ML2017FALL HW5 Report

學號：B04902090 系級： 資工三 姓名：施長元

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

   (collaborator: None)

   Normalize 方法： $Rating_{New} = \frac{Rating_{Old} - Mean}{Std}$

Training/Validation	Normalize 前	Normalize 後
Epoch 1	0.8798	0.8751
Epoch 2	0.8603	0.8598
Epoch 3	0.8503	0.8457
Epoch 4	0.8436	0.8421
Epoch 5	0.8394	0.8375
Kaggle Public	0.84301	0.84273

(RMSE)

Normalize 之後明顯收斂較快，但因為個人資源問題，這次作業都只有跑 10 個 epoch 左右，很難跑到真正收斂的 Epoch；以這種情況而言 Normalize 之後的結果是比較好的。

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

   (collaborator: 上學期學生 B04611015)

   因時間因素，都是 5 個 epoch 做比較，並且沒有做 Normalize

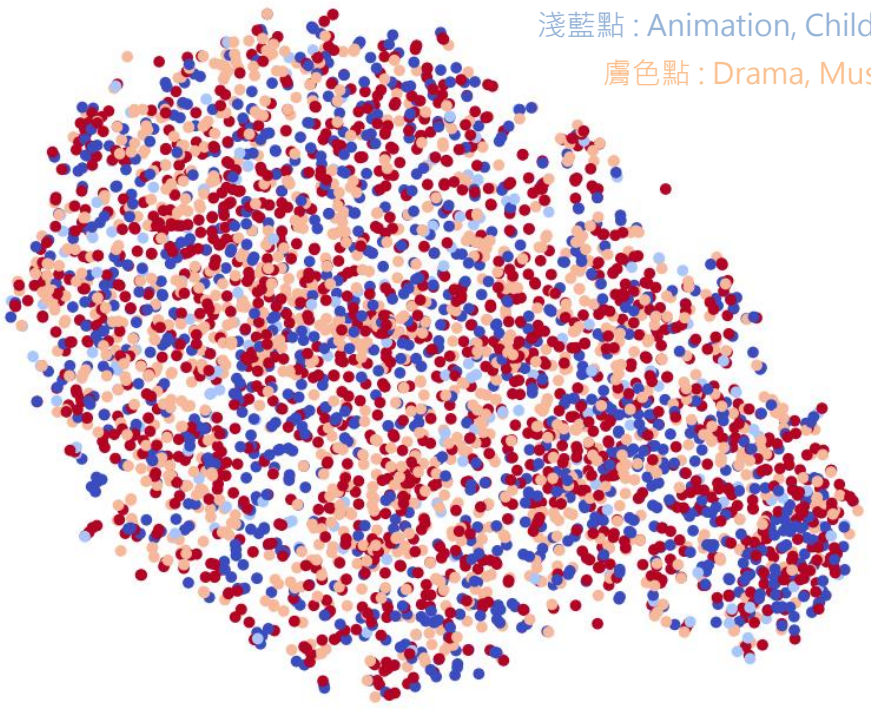
Dimension	Validation RMSE	Kaggle RMSE(Public)
8	0.8520	0.85702
16	0.8463	0.84475
32	0.8411	0.84231
64	0.8394	0.84001
128	0.8462	0.84880

可以見到，個人的 model 在 64 的 dimension 達到最佳值，所以可以參猜想說，實際的資料維度就是大約 64 左右的維度。另外在訓練過程中，越大的 Dimention 收斂的越慢，可能是參數多寡造成訓練過程的難易不同。

3. **(1%)比較有無 bias 的結果。**
   (collaborator: None)
   對於每個 movie 以及 user 個別加入 bias 的 embedding
   Kaggle Public 分數由 0.84089 提昇 rmse 至 0.84001
   可以印證手把手 PPT 當中的假設 : 每個 user 可能都會有自己評分的傾向，同樣的電影也會也這樣的趨勢。但修正幅度並不大。

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: 上學期學生 B04611015)
   將原先 user 以及 movie 的矩陣進行內積 (Dot) 改為將兩個矩陣連接 (concatenate)，Flatten 之後接到 Dense 層，我個人接了三層之後再向外接輸出層，並將 trainable params 湊到都 32 萬左右。
   DNN Kaggle Public : 0.88850
   MF Kaggle Public : 0.84431
   爛有夠多，可能是因為 NN 訓練過程對這個問題而言太過於複雜（卻沒有用的那種），雖然 nn 的速度很快，但訓練過程其實是相較 MF 複雜的；也許經過適當的 Tuning 可以得到較好的結果，但效率必定遠低於 MF。

5. **(1%)請試著將 movie 的 embedding 用 tsne 降維後，將 movie category 當作 label 來作圖。**
   (collaborator: None)



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: None)

   我同樣使用 MF，在 Rating 之外加入 User 的 Gender 以及 Age，我的假設是可能不同性別以及歲數，會有不同的分布傾向，可能中年某個歲數會特別厭世，電影評分特別低之類的；我把 Gender 以及 Age 使用 Concatenation 加入原先的 Matrix，在進行訓練。

   Kaggle Public 分數由 0.84089 退步至 0.84250

   猜想可能是因為年齡與性別的分布難以代表有這種評分傾向，相同年齡的厭世程度有自己的分布傾向，甚至受到地區、時代背景等等因素左右，年齡以及性別其實是干擾等等。