學號：B04611041 系級： 工海三 姓名：簡暐晉

1. (1%)請比較有無 normalize(rating)的差別。並說明如何 normalize. (collaborator:No)

   我用 MF 都有加 bias 的 model 來做比較，將 training data 減掉標準差後 除上平均(train = (train - std)/mean, 再將他的 std 和 mean 存下來後, testing 時再還原(test = test* mean + std)可以發現有 normalize 的結果會比較好一點。

Dim	Normalize(kaggle 成績)	無 Normalize(kaggle 成績)
25	0.85759	0.85887
50	0.85289	0.85408
100	0.85451	0.85712
200	0.85899	0.85903

2. (1%)比較不同的 latent dimension 的結果。(collaborator: No)

   我使用 mf 的 model, 在同樣都有做 normalization 與加 bias 的情況下比較 由表格可看出 dim 取 50 結果為最好，再往下降或往上加正確率都會變差。

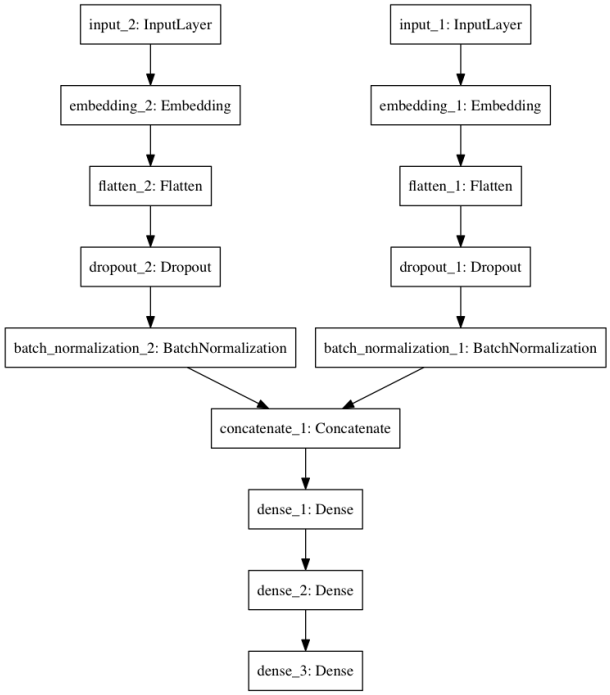
dim	Kaggle score (public + private)/2
25	0.85759
50	0.85289
100	0.85451
200	0.85899

3. (1%)比較有無 bias 的結果。(collaborator: No)

dim	有 bias	無 bias
25	0.85759	0.86581
50	0.85289	0.85529
100	0.85451	0.85443
200	0.85899	0.8600

   我對 4 個不同的 latent dimension 在都做 normalization 的情況下分別比較了有無 bias 的差別，發現在大部分的情況有加 bias 的結果會比沒加的略好一點，可推斷每個 movie 和 user 存在著個體差異，而 bias 可稍微消除這些影響。

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: No)

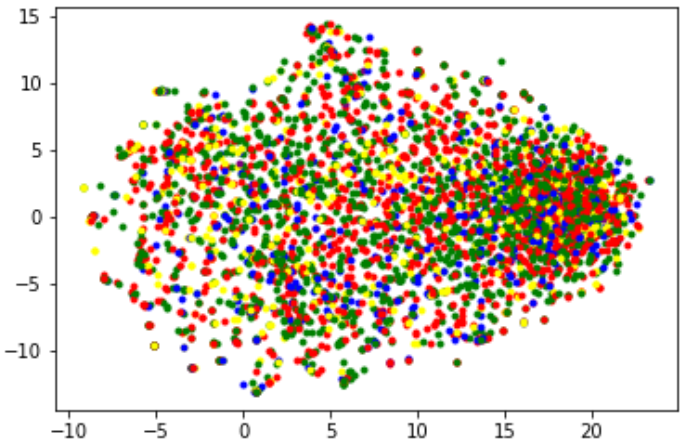


我 DNN 的模型和 MF 一樣，user 和 movie 都先經過兩個 latent dimention 為 50 的 embedding layer，但之後不用 dot，而是將兩個 embedding layer concatenate 起來，最後經過 3 層 dense layer，units 分別為 128,64，最後 output 1 當作答案，activation function 皆使用 selu，並且有做 normalization，最後在 kaggle 上的成績為 0.8695，

而 MF 最好的成績為 0.85289，由此可知 DNN 的結果比 MF 差了一點。

5.(1%)請試著將 movie 的 embedding 用 tsne 降維後，將 movie category 當作 label 來作圖。(collaborator: No)

我將電影分成 4 大類來作圖，可以看到效果並沒有很好，有可能是訓練的時候沒訓練好，或是我分的類別其實彼此之間還是有關連，因此沒有分的很開。



Color	Movie Type
red	['Action' 'Adventure' 'Animation' "Children's"]
green	['Crime' 'Documentary' 'Mystery' 'Sci-Fi']

blue	['Comedy' 'Drama' 'Musical' 'Romance' 'Western']
yellow	['Fantasy' 'Film-Noir' 'Horror' 'War']
purple	other

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: No)

我使用 user 的 gender, age, Occupation 作為額外的 feature, 由於種類都不多，因此我將他們都做 one-hot encoding，可得到 30 維的 feature。movie 則將 **Genres** 做 one-hot encoding，再取他的年份得到 19 維的 feature，把他們，之後把這兩個 feature 各經過一層 50 units 的 NN 再 concatenate 起來，之後接到和第 4 題的架構一樣 model，經過 3 層 dense layer 後得到 output，在 kaggle 上的成績為 0.85075，比用 MF 的最佳 model 還要進步一點。