

MF、DNN model summary 在最後一頁(MF:沒有特別提的話都是後面附上那個 model)

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

使用最直覺的作法,將 rating- mean/std, mean=3, std=2, 將 rating normalize 到-1~1 之間, 使用 normalize 之後,結果只好了了一點點。

有無 normalize	public	private
無	0.84144	0.84236
有	0.84022	0.84105

2. (1%)比較不同的 latent dimension 的結果。

Number of Latent dimension	public	private
16	0.85861	0.85846
32	0.84945	0.84939
64	0.84292	0.84327
128	0.84144	0.84236

latent dimension 越小結果越差,可能原因是參數量不夠,但不斷增加也會有個極限,到達那一個最大的 latent dimension 之後,結果應該會不升反降

3. (1%)比較有無 bias 的結果。

有無 bias	public	private
無	0.84668	0.84683
有	0.84144	0.84236

由於 user 和 movie 都有各自的 rating 傾向,導致增加 bias 的情況下可以使得結果更好

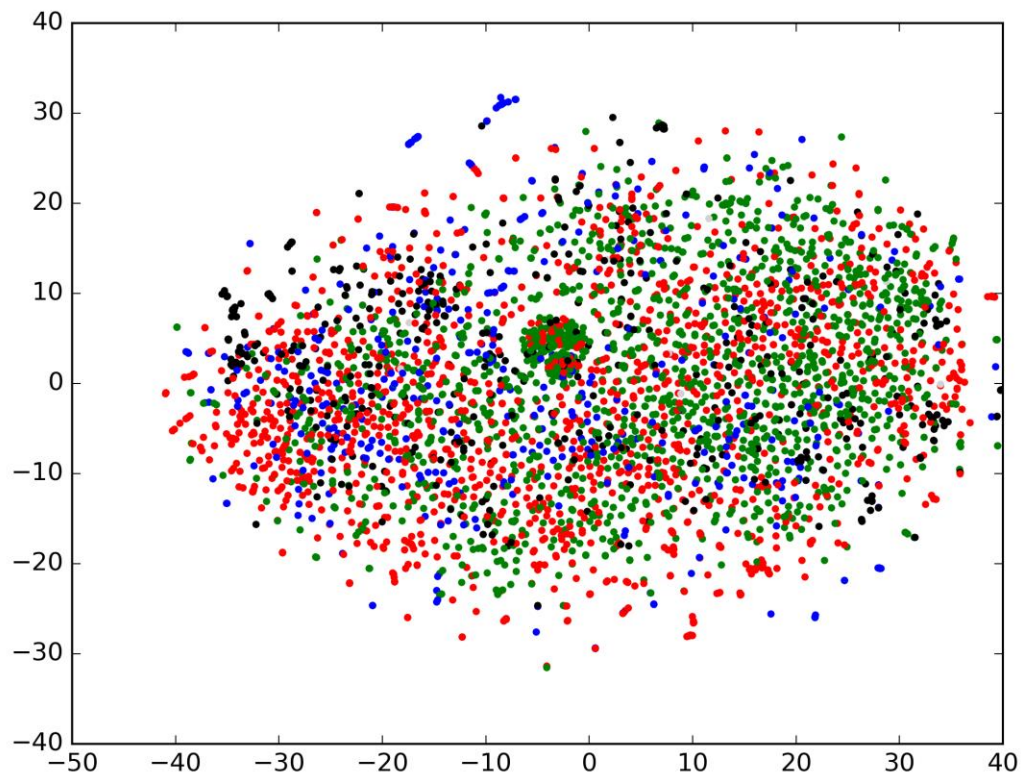
4. (1%)請試著用 DNN 來解決這個問題,並且說明實做的方法(方法不限)。並比較 MF 和 NN 的結果,討論結果的差異。

Model	public	private
MF	0.84144	0.84236
DNN	0.85838	0.85970

兩者使用的參數量故意設計成差不多,但是沒有認真 tune DNN,故結果較糟,另外一個可能的原因其實跟語音處理很像,把三個 model 的結果和 DNN train 全部的結果來比也是較差的,但 DNN 的極限就差不多跟三個 model train 的差不多,故使我猜想這次作業使用 DNN 的極限大概就跟 MF 差不多。

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

來作圖。



Color	Genre
Red	Animation, Children' s, Comedy, Adventure
Green	Romance, Drama, Documentary, Musical
Blue	Fantasy, Action, Sci-Fi, War, Western
Black	Crime, Thriller, Horror, Film-Noir
Gray	Other

(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。

我把助教給的其他 data 也一起拿來使用, 分別是 user occupations 和 movie genres, 加上原本 userID 和 moiveID 共四種之後兩兩取 dot 並 concatenate 起來當作 nn(一層當作 output layer)的 input 去 learn 且加入 userID 和 moiveID 當作 bias 之後我去比較可能全部六種 (C4 取 2) 或少了哪幾個 dot 結果會比較準確, 實驗結果發現少了 user occupations 和 userID 的 dot 結果會是最好實驗數據如下:

Model	Public	private
4features+MF+NN	0.84082	0.84150

MF:

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_2 (Embedding)	(None, 1, 128)	505984	input_2[0][0]
input_6 (InputLayer)	(None, 18)	0	
input_5 (InputLayer)	(None, 21)	0	
flatten_1 (Flatten)	(None, 128)	0	embedding_1[0][0]
flatten_2 (Flatten)	(None, 128)	0	embedding_2[0][0]
dense_2 (Dense)	(None, 128)	2432	input_6[0][0]
dense_1 (Dense)	(None, 128)	2816	input_5[0][0]
dropout_1 (Dropout)	(None, 128)	0	flatten_1[0][0]
dropout_2 (Dropout)	(None, 128)	0	flatten_2[0][0]
dropout_4 (Dropout)	(None, 128)	0	dense_2[0][0]
dropout_3 (Dropout)	(None, 128)	0	dense_1[0][0]
dot_1 (Dot)	(None, 1)	0	dropout_1[0][0] dropout_2[0][0]
dot_2 (Dot)	(None, 1)	0	dropout_1[0][0] dropout_4[0][0]
dot_3 (Dot)	(None, 1)	0	dropout_2[0][0] dropout_3[0][0]
dot_4 (Dot)	(None, 1)	0	dropout_2[0][0] dropout_4[0][0]
dot_5 (Dot)	(None, 1)	0	dropout_3[0][0] dropout_4[0][0]

input_3 (InputLayer)	(None, 1)	0	
input_4 (InputLayer)	(None, 1)	0	
embedding_3 (Embedding)	(None, 1, 1)	6041	input_1[0][0]
embedding_4 (Embedding)	(None, 1, 1)	3953	input_2[0][0]
concatenate_1 (Concatenate)	(None, 7)	0	dot_1[0][0] dot_2[0][0] dot_3[0][0] dot_4[0][0] dot_5[0][0] input_3[0][0] input_4[0][0]
flatten_3 (Flatten)	(None, 1)	0	embedding_3[0][0]
flatten_4 (Flatten)	(None, 1)	0	embedding_4[0][0]
dense_3 (Dense)	(None, 1)	8	concatenate_1[0][0]
add_1 (Add)	(None, 1)	0	flatten_3[0][0] flatten_4[0][0] dense_3[0][0]

=====
Total params: 1,294,482

Trainable params: 1,294,482

Non-trainable params: 0

DNN:

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_2 (Embedding)	(None, 1, 128)	505984	input_2[0][0]
input_5 (InputLayer)	(None, 21)	0	

input_6 (InputLayer)	(None, 18)	0	
flatten_1 (Flatten)	(None, 128)	0	embedding_1[0][0]
flatten_2 (Flatten)	(None, 128)	0	embedding_2[0][0]
dense_1 (Dense)	(None, 128)	2816	input_5[0][0]
dense_2 (Dense)	(None, 128)	2432	input_6[0][0]
dropout_1 (Dropout)	(None, 128)	0	flatten_1[0][0]
dropout_2 (Dropout)	(None, 128)	0	flatten_2[0][0]
dropout_3 (Dropout)	(None, 128)	0	dense_1[0][0]
dropout_4 (Dropout)	(None, 128)	0	dense_2[0][0]
input_3 (InputLayer)	(None, 1)	0	
input_4 (InputLayer)	(None, 1)	0	
concatenate_1 (Concatenate)	(None, 514)	0	dropout_1[0][0] dropout_2[0][0] dropout_3[0][0] dropout_4[0][0] input_3[0][0] input_4[0][0]
dense_3 (Dense)	(None, 128)	65920	concatenate_1[0][0]
dense_4 (Dense)	(None, 64)	8256	dense_3[0][0]
dense_5 (Dense)	(None, 32)	2080	dense_4[0][0]
dense_6 (Dense)	(None, 1)	33	dense_5[0][0]
=====			
Total params: 1,360,769			
Trainable params: 1,360,769			
Non-trainable params: 0			