學號：R06946003 系級：資料科學碩一 姓名：湯忠憲

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

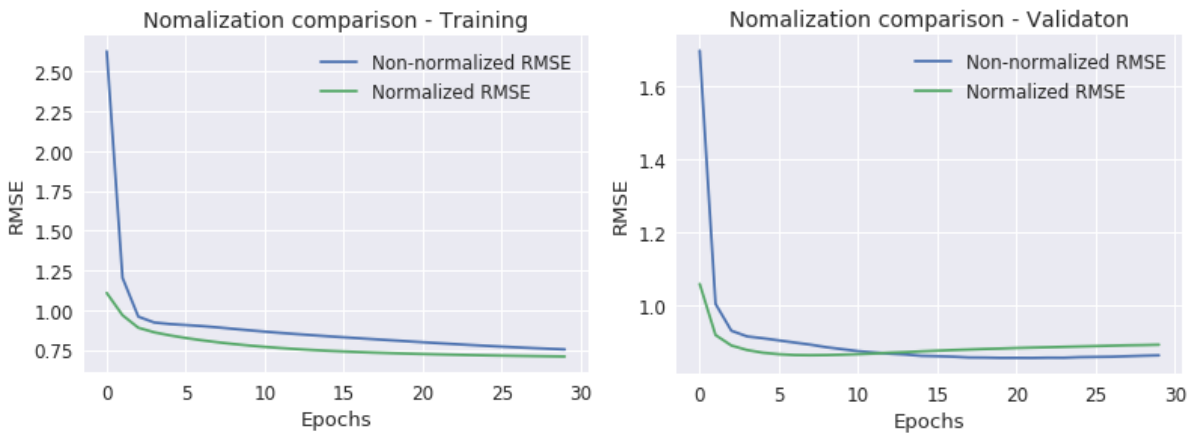
我使用的 normalization 是對所有的 rating 做 standardization，如下列式子:

$$\frac{Ratings\ -mean\ of\ all\ ratings}{standard\ deviation\ of\ all\ ratings}$$

計算後可得σ=1.116897661; μ=3.58171208。透過這樣的轉換可以將 rating 映射到[-1,1]，再將標準化過後的 rating 給進 model。在做 evaluation 或者 prediction 時再把標準差跟平均值分別乘回去和加回去。給定模型參數如下:

latent dim = 20
batch size = 1024
validation split = 0.05
epochs = 30

比較 MF 模型在有無標準化情況下的準確率。訓練過程如圖一:

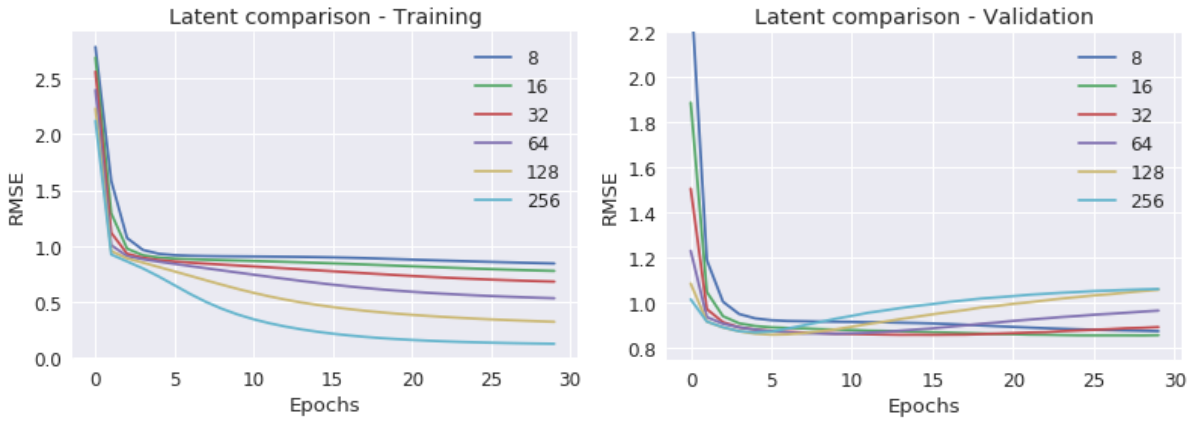


圖一、Normalization 比較圖

在 training 過程中(左圖)，可以看到 Normalize 後的 MF 可以有較低的 RMSE，收斂較快。在 validation 上(右圖)，Normalize 後的 MF 也是在很少的 epoch 數就能有不錯的表現，相對的也比較早 overfitting。而沒有 Normalize 的 MF 在多跑幾個 epoch 也能有差不多的表現。*Kaggle average performance: Non-normalized-0.85471, normalized- 0.84986*

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

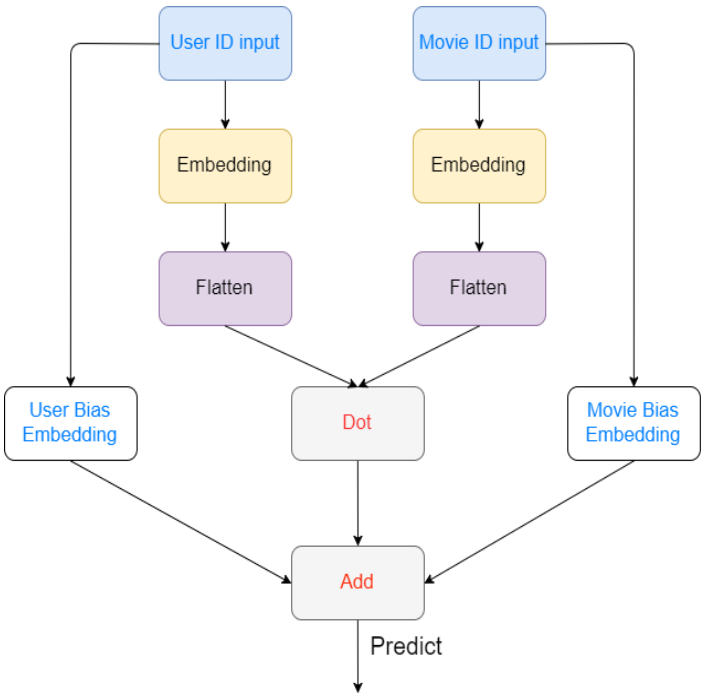
這部分我比較的 latent dimension 有 8, 16, 32, 64, 128, 256。從圖二可以發現在 training 過程中，越大的 latent dimension 可以擬合到更低的 RMSE。不過從 validation RMSE 中就可以看到，latent dimension 越大，越容易過擬合。在實驗組合中可以看到 16 跟 8 表現比較穩定。



圖二、Latent dimension 比較圖

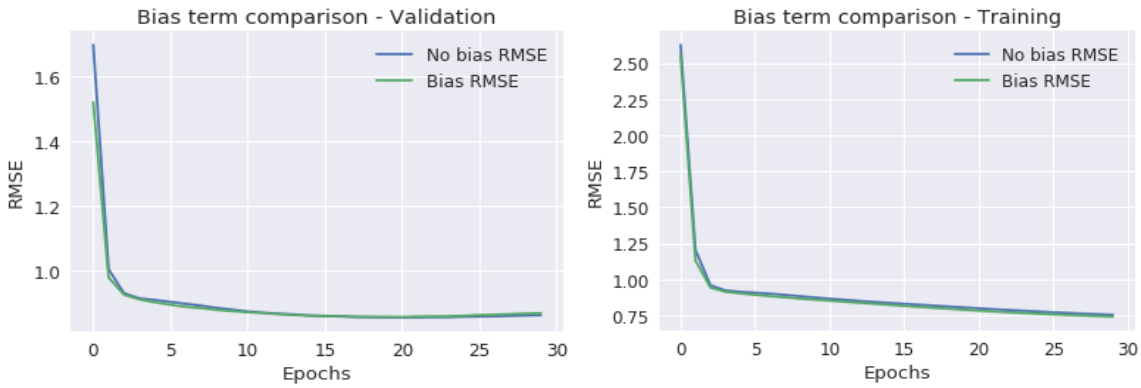
3. (1%)比較有無 **bias** 的結果。

這題我加入 User Bias 和 Movie 的 Bias，這兩項可以考慮到每個 user 評分的習慣和每個 movie 是否容易拿到較高分等因素。例如有些 user 習慣將分數打很高或者有的影片就是很熱門，易獲得較高的分數。MF 經修改後架構如下:



圖三、MF with Bias 架構

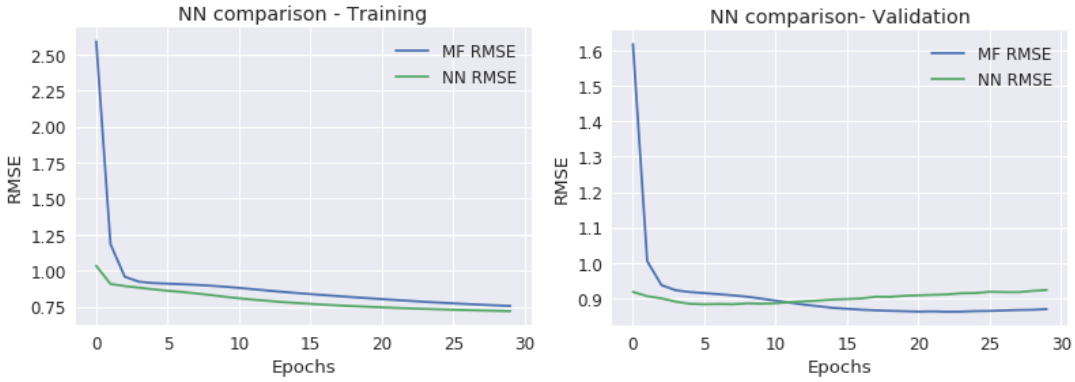
訓練過程如，可以看到其實有加 bias 跟沒加 bias 在 training 或 validation 過程中沒有很大的差別。但是雖然沒有顯著進步，多次實驗下來可以發現有加 bias 都會有較好的 validation RMSE。所以 Kaggle 上的 model 都是採用有加 bias 的。*Kaggle average performance: Without bias-0.84971, With bias- 0.84686*



圖四、MF Bias 比較圖

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 模型則是將 MF 中的 dot 取代，直接將 user vector 和 movie vector 接起來。之後再過兩層 fully-connected layer，分別是 150 和 50 維。訓練過程如圖五:

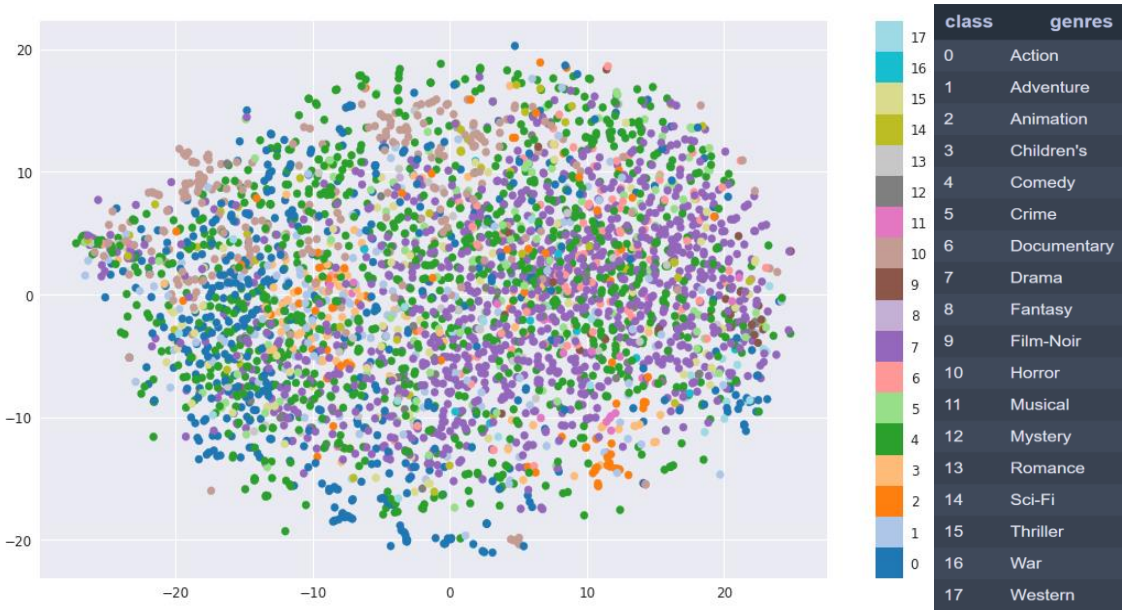


圖五、NN 比較圖

可以看到在 training 過程中，NN 模型的 RMSE 均比 MF 低，收斂速度較快，擬合資料的能力較好。然而在 validation 上，NN 很快地達到最低點，之後 RMSE 就往上跑，呈現 overfitting 的趨勢。MF 模型則是穩定下降，並且獲得了更低的 RMSE。若是將 DNN 模型加點 dropout 或者 fine-tune 參數，應該可以避免 overfitting。但最後效果沒有比 MF 好。

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

從 movie.csv 中可以提取電影的類型，有的電影有可能會屬於兩種一樣的類型，因此我從中隨機挑一種。將 movie 的 embedding 用 tsne 降維後投影在二維空間上繪圖，結果如下圖:

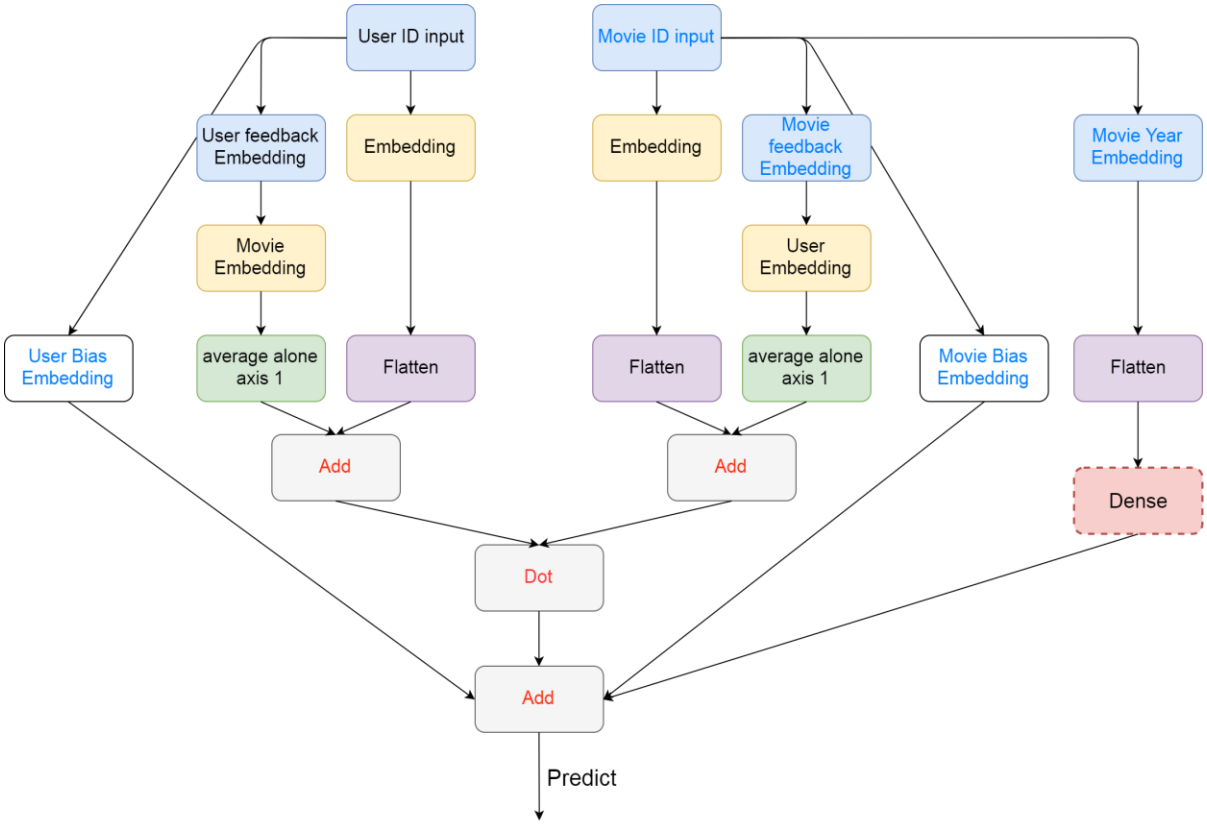


圖六、movie embedding

從上圖可以看到藍色(Action)跟紫色(Drama)有較明顯的區隔，可以直覺的理解為兩種差異較大的電影類型於二維空間的表現。而綠色(Comedy)則幾乎散佈在整張圖案，應該是這種類型的電影元素可以包含在其他不同電影類型中。

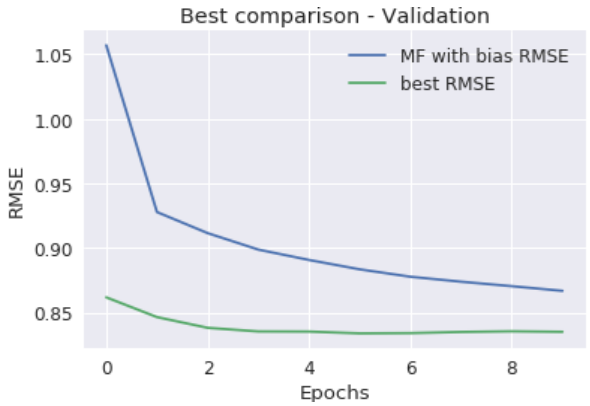
6. **(BONUS)(1%)**試著使用除了 **rating** 以外的 **feature,** 並說明你的作法和結果，結果好壞不會影響評分。

MF 模型除了 rating feature 和 bias，還可以考慮每個 user 評價過哪些電影，以及每個電影分別被哪些 user 評價過，還有電影的年份。處理上，可以將上述三種額外的特徵做成 embedding matrix，當 user ID 或 movie ID 進來時可以直接索引，找到該 ID 對應到的特徵向量。整體架構如下圖，其中一些 dropout layer 就沒有特別畫出來(主要是接在 Flatten 或 Embedding layer 後面):



圖七、Bonus model 架構

與加了 bias 的 MF 比較如圖八，在 training 時，觀查 validation RMSE 就可以發現收斂的更快，RMSE 也低許多。最終經過 20 個相同架構的模型 ensemble 後，在 Kaggle 上的 RMSE 約為 0.826/0.824。



圖九、Bonus model 比較