

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

- `normalize` 的方法

我總共測試兩種 `normalize` 的方法，第一種是把所有的 `rating` 直接 `normalize`，第二種是每一個 `UserID` 都各自 `normalize` 各自的 `rating`。

- 實驗結果

我使用相同的 `model`，並切 `validation set`，而且 `save best` 來測試三種不同 `normalize` 的方法。實驗結果如下表：

	no normalize	normalize on all rating	normalize for each user
validation RMSE	0.84909(勝)	0.86001	0.86832

從實驗結果可以發現：沒有加上 `normalize` 的結果是三者中最好的。

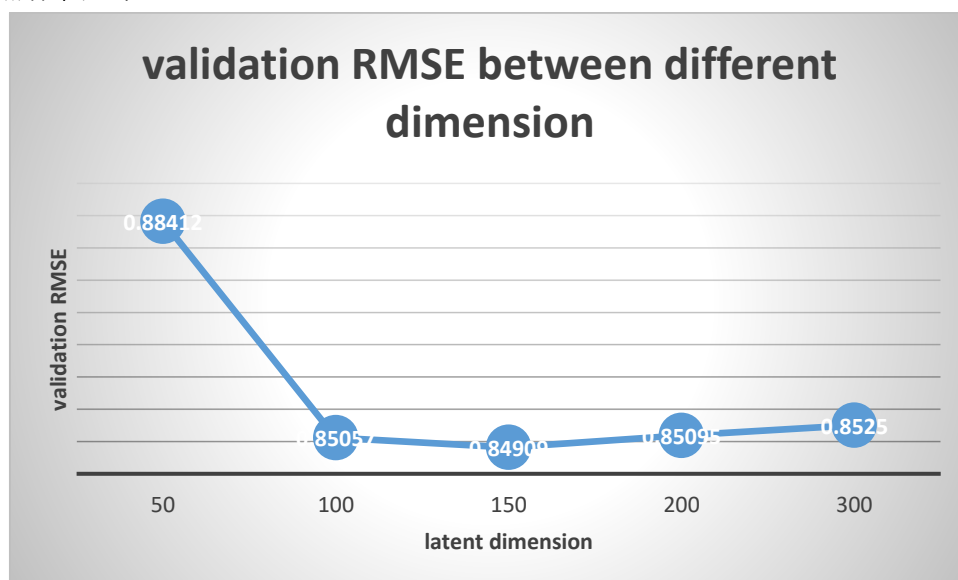
2. (1%)比較不同的 `latent dimension` 的結果。

為了方便比較，我讓 `user` 和 `movie` 的 `embedding` 的 `latent dimension` 相同。

- 實驗結果

我都使用相同的 `model`，並切 `validation set`，而且 `save best` 來測試。

實驗結果如下圖：



從圖片中可以發現：當 `latent dimension` 降到 150 的時候，`RMSE` 是最小的。

其中，當 dimension 小於 150 的時候，RMSE 會特別大；當 dimension 大於 150 之後，RMSE 只會稍微增加。

- 推測原因

會有這樣的結果的原因可能是：如果 latent dimension 太小 (<150)，無法完全代表原本資料隱含的成分，結果當然很差。相反的，當 latent dimension 夠大並且越來越大的時候，雖然 embedding 能夠代表所有資料隱含的成分了，但是因為 dimension 太大了，gradient descent 很容易會 overfit training set，而無法提升 validation set 的結果。

3. (1%)比較有無 bias 的結果。

我把 user 和 movie 都個別加上 bias，都使用相同的 model，並切 validation set，而且 save best 來測試。

實驗結果如下表所示：

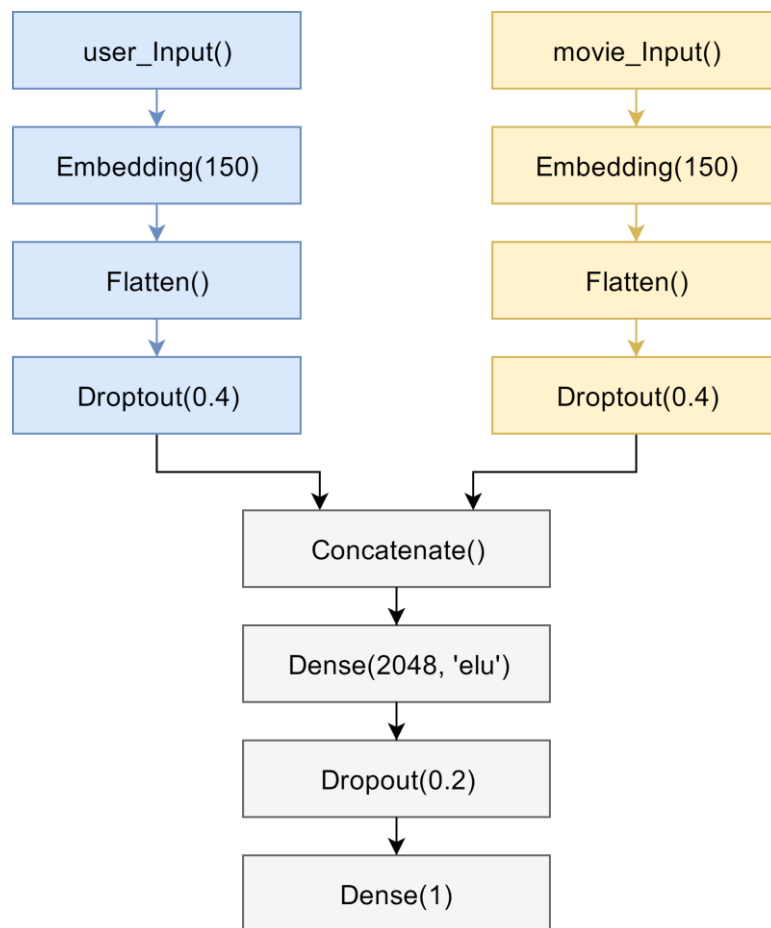
	no bias	add bias
validation RMSE	0.84909(勝)	0.85498

從實驗結果可以發現：沒有加上 bias 的結果是最好的。

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。

- 實做的方法：

我把 user 的 embedding 和 movie 的 embedding 接在一起，作為 DNN 的 input，並且設定 output 的結果直接是 predict 的 rating。完整的 model 如下圖：



- 結果及比較：

測試兩種 model 的時候，我都切 validation set，並且 save best 來測試。

實驗結果如下表：

	MF	DNN
validation RMSE	0.84909	0.84847(勝)

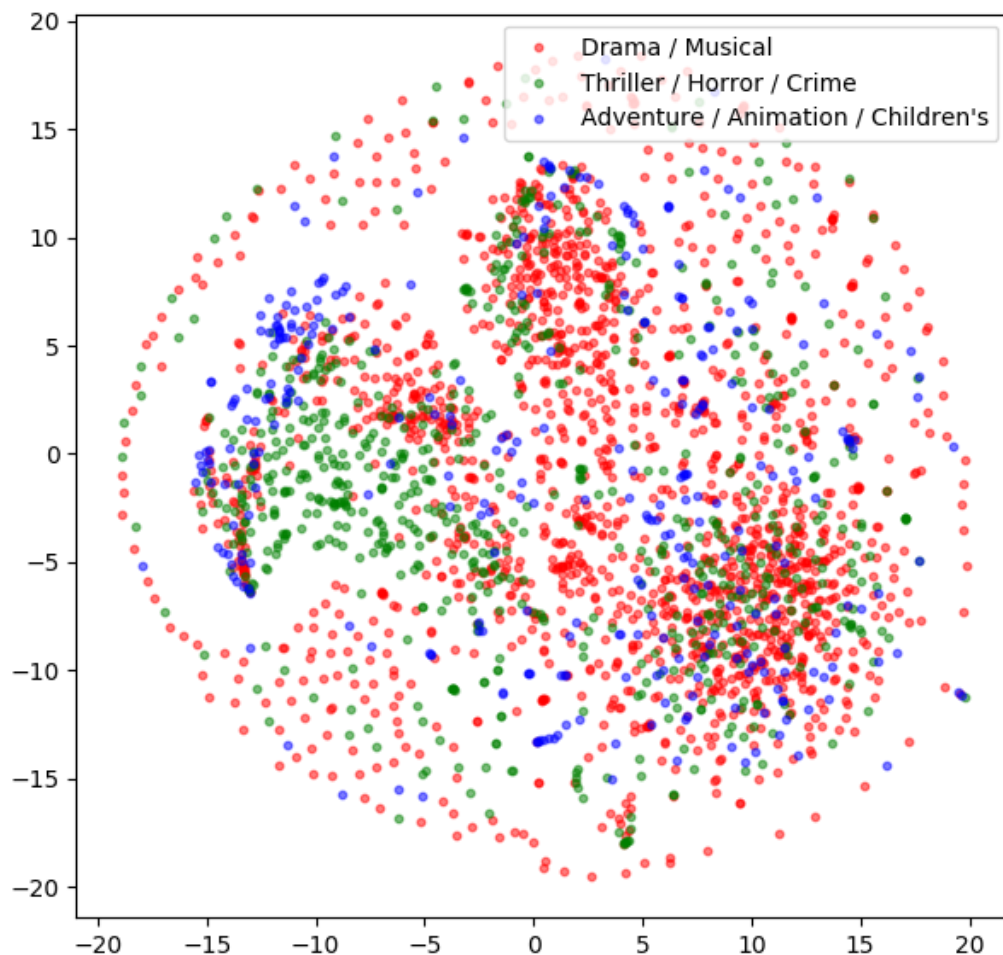
從表格中可以發現：使用 DNN 的結果是比較好的。

- 推測原因

會有這樣的結果的原因是因為：DNN 的 model 其實包含了 MF 的 model，所以 DNN 的結果至少會和 MF 一樣或更好。

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

我使用和助教的投影片一樣的分類，也就是把 Drama / Musical 作為一類，Thriller / Horror / Crime 作為一類，Adventure / Animation / Children's 作為一類。畫出來的結果如下圖：

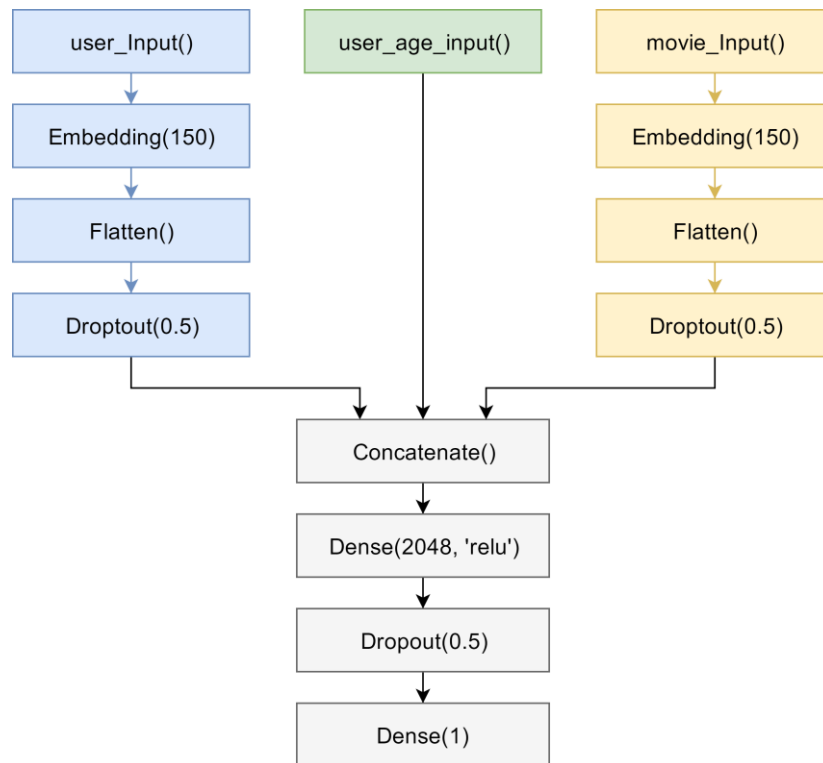


從圖中可以看出：有一部分的綠色、紅色、藍色的點會自己聚集在一起，而不會跟其他顏色混在一起。這代表我的 model 有一定能力可以區分不同類別。

但是三種顏色混在一起的情況也不少，這代表我的 model 還沒有辦法很好的處理這些情況下的資料。

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。

我只取 user 的 age，並 Concatenate 到 user 和 movie 的 embedding 上，再使用 DNN 來訓練出 model。model 的架構如下圖所示：



訓練出來的結果如下表：

	only training set	training set + user age
validation RMSE	0.84847	0.84547(勝)

從表格中可以發現，加入了 user 的 age 比只使用 training set 的資料的結果要好一點。