學號：B04901060 系級： 電機三 姓名：黃文璁

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

　　Normalize的部分使用 ，其中 為 training data的rating平均、  
 為training data的rating標準差，predict 的時候再將model的輸出乘上 、  
加上 即為rating的預測值。  
　　以下比較有無normalize的validation loss (RMSE)：

　　（有normalize的RMSE在此數據中已經乘回 ）

Latent dimension 皆為64的情況下：

**有Normalize** 0.8690 (15個epoch)

**沒有Normalize** 0.9243 (31個epoch)

由此結果可以看出：

　　　　1. 有normalize會更快收斂

　　　　2. 有normalize的RMSE loss更小

　　故整體來說normalize是比較好的。

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

　　以下結果皆有對rating進行normalize，比較8~256的latent dimension：

**Latent dimension loss 　　 val\_loss epochs**

**16** 0.8019 　　　0.8732 32

**32** 0.7833 　　　0.8734 22

**64** 0.7742 0.8659 14

**128** 0.7264 0.8672 11

**256** 0.6859 0.8681 8

　　可以注意到latent dimension越高training loss越低，但validation loss則沒有明顯下降，也就是latent dimension太高的情況下更容易overfit。整體而言latent dimension大約選取64即可有不錯的效果。

　　此外也可以注意到latent dimension越高的情況下，只需要更少個epoch即可達到最低的val\_loss，也就是training可能會稍微更快一些。

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

　　加上bias的方法為使用輸出為一維的Embedding layer。

　　以下比較有無normalize的validation loss (RMSE)：

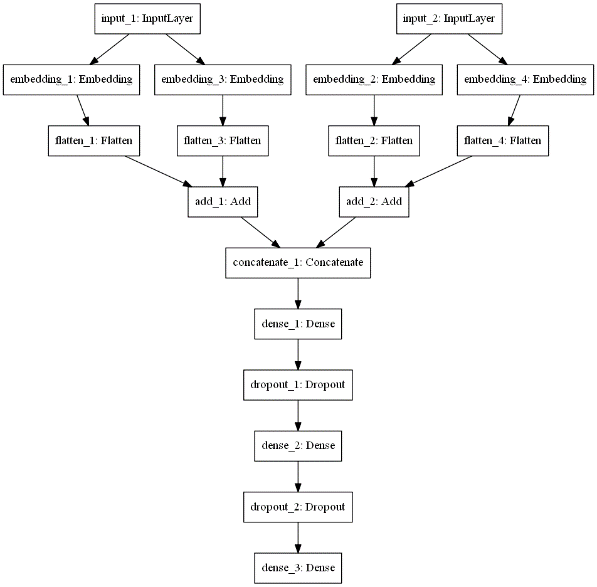
　　取latent dimension為64進行本題實驗。

**有bias** 0.8668 (15個epoch)

**沒有bias** 0.8668 (14個epoch)

　　由上列兩個結果來看，在本題中有無bias的差異並不是很大。

1. **(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)**

　　本次利用DNN的方法為：將MF中的Dot layer部分改為Concatenate後，再經過數層Dense layer，最後輸出一個實數代表rating。

　　架構如右圖，其中前兩層Dense皆為512個units，最後一層為1個。Latent dimension和ML相同皆為64兩種model的loss和val\_loss比較如下：

loss val\_loss

　MF 0.7742 0.8659 (14 epochs)

　NN 0.8052 0.8663 (7 epochs)

經過NN作法後得到的結果和MF差不多，但可以觀察出NN的model更快得到最低的loss，這應該是由於NN參數較多。

1. **(1%)請試著將movie的embedding用tsne降維後，將movie category當作label來作圖。(collaborator: 無)**
2. **(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)**