

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

(collaborator:盧玄真、施宣安、劉紹增)

以下為我 normalize 的方法:

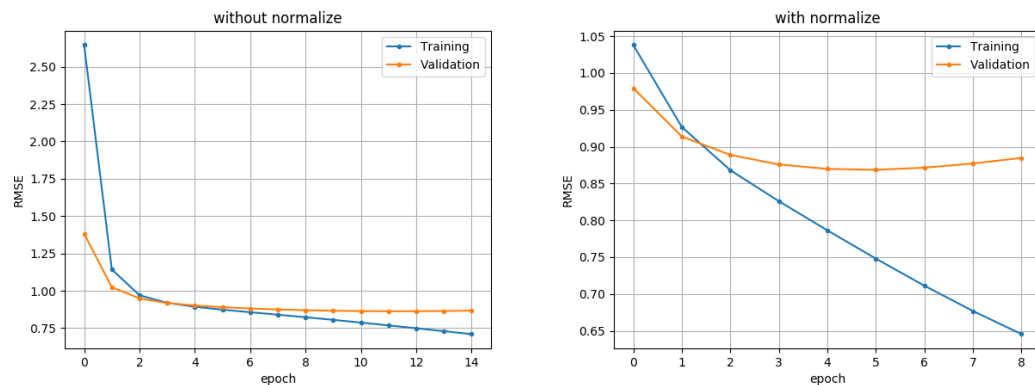
$$train^* = \frac{train - mean(train)}{std(train)}$$

$$pred = pred^* \times std(train) + mean(train)$$

下表則是有無 normalize 的比較，model 皆是使用 128 維有加 bias 的 MF:

NORMALIZATION	RMSE
NO	0.86564
YES	0.87041

訓練過程如下兩圖所示，左圖為無 normalize，右圖則是有 normalize:

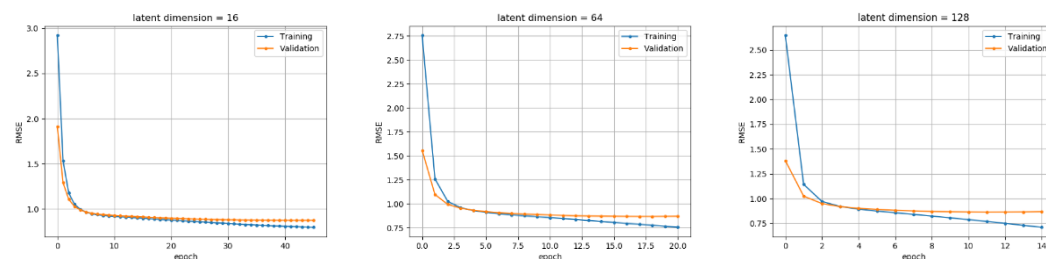


以 latent dimension 為 128 維的結果來說，有 normalize 的 model 其 RMSE 反而較大一些且 overfitting 的情形非常嚴重，但其訓練過程相較無 normalize 的 model 變動較小也收斂較快。

2. (1%)比較不同的 latent dimension 的結果。

(collaborator:盧玄真、施宣安、劉紹增)

本題我設定 latent dimension 為 16、64 及 128 皆加上 bias 且資料沒有經過 normalize，下三圖比較其訓練過程:



結果則如下表所示:

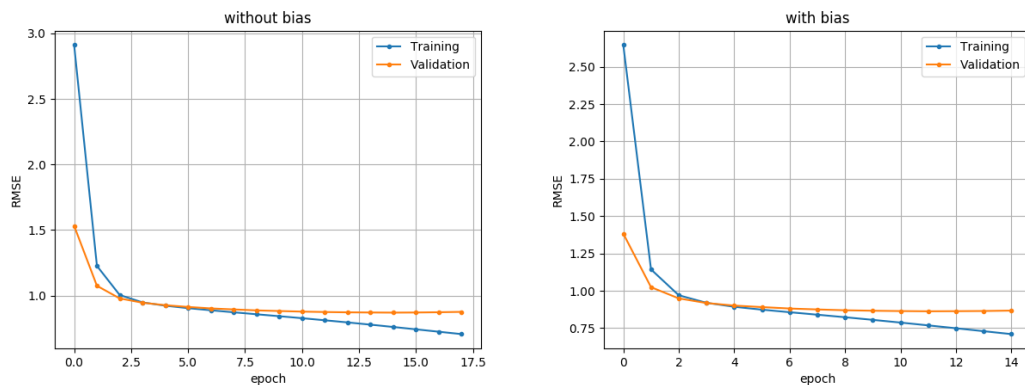
LATENT DIMENSION	RMSE
16	0.87461
64	0.86719
128	0.86564

由訓練過程及結果可知 latent dimension 為 128 維的結果最好且最快收斂。

3. (1%)比較有無 bias 的結果。

(collaborator:盧玄真、施宣安、劉紹增)

本題 model 皆是使用 128 維的 latent dimension 且資料沒有經過 normalize，有無 bias 的訓練過程如下兩圖：



下表則是其結果:

BIAS	RMSE
NO	0.86940
YES	0.86564

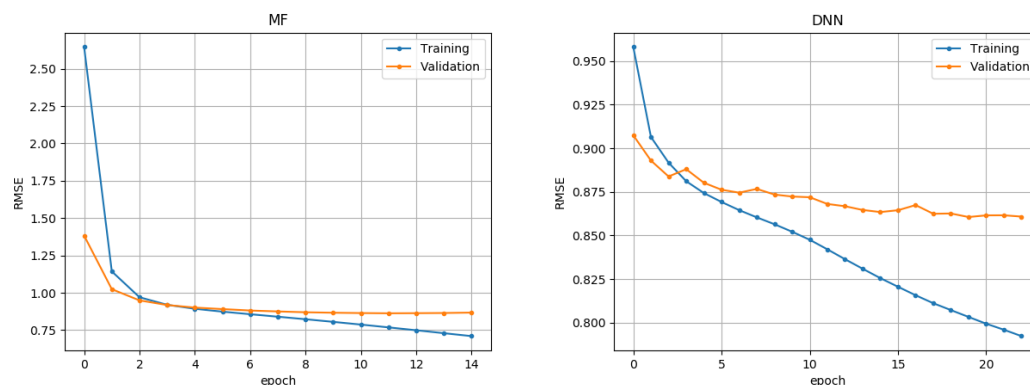
以此次測試的結果來說，我認為對沒有經過 normalize 的資料其訓練 model 中加入 bias 其 RMSE 表現會稍好且有加 bias 的 model 其訓練過程變動較小且收斂稍快一些。

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:盧玄真、施宣安、劉紹增)

這題我將 MF 簡單改成 DNN，先將連接 users 跟 items 的 Embedding 的方法從 Dot 改成 Concatence，再接上一個 128 維的 Dense 並且 activation function 使用 relu 先做分類，最後 Output Layer 是一維的 Dense 且 activation function 為 linear，loss function 則是 mse，最後只要將 loss 開根號就能得到 RMSE。

下兩圖為 latent dimension 為 128 維且訓練資料都沒有經過 normalize 的 MF 及 DNN 訓練過程比較:



下表為兩者結果的比較:

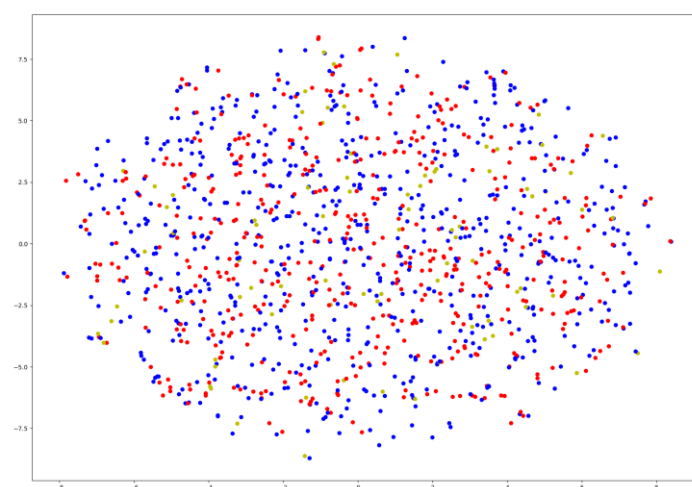
MODEL	RMSE
MF	0.86564
DNN	0.86051

由結果可知 DNN 經過調整表現可比 MF 好一些，但是也是會有較嚴重的 overfitting 問題。

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

(collaborator:盧玄真、施宣安、劉紹增)

COLOR	MOVIE CATEGORY
黃色	Fantasy、Mystery、Sci-Fi
紅色	Crime、Horror、Thriller
藍色	Action、Adventure



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:施宣安、劉紹增)