

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

(collaborator:)

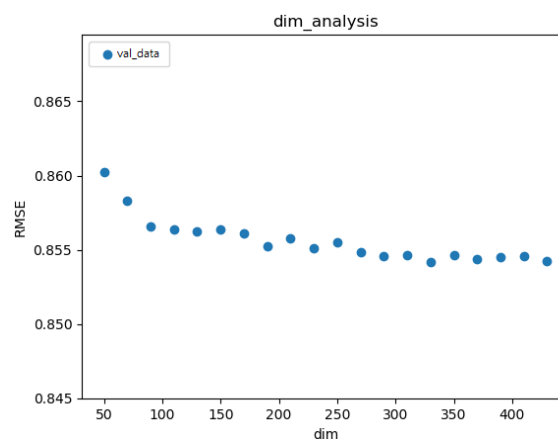
	Normalized	Without Normalized
epoch	43	103
Validation RMSE	0.8479	0.8554

先求出 rating 的 mean 與 standard deviations，接著把 (rating-mean)/std
在 measure RMSE 時則是將 $y_{pred} * std + mean$ 。

可以發現到有 Normalized 的狀況下收斂速度較快，雖然 performance 較沒有 normalized 的好，但把 learning rate 調低的狀況下沒有 normalized 的 performance 可以跟有 normalized 的差不多，相對的 epoch 會比 103 大出很多(因為要比較故設定都相同)

2. (1%)比較不同的 latent dimension 的結果。

(collaborator:)



可以發現到 latent dimension 增加之後會有下降，但到一定的值之後就不會有明顯的改變，而且記憶體 cost 會大的很多，其實不是很划算。

3. (1%)比較有無 bias 的結果。

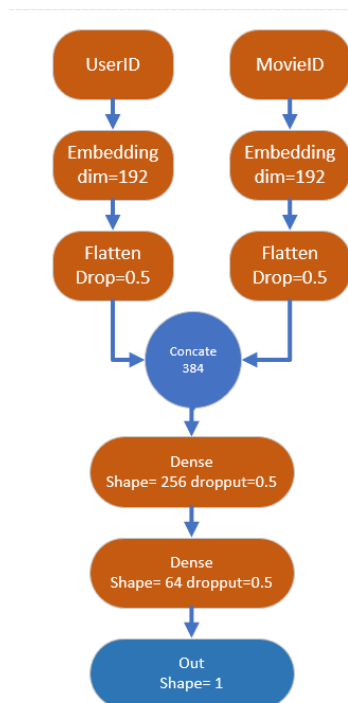
(collaborator:)

	With bias	Without bias
epoch	103	169
Validation RMSE	0.8554	0.8714

可以發現在沒有 bias 的狀況下，可以發現 validation 的 RMSE 變高了(在與第一題相同 setting 的狀況下)，可能是因為在加入 bias 的狀況下(glorot_normal 的 initializer)可以使這個網路更加彈性。

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:)



	MF	NN
epoch	103	56
Validation RMSE	0.8554	0.8663

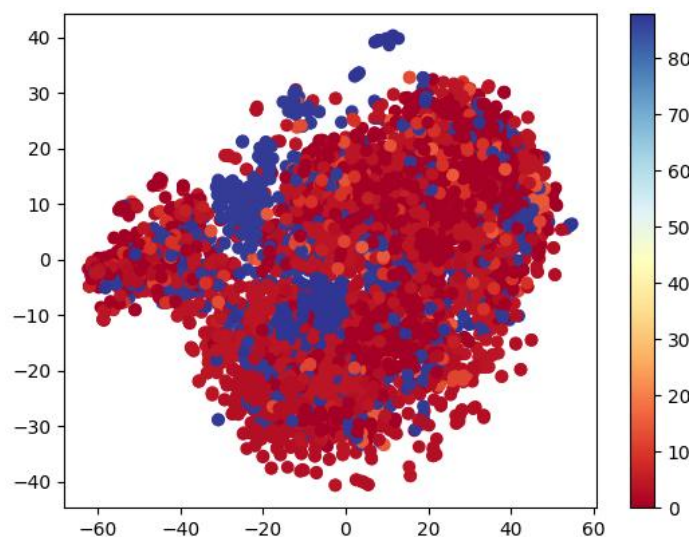
可以發現到 NN 的效果並沒有比 MF 差很多，而且相對的收斂比較快，可能在於已經 concat 了具有意義的 Feature 並且進去訓練，故出來的結果比第三題沒有 bias 的狀況下好一些。

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

(collaborator:)

分類如下：

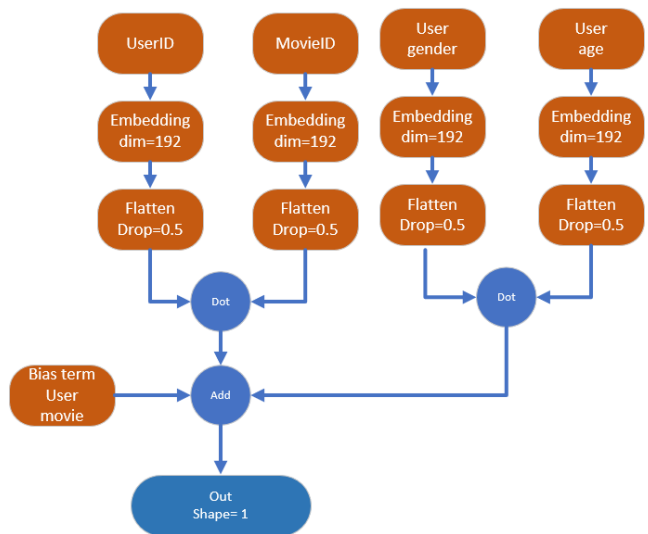
Drama, musical,
 Animation, adventure, Cchildren's
 Action,war
 Comedy
 Romance
 Crime, horror, thriller
 Documentary
 Western
 Sci-FI
 Film-Noir
 Mystery, Fantasy



(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:)

	MF	new feature
epoch	103	46
Validation RMSE	0.8554	0.8733



我透過把 user gender、user age 加近來 embedding, 再內積然後與原本架構的內積相加。可以發現到實驗效果並不會提升, 甚至下降。可能是因為主要的 feature 比較著重於 UserID、MovieID 與 User gender、User age 就相對較低。