學號：R04921094 系級： 電機碩二 姓名：葉孟元

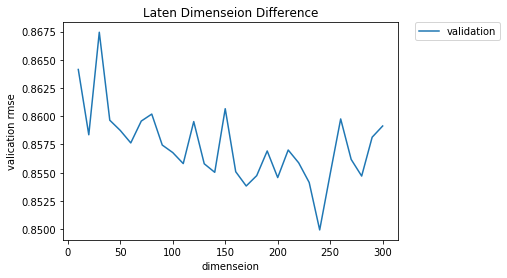
1. (1%)請比較有無normalize(rating)的差別。並說明如何normalize.  
   答：是有所差別的，做了normalize之後竟然在我這邊會比較差，其實我自己不是很清楚爲什麼，我也看了助教在fb上的討論，我試着解釋一下我的想法，我認爲，在feature上做比label上做更有意義，在feature上做會讓最佳化中所說的condition number變小，更好做gradient decent，但是在label上做就導致了variance變小更難被區分開。當然，還有一種可能性，因爲一個factor變了，包括regularization和dimension也應該跟着做對應的調整，自己沒做得比較好也只是因爲沒有找到合適的其他參數而已。

我採用的是sklearn中的StandardScaler，先做fit\_transform(), 之後再用inverse\_transform()。由於rating的數值改變了，所以在learning curve上的rmse和val\_rmse並不能直接比較。我最後比較是採用一起放到kaggle上看他們的public score和private score的對比。這兩個模型是完全一模一樣，除了rating上的差別。

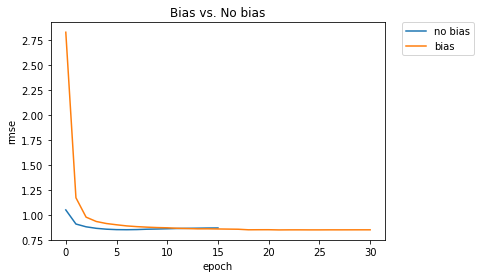
|  |  |  |
| --- | --- | --- |
|  | **Public score** | **Private score** |
| **normalized** | 0.85498 | 0.86047 |
| **Non-normalized** | 0.85325 | 0.85616 |

1. (1%)比較不同的latent dimension的結果。  
   答：因爲看了Netflix的論文，實驗結果中表明dimension越大越好，也符合直覺。但是，在自己實作的時候卻發現一個有趣的現象，當regularization固定的時候，dimension並不是越大越好，越大的dimension同時也是需要越大的regularization，經過一個學期的學習才發現這才是對的。可是，我在做這個作業的時候才發現，可是已經過了kaggle deadline，原本可以再調整我的模型，但是已經晚了。

可以看到，圖上到240是最好的，因爲我就是按照這樣dimension去調整regularization的，也必然會發生這樣的情況。之後就開始發生overfitting了。可以稍微看到240之後趨勢是上升的。



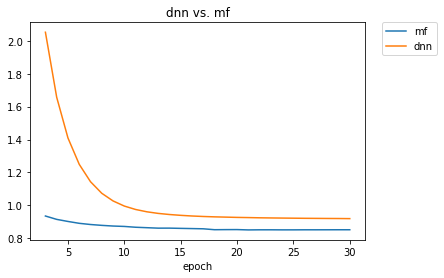
1. (1%)比較有無bias的結果。  
   答：有bias的結果會比沒有bias的結果明顯更好。我這邊採用的是固定的bias，也就是不能train的。其實考慮過trainable的模型，發現結果並沒有比較好。並且在這個題目上我發現了一個有趣的現象，validation的分數並不能說明一切，還是會有overfitting 盡管這邊看到的沒有bias的模型看起來也是不錯的，可是在kaggle上分數還是比較差，讓我知道k-fold還是有必要的。下圖確實是表明了有bias的模型會比較好一點。但是卻沒有kaggle上的分數差別明顯。



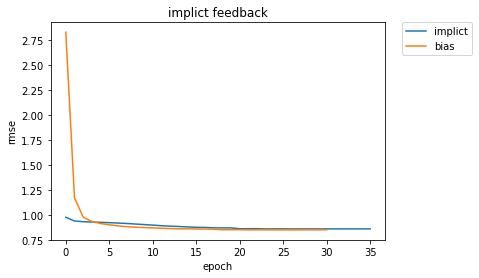
|  |  |  |
| --- | --- | --- |
|  | Public score | Private Score |
| bias | 0.85325 | 0.85616 |
| No bias | 0.85975 | 0.86378 |

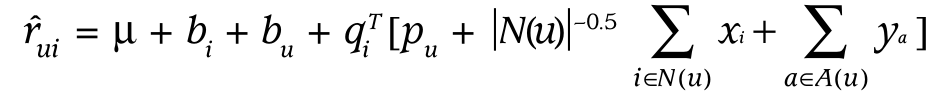
1. (1%)請試著用DNN來解決這個問題，並且說明實做的方法(方法不限)。並比較MF和NN的結果，討論結果的差異。

答：我採取的方法還是利用embedded layer然後將user和movie vector相加得到一個新的vector，再接上3層128的DNN，原本想將這個做成classfication的問題，可是由於是用rmse做最後的衡量，結果很差。就還是利用regression的方法來做，下圖是做出來的結果。

我認為，要利用DNN來做這個問題如果還是用兩個vector來描述這個問題並不會比較好，畢竟只有對於兩個vector 的inner product才可以描述這個ratting，而不是我這種方法。這樣必然會導致結果更差。

1. (1%)請試著將movie的embedding用tsne降維後，將movie category當作label來作圖。  
   答：因為final時間問題，最後一天才開始寫report，時間關係這題沒有實作。請助教扣分吧，不好意思。
2. (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。

答：自己採用的是Netflix prize論文處理implicit feedback的做法，但是只做了user方面的，並沒有將movie的feature加進去。因爲我user做出來就沒有比之前好，於是就放棄了。將user的feature先做preprocessing，首先先將age變成age group，0歲換爲mean。之後將age group, Gender, Occupation, Zip-code都變爲一個和user同一維度vector。最後將這些都加起來做爲user的vector。

下圖是比較圖，其實加上implicit feedback並沒有更好，當然，這確實是有可能我沒有用好的regularization的結果。因為自己用bagging的方法取得了public 0.84467的分數也就沒有再深究這個方法了。