學號：B03902016 系級： 資工四 姓名：周聖筌

1. (1%)請比較有無normalize(rating)的差別。並說明如何normalize.
   normalize方法：(x – mean(x)) / std(x)
   latent dimension：256維
   以kaggle上的public score來比較
   沒有normalize：0.85886
   有做normalize：0.86609
   做normalize並沒有變好，反而變差了。

2. (1%)比較不同的latent dimension的結果。
   以我自己切的validation rmse來比較結果(切法：取資料時先shuffle，再將後10%當作validation set)

Latent dimension	Validation rmse
128	0.8577
256	0.8562
512	0.8609
1024	0.8653

   由此可以得知太大或太小的latent dimenstion都會造成結果變差，經過嘗試後，256應該是最佳的結果。

3. (1%)比較有無bias的結果。
   以我自己切的validation rmse來比較結果(切法和上述相同)，且所有的model都是256 latent dimention。

	Validation rmse
同時加movie和user的bias	0.8562
只加movie的bias	0.8578
只加user的bias	0.8572
沒加bias	0.8605

   發現同時加movie和user的bias結果會最好，因為每個user會有自己的評分習慣，而好的電影會被大家評的比較高分。

4. (1%)請試著用DNN來解決這個問題，並且說明實做的方法(方法不限)。
並比較MF和NN的結果，討論結果的差異。

DNN的model summary

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 1)	0	
input_2 (InputLayer)	(None, 1)	0	
embedding_1 (Embedding)	(None, 1, 500)	3020500	input_1[0][0]
embedding_2 (Embedding)	(None, 1, 500)	1976500	input_2[0][0]
flatten_1 (Flatten)	(None, 500)	0	embedding_1[0][0]
flatten_2 (Flatten)	(None, 500)	0	embedding_2[0][0]
dropout_1 (Dropout)	(None, 500)	0	flatten_1[0][0]
dropout_2 (Dropout)	(None, 500)	0	flatten_2[0][0]
concatenate_1 (Concatenate)	(None, 1000)	0	dropout_1[0][0] dropout_2[0][0]
dense_1 (Dense)	(None, 128)	128128	concatenate_1[0][0]
dropout_3 (Dropout)	(None, 128)	0	dense_1[0][0]
batch_normalization_1 (BatchNor	(None, 128)	512	dropout_3[0][0]
dense_2 (Dense)	(None, 256)	33024	batch_normalization_1[0][0]
dropout_4 (Dropout)	(None, 256)	0	dense_2[0][0]
batch_normalization_2 (BatchNor	(None, 256)	1024	dropout_4[0][0]
dense_3 (Dense)	(None, 512)	131584	batch_normalization_2[0][0]
dropout_5 (Dropout)	(None, 512)	0	dense_3[0][0]
batch_normalization_3 (BatchNor	(None, 512)	2048	dropout_5[0][0]
dense_4 (Dense)	(None, 1)	513	batch_normalization_3[0][0]

Total params: 5,293,833
Trainable params: 5,292,041
Non-trainable params: 1,792

以kaggle上的public score來比較。

MF：0.85886

NN：0.87792

經過許多嘗試還是無法超越MF。

5. (1%)請試著將movie的embedding用tsne降維後，將movie category當作label來作圖。

   我的分類：

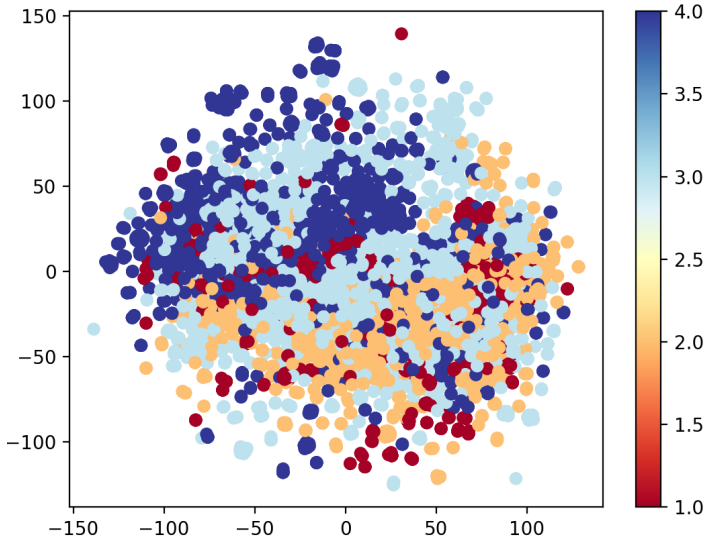
   1 : [“Thriller" , "Horror" , "Crime" , "Mystery" , “Film-Noir"]

   2 : ["Drama" , "Musical" , "Romance"]

   3 : ["Children's" , “Animation" , "Comedy"]

   4 : ["War" , "Action" , "Sci-Fi" , "Fantasy" , "Adventure" , "Documentary" , “Western”]

   結果：可以看到第2類和第4類分得比較好、比較集中，第1類和第3類就比較分散。



6. (BONUS)(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。

   我的作法是將user的embedding加上了年齡和性別資訊。

   在讀測資時，多開兩個age和gender的陣列儲存對應的資料(gender男生給0 女生給1)

   在embedding時，age給一個4維embedding，gender給一個60維embedding，而原先的userid就是256 – 64 = 192維的embedding。

   將上述三個embedding concate起來後和movie的embedding做dot，同時也還是會add user和 movie的 bias。

   結果在public score進步到0.85649的成績。