一味地將 embedding dimension 設大就可以了，雖然理論上將 dimension 設高可以將複雜的數據表示得更好，但是這種作法同時也會造成 model 有 overfitting 的狀況發生，所以應該依照自己數據的性質選擇適合的 embedding dimension。

3. (1 %)比較有無 bias 的結果。

(collaborator : No)

	bias	no bias
Public score	0.85381	0.85349

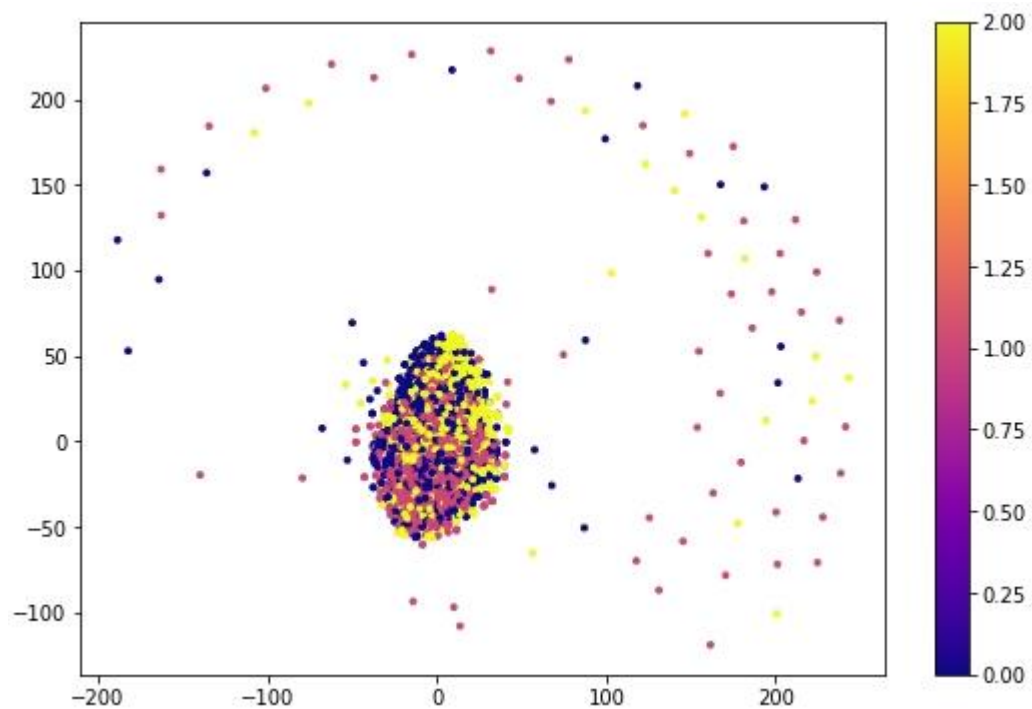
依照理論來看這結果的話，有點不符常理，反而是沒有放上 bias 的 model 表現比較好。因為照一般的理論，在做 matrix factorization 的時候，每一個人的個人因素或者是電影本身的性質，總是會有點積的時候無法預測到的地方，所以依照常理，有放上 bias 的 model 會表現較好。觀察兩個 model 的表現，雖然 no bias 的表現較好，但其實兩者的分數接近，也不至於到不合理的地步。

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

(collaborator : No)

我將 movie embedding layer 的參數取出來，也就是上圖 embedding\_3 的參數，將 movie\_num \* embedding\_dimension 丟進 sklearn 的 TSNE 並解將之降至 2 維，label 是讀取 movie 的類型之後，因為一個電影有多個類型，而挑選我是用亂數去挑選出一個類型之後，並根據它的類型上 label。

Label 的部分我是將 Animation、Children's、Comedy、Fantasy、Western 標為 0，Documentary、Drama、Film-Noir、Musical、Romance、War 標為 1，剩下的則標為 2，而結果如下圖。



雖然 model 的表現是有過 strong baseline 的，但是由圖的分布的狀況可知，其實 embedding 層的結果並沒有可以明顯看出分類的狀況，所以其實 model 內部的狀況就算用優秀的演算法降維之後，還是很難用人眼判斷。

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。

(collaborator : No)

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 4)	0	
input_2 (InputLayer)	(None, 1)	0	
embedding_1 (Embedding)	(None, 4, 128)	773248	input_1[0][0]
embedding_3 (Embedding)	(None, 1, 512)	2023936	input_2[0][0]
embedding_2 (Embedding)	(None, 4, 1)	6041	input_1[0][0]
embedding_4 (Embedding)	(None, 1, 1)	3953	input_2[0][0]
dropout_1 (Dropout)	(None, 4, 128)	0	embedding_1[0][0]
dropout_3 (Dropout)	(None, 1, 512)	0	embedding_3[0][0]
dropout_2 (Dropout)	(None, 4, 1)	0	embedding_2[0][0]
dropout_4 (Dropout)	(None, 1, 1)	0	embedding_4[0][0]
reshape_1 (Reshape)	(None, 1, 512)	0	dropout_1[0][0]
reshape_2 (Reshape)	(None, 1, 512)	0	dropout_3[0][0]
flatten_1 (Flatten)	(None, 4)	0	dropout_2[0][0]
flatten_2 (Flatten)	(None, 1)	0	dropout_4[0][0]
dot_1 (Dot)	(None, 1, 1)	0	reshape_1[0][0] reshape_2[0][0]
add_1 (Add)	(None, 1, 4)	0	flatten_1[0][0] flatten_2[0][0] dot_1[0][0]
flatten_3 (Flatten)	(None, 4)	0	add_1[0][0]
dense_1 (Dense)	(None, 1)	5	flatten_3[0][0]
Total params: 2,807,183			
Trainable params: 2,807,183			
Non-trainable params: 0			

Public score : 0.86193

這是我使用 `user.csv` 的資料，除了郵遞區號以外全用的 `model`，因為還是使用 `matrix factorization`，所以將 `movie` 的 `embedding dimension` 增加到可以內積的程度，而最後表現並沒有比單純使用 `train.csv` 資料來的好。

我想原因可能是因為，像是用戶個人資料之類的，應該只在 `bias` 層使用，而 `movie` 的資料也可以考慮使用此方法做輔助，只增加在 `bias` 層更能表示此種 `model` 加上 `bias` 的意義，而這也讓我知道其實雖然建構 `model` 的方式有千百種，但是合適的審視 `data` 的性質真的是在做 `machine learning` 跟 `deep learning` 最重要的事情。