學號：R05944043 系級： 網媒碩一 姓名：宋焱檳

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

Normalize 方法： $Data_{norm} = \frac{Data - mean(Data)}{Std(Data)}$

還原的方法： $\Pr ed_{test} = pred^{*}_{test} * Std(Data) + mean(Data)$

Normalize	Pubilic loss	Valid loss	Public RMSE	Valid RMSE	Public score
有	0.5027	0.5857	0.7053	0.76151	0.84922
無	0.6853	0.6519	0.8237	0.8035	0.85811

Validation data = 10% * training data

在訓練過程中，經過 normalize 的數據收斂的速度比較快，loss 和 RMSE 也一開始就相對比較小。觀察上表，Normalize 有對於 loss 和 RMSE 的減少效果，十分地明顯，因為我們把原來的相對距離縮小到 1 了。所以就這一點無法說明 normalize 的好壞。但是可以確定的是 normalize 對於 model 有積極的作用。觀察在 public score 上的表現好壞，也確定了之前的假設，normalize 提高了 0.9%的 RMSE 的 score。在最後結果的處理上注意，分數是 1-5，所以小於 1 大於 5 都應該縮小到這個範圍的邊界。

(1%)比較不同的 latent dimension 的結果。

在都有 normalize、以及結合 DNN 的情況下，測試使用不同的 latent dimension 結果如下：

latent dimension	Pubilic loss	Valid loss	Public RMSE	Valid RMSE	Public score
100	0.5543	0.5866	0.7253	0.76995	0.85874
120	0.5027	0.5857	0.7053	0.76151	0.84922
140	0.5786	0.5929	0.7588	0.76801	0.86060

根據多次測試的經驗，發現，latent dimension 在 100-150 之間的時候 validation 和 score 有比較好的結果，所以以 20 為 step，測試 model。可以看出，在 latent dimension 為 120 的時候 Vlidation 的 RMSE 會比較小，score 也比較好。整體來看，validation 上的 loss 和 RMSE 與 latent dimension 關係不是很大，因為都比較相似。latent dimension 比較影響在 public set 上的結果，100 維的時候會比 140 維的結果好。

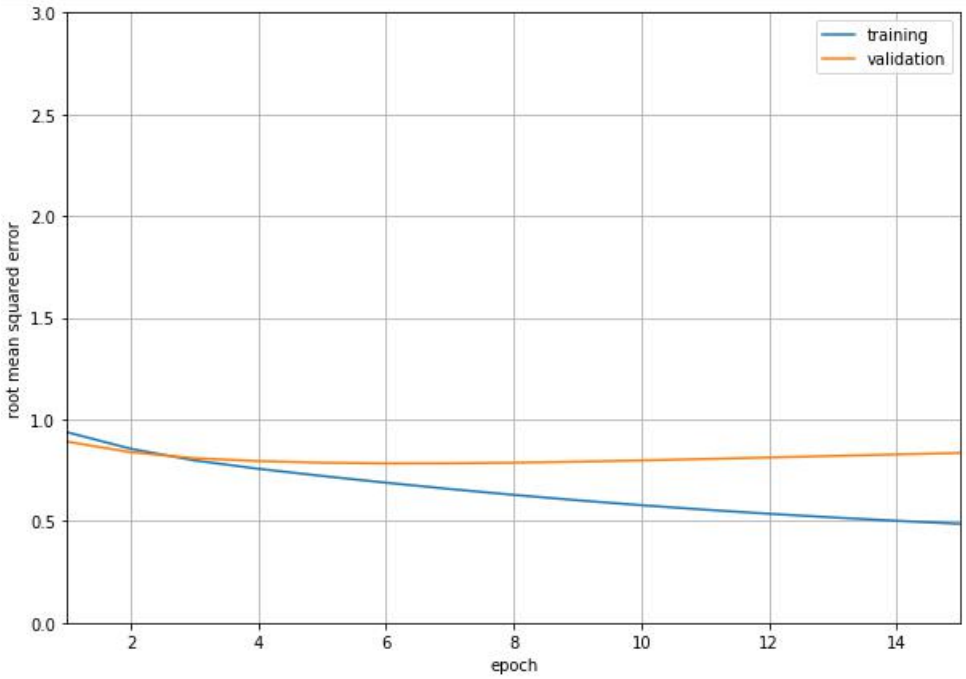
(1%)比較有無 bias 的結果。

根據公式： $r_{i,j} = U_i \cdot V_j + b_i^{user} + b_j^{movie}$

對 MF 增加 bias，結果如下：

	Pubilic loss	Valid loss	Public RMSE	Valid RMSE	Public score
無 bias	0.5683	0.5683	0.7515	0.7916	0.86577
Item_bias+User_bias	0.4739	0.6846	0.6139	0.77991	0.87798

Training 過程中 RMSE 的變化圖如下：



查看上表，可以看出，在 MF 中加入 bias 之後，理論上可以使得在 public 上得到比較好的結果，不過 validation 的 loss 變大了，validation 的 RMSE 雖然小了，在 public score 上的得分變差。可能是因為 Loss 變大後，正好 random 的值會偏離這組 test data 的值；也可能是需要結合 private data 從而確定具體好壞。

查看上圖，是在加了 bias 後的 RMSE 變化圖。可以看出，3epoch 之後，其實就有點 overfitting 了,因為 validation 上的 RMSE 已經開始回升了。然後因為 earlystop 的原因，所以很快就 stop 了。雖然，training 上的 RMSE 可以很小，但是 validation 上的 RMSE 在 0.78 的時候就差不多了。也有一部分可能是因為 model 的 parameters 沒調到最佳。

(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。
我使用的 DNN 架構圖如下：

Layer (type)	Output Shape	Param #
merge_8 (Merge)	(None, 240)	0
dropout_15 (Dropout)	(None, 240)	0
dense_15 (Dense)	(None, 120)	28920
dropout_16 (Dropout)	(None, 120)	0
dense_16 (Dense)	(None, 1)	121

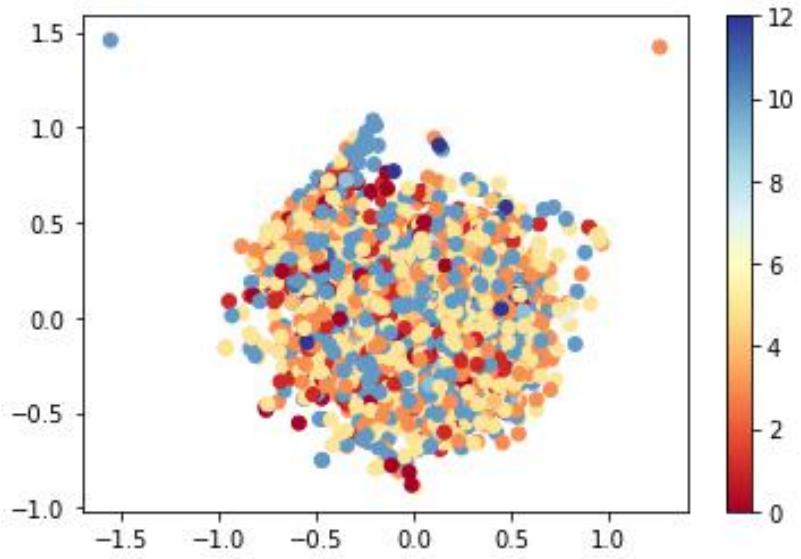
Total params: 1,228,081
Trainable params: 1,228,081

由一個 latent dimension 長度、Activation 為 relu 的 dense 層，和一個長度為 1、Activation 為 linear 的 dense 層做緩衝和預測。中間 dropout 率為 0.25

	Pubilic loss	Valid loss	Public RMSE	Valid RMSE	Public score
DNN	0.5027	0.5857	0.7053	0.76151	0.84922
MF	0.5683	0.5683	0.7515	0.78160	0.86577

對比 DNN 和 MF，我的最優解是在 DNN 得到的。我是將 user embedding 以及 movie embedding concatenate 在一起再過 DNN 得出 rating。DNN 訓練的 model 比較不容易 overfitting，而且，也可以使 Public loss 更小，score 上可以說明 DNN 的 model 會比 MF 的更優。不過，MF 的優勢是可以把 validation 上的 loss 減的比較小，variance 就比較小，model 比較穩定。

(1%)請試著將 movie 的 embedding 用 tsne 降維後，將 movie category 當作 label 來作圖。
我是把 movie 的 tag 經過分類，然後經過 TSNE 後，繪製出來：



具體做法如下：

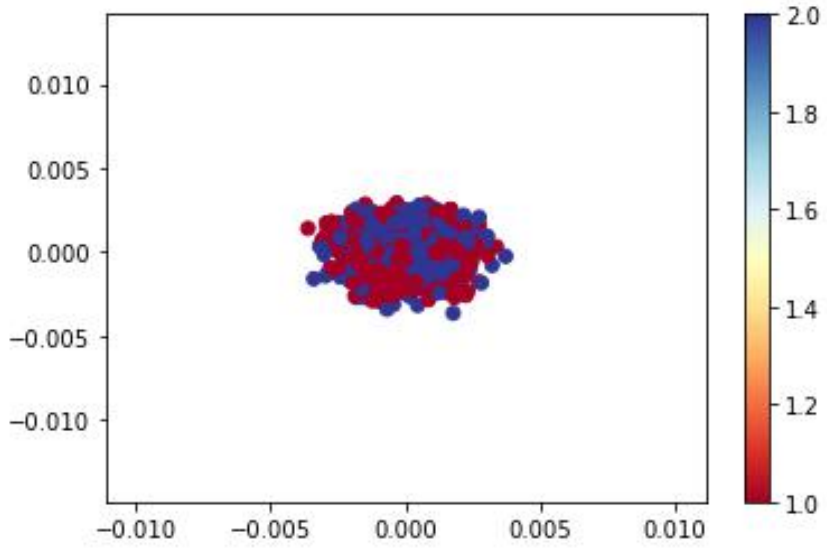
tag1 = ['Thriller','Horror','Crime']

tag2 = ['Drama','Musical']

tag3 = ['Adventure',"Children's"]

tag4 = ['Comedy', 'Romance']

經過觀察，發現這幾個 tag 會常常共同出現，所以我把這四類標記為同一個 label，然後其他就是以 tag 中先出現的那個類別作為這個 movie 的 tag。最後一共分出了 13 個 tag，然後經過 tsne 降維然後 plot 出來。由於數據比較多，所以只能大致觀察出一個輪廓。每種類型的 movie 分佈的。大致看出，第十類（tag1）和第四類（tag4）分別有一個在兩端的點距離比較遠。因為可能選取了所有的 tag 的情況，所以交雜在一塊。所以，我用 tag1 和 tag2 又繪製了一個圖：

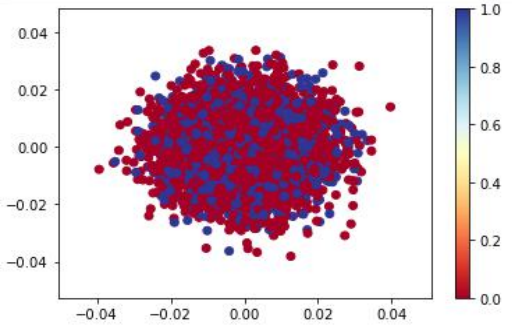


經過多次對比結果，我嘗試先用 PCA 降維到 40 維，這樣可以看出紅色的 tag1 和藍色的 tag2 分的相對比較清楚一點。紅色分佈比較下方，tag1 和 tag2 的組間距離不是太大。也有可能是因為數據量太大，所以重疊起來後有點密集，不是十分方便觀察。

(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。

觀察 user.csv 中 Female 和 male 的數量比：1：3，結合 sex 的 Tag 的 TSNE 分佈圖（藍色是'F'，紅色是'M'），所以希望使用這個 feature 對於結果進行修正，使得 female 的評分更有說服力，和 male 近似。所以，我嘗試讓 Female 的評分做 10%-（-10%）範圍的波動，並且觀察結果，以下是對於不同程度 Female 在 test 結果上評分修正對於 score 評分的影響情況：

Female	Public score
+0.10*female	0.85564
+0.05*female	0.84936
+0.01*female	0.84975
1.00*female	0.85012
-0.01*female	0.84909
-0.05*female	0.85258
-0.10*female	0.86584



經過上圖在 public score 上得分情況的分析，可以看出，在 female 的評分變化為 99%female 原來分數的時候，在 public score 上有 0.1%的提升。