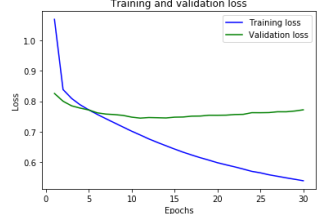
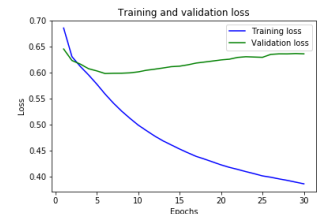


學號：B04505026 系級：工海三 姓名：蔡仲閔

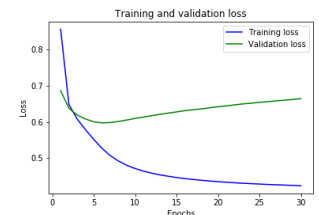
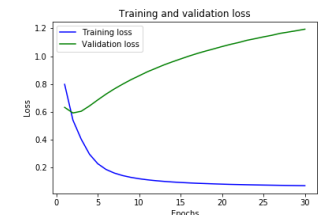
1. (1 %)請比較有無 normalize 的差別。並說明如何 normalize.

我這次是使用 $x = \frac{X - \mu}{\sigma}$ 的方式，將 rating 平移至-1~1 之間，在最後 predict 完之後再進行還原，最後我們觀察他們在 Private Score 上的表現，可以看到經過 normalize 的資料有相對較好的表現，而較明顯的差別則是在訓練次數上，原始資料大約在 10 個 epoch 後漸漸收斂，而經 normalize 的資料大約在 5 個 epoch 之後就收斂，但也很快出現 overfitting 的狀況。

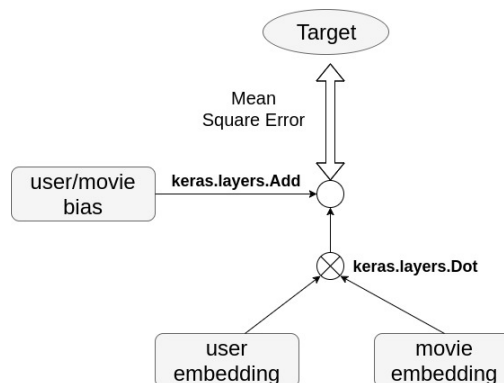
| | Without Normalize | Normalize |
|---------------|---|---|
| Private Score | 0.86388 | <u>0.86287</u> |
| Loss |  |  |

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

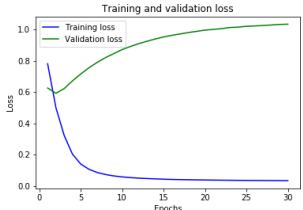
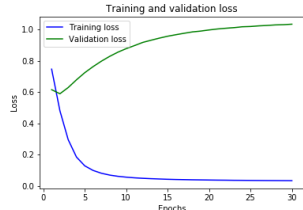
在這個實驗中我們調整不同的 embedding dimension 並觀察他們在 Private Score 上的表現以及 traing 過程的變化，從下表我們可以看到 dimension 的增加可以有效降低 RMSE，但訓練後期則出現明顯的 overfitting 的問題，這與 DNN 參數量增加的問題類似，而我也嘗試做了 196 的實驗，其表現並沒辦法再提升。

| Dimension | 16 | 128 |
|---------------|---|---|
| Private Score | 0.86432 | <u>0.85961</u> |
| Loss |  |  |

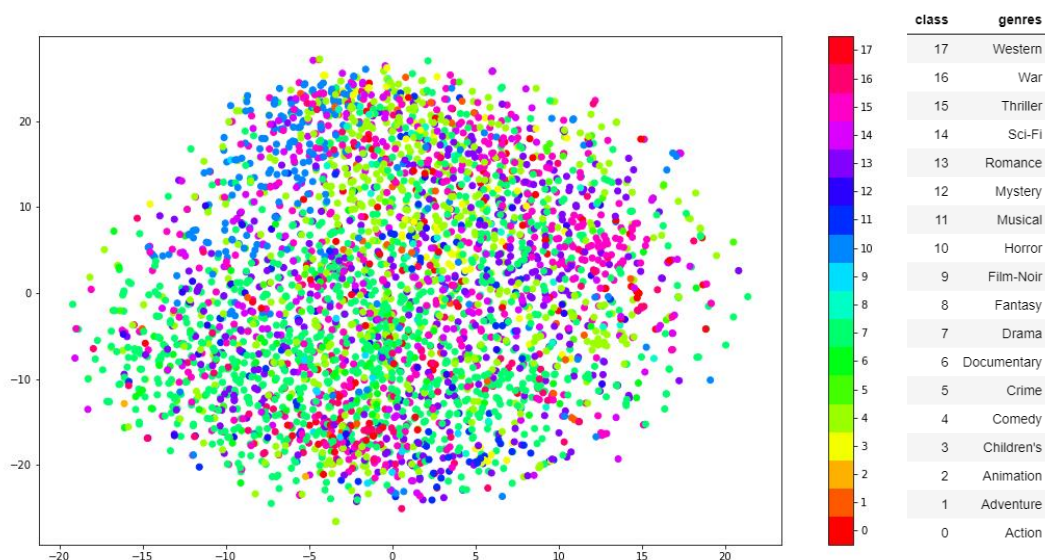
3. (1 %)比較有無 bias 的結果。



我參考投影片的架構設計出我的 model，在原先的 model 加入 bias 項，從下表可以看到加入 bias 確實可以小幅度的增進預測結果，而在訓練時間上也不會大副增加，是有效提升表現的方法。

| | Without Bias | With Bias |
|---------------|---|--|
| Private Score | 0.85980 | <u>0.85692</u> |
| Loss |  |  |

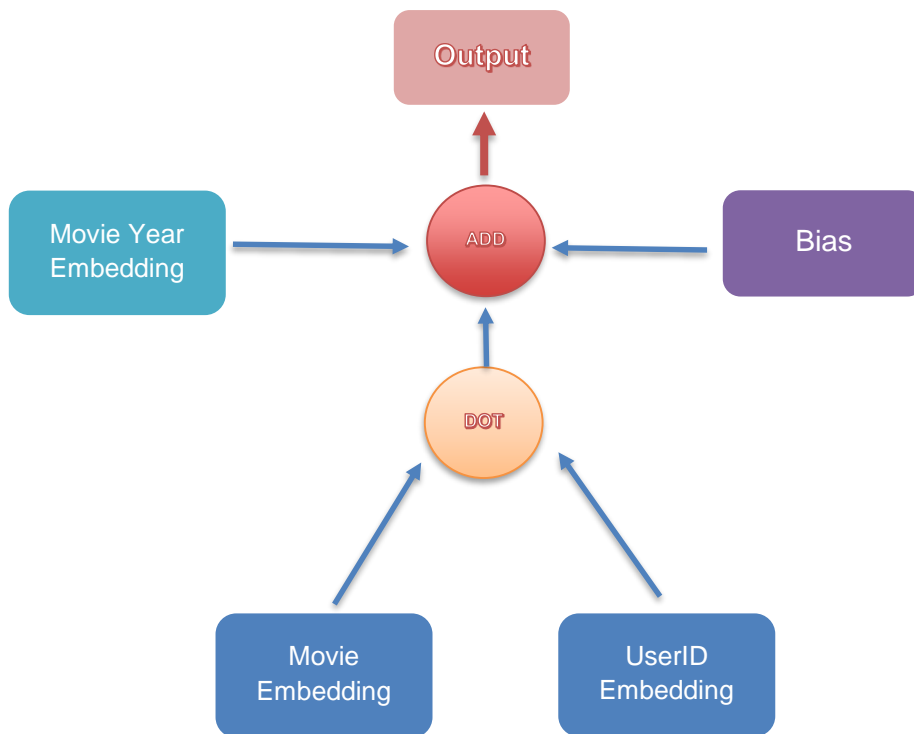
4. (1 %)請試著將 movie 的 embedding 用 tsne 降維後，將 movie category 當作 label 來作圖。

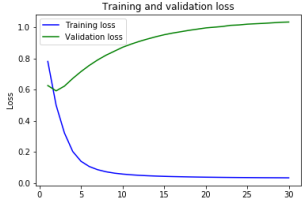
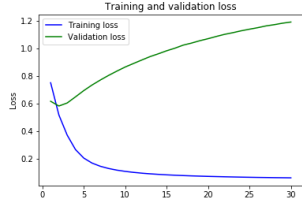


從上圖中可以看到由於種類較多，且一部電影分類多半不只一種，造成作圖結果並沒有太好。但其中我們仍能觀察到淺藍色，也就是 Horror 相當集中在左上角，而蘋果綠 Comedy 則是散佈在中右區塊與 Horror 較無重疊，這也是兩種差異較大的種類對於 User 造成對電影進行評分時可能造成的影響。

5. (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。

考慮到不同 User 對不同年份的電影可能會有不同的喜好，我試著在 Model 中 MovieID input 後，另外加入了 Movie Year Embedding，最後在跟 bias 項一起 Add 起來。而從下表中也可以看到，加入此項後，在 Private Score 上也可以得到較佳的表現。架構大致如下圖。



| | Original | With Movie Year Embedding |
|---------------|---|--|
| Private Score | 0.85621 | <u>0.85500</u> |
| Loss |  |  |

Reference: keras-movielens-cf/ bradleypallen
 (<https://github.com/bradleypallen/keras-movielens-cf>)