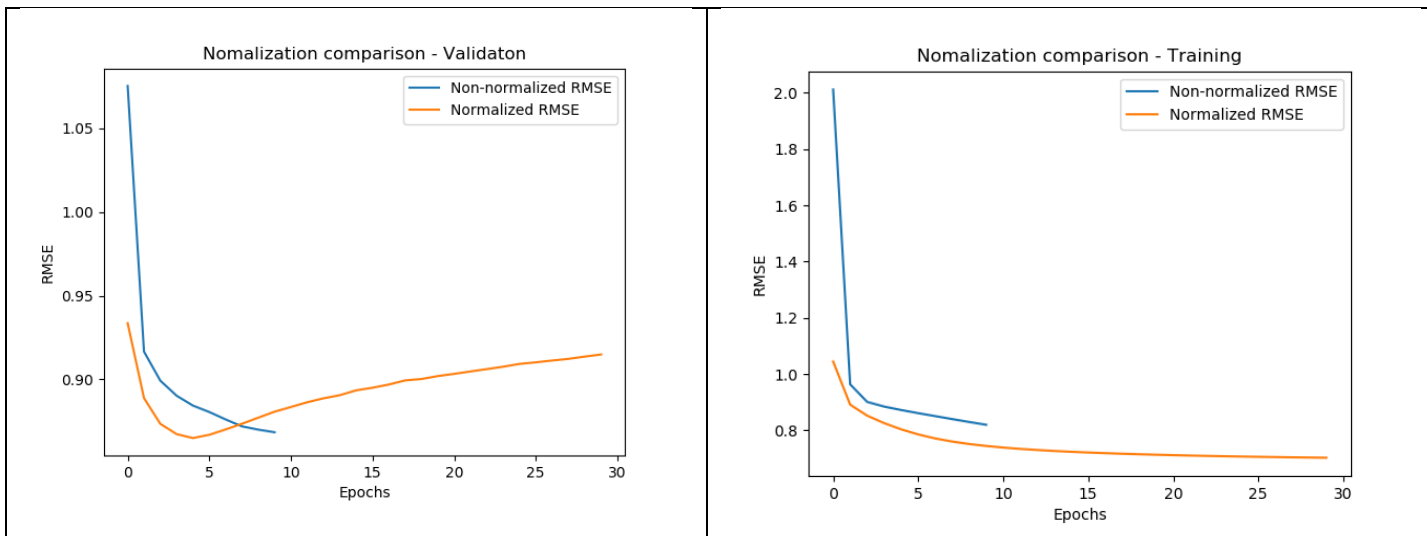


1. (1%)請比較有無 `normalize(rating)` 的差別。並說明如何 `normalize`.
(collaborator:)



左圖是 Validation 的比較，右圖是 Training 時的比較。

我使用的 Normalize 方式是對 Rating 減掉平均之後除以標準差。

可以看出有 Normalize 的收斂速度比較快，但是在運行夠多 EPOCH 之後沒 NORM 的也可以有差不多的結果。

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

outputFeedBack2_15D.csv 5 days ago by r06921058_>O add submission details	0.84638	0.85374	<input type="checkbox"/>
outputFeedBack2.csv 5 days ago by r06921058_>O add submission details	0.84801	0.85733	<input type="checkbox"/>

上方的是 15 維，下方的是 256 維，15 維的收斂速度相較 256 維長非常多，但是結果較 256 維好，推測可能是因為這次的題目中並不需要那麼多的維度，會影響的因素可能比較少一點。

3. (1%)比較有無 bias 的結果。
(collaborator:)

[output5.csv](#)
12 days ago by [r06921058_>.O](#)
[add submission details](#)

0.91119

0.91889

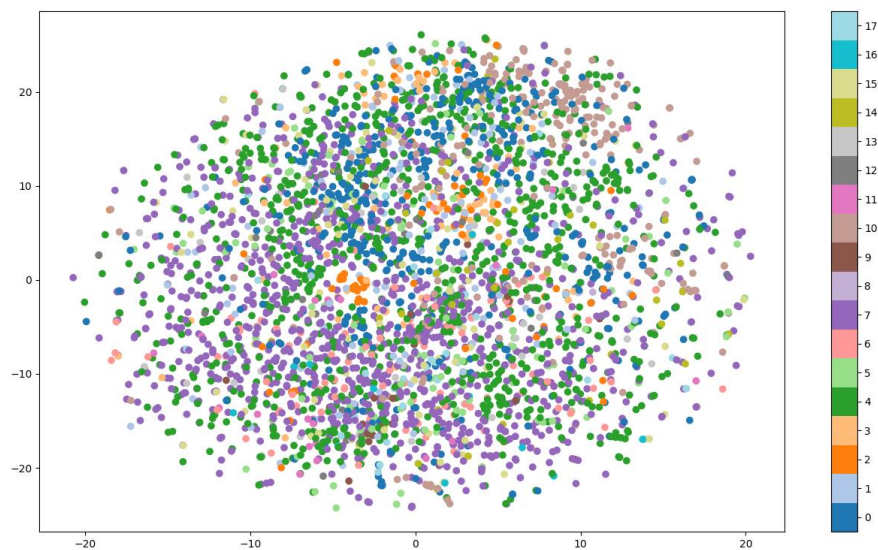
[output3.csv](#)
12 days ago by [r06921058_>.O](#)
[add submission details](#)

0.92208

0.92712

上方式有 BIAS 的，下方是沒有 BIAS 的。可以看出有 BIAS 的結果會好一點，推測是因為使用者本身打分數的習慣也會影響到結果，例如有人習慣從 4 開始給，有人可能習慣從 2 開始給。

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



以上是用 TSNE 將 movie 的類別降維到二維空間上的圖。紫色(主要在左下)和土褐色(偏右上)分的較開，可能是因為這兩種電影類別差比較多，而有些散步在整張可能是代表他是比較籠統的分類，可能很多電影都會同時具有這個特性。

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。
(collaborator:)

output6.csv 5 days ago by r06921058_>.O add submission details	0.84835	0.85608	<input type="checkbox"/>
output5.csv 11 days ago by r06921058_>.O add submission details	0.86425	0.87293	<input type="checkbox"/>

上方是有使用其他 **Feature** 的結果，下方是沒有使用的結果。使用的 **Feature** 是 **User** 評價過甚麼電影，電影被誰評價過，以及電影的年份。使用方式是將這三個特徵做成 **Embedding Matrix** 進去 **Train**。

6. Reference

<https://github.com/thtang/ML2017FALL/tree/master/hw5>

<https://github.com/idea7766/ML2017FALL/tree/master/hw5>