

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

答：

我 **normalize** 的方式是 $\text{rating} / 5$ 。我也試過用 $(\text{rating}-2.5) / 5$ ，發現得到的 **validation MSE**（**validation set** 會隨機抽 **training data** 的 0.1）比較大（0.1 以上），因此以下的結果是使用 $\text{rating} / 5$ 所得。

Normalize 的 **model** 比較快 **train**，且 **MSE** 較小：**normalize** 的 **model** 只過大概 5 個 **epoch** 就 **converge** 在 0.03（**validation MSE**），沒有 **normalize** 的 **model** 過了 10 個 **epoch** 只 **converge** 在 0.75。

2. (1%)比較不同的 **latent dimension** 的結果。
(collaborator:無)

答：

我用 $\text{batch size}=200$ ， $\text{dimension size}=1,2,5,10,25,50$ 的上題 **normalized model**，分別得到的 **validation MSE** 是 0.0367、0.0321、0.0308、0.0305、0.0303、0.0308，除了 $\text{dimension size}=1,2$ 的時候比較低之外，沒有顯著差異，因此這個 **model** 比較不受 **latent dimension** 影響。

3. (1%)比較有無 **bias** 的結果。
(collaborator:無)

答：

我用 $\text{batch size}=200$ ， $\text{dimension size}=25,50$ 的 **normalized model**，在有 **bias** 的時候 **validation MSE** 是 0.0303、0.0308，移除掉 **bias** 的時候 **validation MSE** 增加到 0.0318、0.0328，結果顯示有 **bias** 會讓這個 **model** 比較準確。

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。

(collaborator:無)

答：

Layer (type)	Output Shape	Param #	Connected to
input_55 (InputLayer)	(None, 1)	0	
input_56 (InputLayer)	(None, 1)	0	
embedding_39 (Embedding)	(None, 1, 50)	302050	input_55[0][0]
embedding_40 (Embedding)	(None, 1, 50)	184450	input_56[0][0]
flatten_39 (Flatten)	(None, 50)	0	embedding_39[0][0]
flatten_40 (Flatten)	(None, 50)	0	embedding_40[0][0]
concatenate_11 (Concatenate)	(None, 100)	0	flatten_39[0][0] flatten_40[0][0]
dense_47 (Dense)	(None, 50)	5050	concatenate_11[0][0]
dense_48 (Dense)	(None, 50)	2550	dense_47[0][0]
dense_49 (Dense)	(None, 1)	51	dense_48[0][0]
Total params: 494,151 Trainable params: 494,151 Non-trainable params: 0			

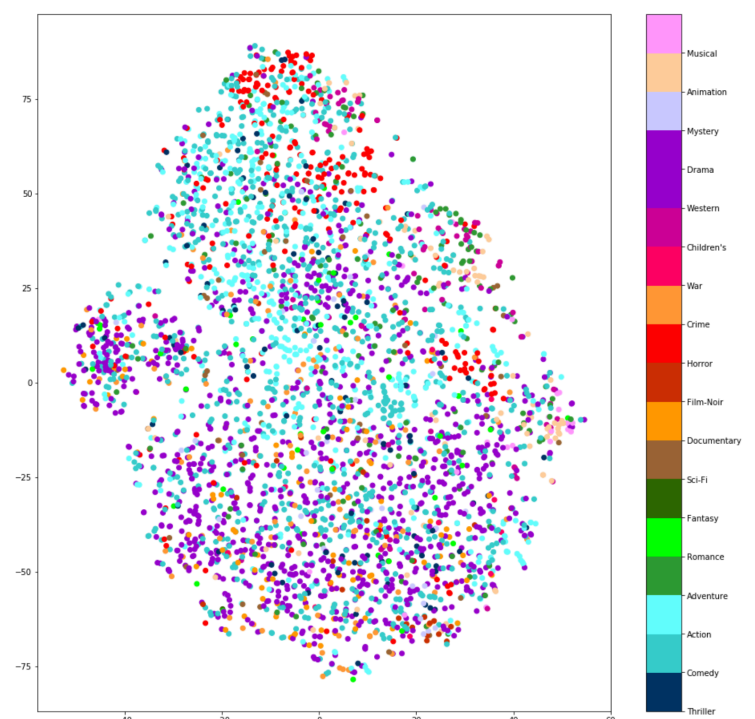
上圖是我的 DNN model，將 user、movie embedding concatenate 在一起後，丟到兩個 size=50 的 hidden layer 中，最後再得到一個 scalar output，loss 使用 MSE 去 train。這個方法得到的 validation MSE 為 0.0335，比 matrix factorization 的 0.03080 還差。

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

(collaborator:無)

答：

右圖是我使用第 1 題的 model 所得的 embedding，去做 t-SNE 所得到的結果，可以發現各個 category 混雜在一起，看來 model train 出來的 embedding 不是根據 category 在做區分。



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。

(collaborator:無)

答:

我改了一下第 4 題的 DNN, 在 movie embedding 上 concatenate 「bag of categories」 vector, 在 user embedding 上 concatenate 性別、職業的 one-hot vector 以及 normalized age 數字 (年紀 / 60), 接著再把這一長條 vector 餵進 feedforward layers, 以 MSE 為 loss 做 training。得到的 validation MSE 是 0.0328, 比原先的 0.0335 好一些。