

程式部分有問去年的學長:簡上淵、洪建豪，其中加上 3.5817 是她們說去年作得很好的同學分享的內容。

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

有 normalize:0.8732

無 normalize:0.8703

其實並沒有很大的差別，而我這邊使用的方法是在 Dot 這一層的參數，把 normalize 打開，而打開了這個之後 layer 會做的事情是取 l2-normalize，相當於是取 cosine proximity between the two samples。

也有將 output 的 rank score normalize，

$$(x - \text{mean}) / \text{std}$$

不過結果在自己算的 val_rmse 並沒有提升，大概 0.89 左右。

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

這邊都是跑了 100 個 epoch，而這邊的 rmse 是用 keras 的 custom metrics 自定義的。

	Train_rmse	Val_rmse
8	0.84	0.984
16	0.80	0.996
32	0.74	1.08
64	0.70	1.21

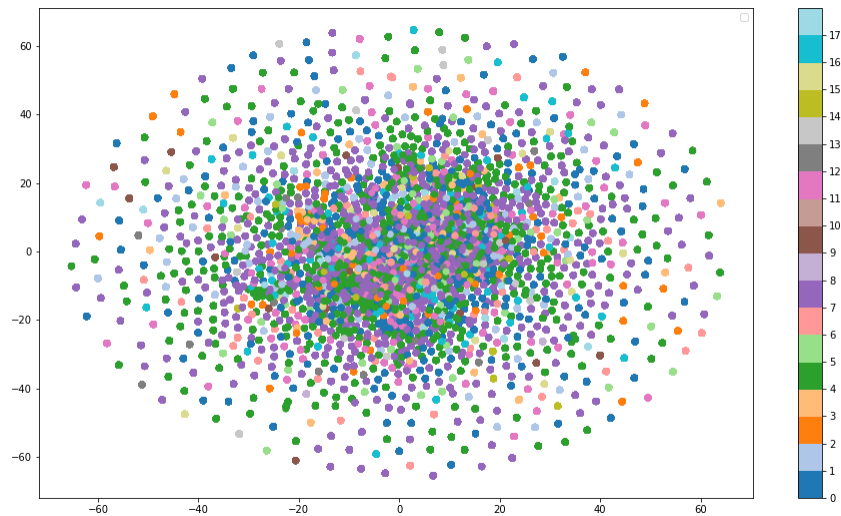
可以從上面數據觀察出來，當 latent dimension 越來越大的時候，其實 model 會開始 overfitting，這邊實驗的結果也符合上次 HW5 助教在 TA 時間分享的概念，如果 dimension 越大，matrix 越 sparse 之後，反而會降低模型的準確度的。

3. (1 %)比較有無 bias 的結果。

	Train_rmse	Val_rmse
Bias	0.8076	0.982
無 bias	1.19	1.31

把 model 改成只有 user embedding 跟 movie embedding flatten 之後 dot 再一起，與有把 user_bias 加進去的 model 做比較，這也是跑了 100 個 epoch 之後的結果，可以觀察到其實每個觀眾對於评分的習慣，的確是會影響到他對一部沒看過的電影的评分，其實也滿符合道理的，有些人平時給分數，就不會給到一，或是有些人就是特別喜歡給五分，所以 bias 這一條資訊是滿重要的。

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

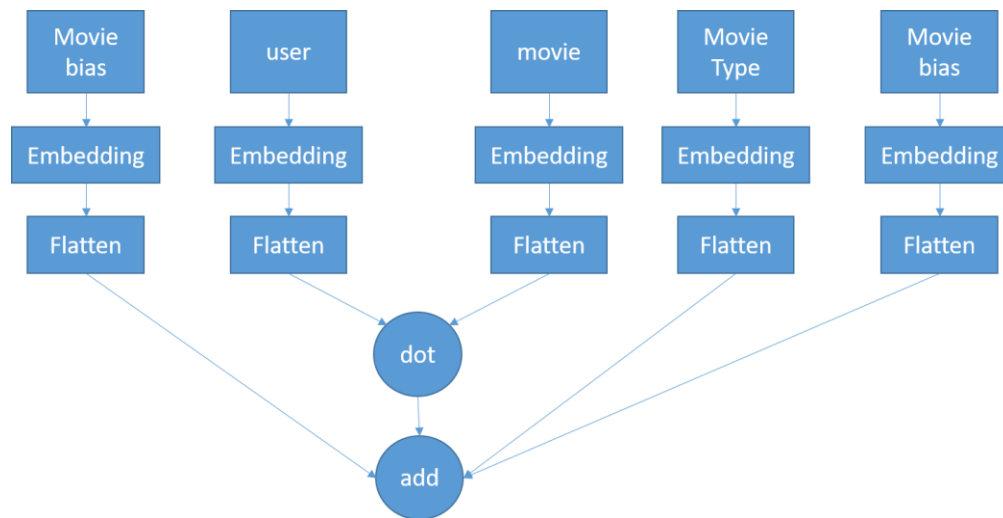


```
[0: 'Action',1: 'Adventure',2: 'Animation',3: "Children's",4: 'Comedy',5: 'Crime',6: 'Documentary',7: 'Drama',8: 'Fantasy',9: 'Film-Noir',10: 'Genres',11: 'Horror',12: 'Musical',13: 'Mystery', 14: 'Romance',15: 'Sci-Fi',16: 'Thriller',17: 'War',18: 'Western']
```

分出來的結果圖，大部分的都集中在中間，而有些會比較四散在各個區域，譬如黃色，這邊可能的原因是，很多電影可能都可以被歸類到這一類，像是劇情片這個類別，就幾乎所有的電影都可以被歸類，而可以比較看出來喜劇大部分都集中在中間，而 western(18)西部片的數量，可以看出來是比較少的。

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。

我在想可能影片的種類以及年分，也會對於最後的 rating 有所影響，於是將影片的年分以及種類與原本的 dot 結果 add 再一起，如下圖：



想說這樣加入了影片的種類應該可以增加準確度，不過結果好像準確度變低了不少，在 `val_rmse` 變高到 3.多，好像加入了之後結果變得比較不好，應該是這邊沒有把資料先處理好，或是這條資訊要原本的 `movies concatenate`，再一起 `dot` 之後可能結果會比較好。