

學號：R05229014 系級：大氣碩二 姓名：鄒適文

1. (1%)請比較有無normalize(rating)的差別。並說明如何normalize.
(collaborator: r05229016 羅章碩)

*使用latent dim = 256

我的normalize方法是將training data的rating減去rating平均再除掉rating標準差，
而在testing的時候是這樣預測的；

$$pred_{test} = pred_{test}^* \times train_{std} + train_{mean}$$

並且將pred_test > 5的設成5、pred_test < 1的設成1。

但是發現結果變差了，在原本沒有normalize的kaggle分數是:(pri: 0.85928 pub : 0.85928)

而normalize後的分數是：(pri: 0.88700 pub: 0.88778)

可能這種normalize方式並不適合做在這種線性的預測方法，因為使得rating的分數差異變得沒有這麼分開，我覺得可能要在nn上實作看看用分類的方法預測有比較好的結果。

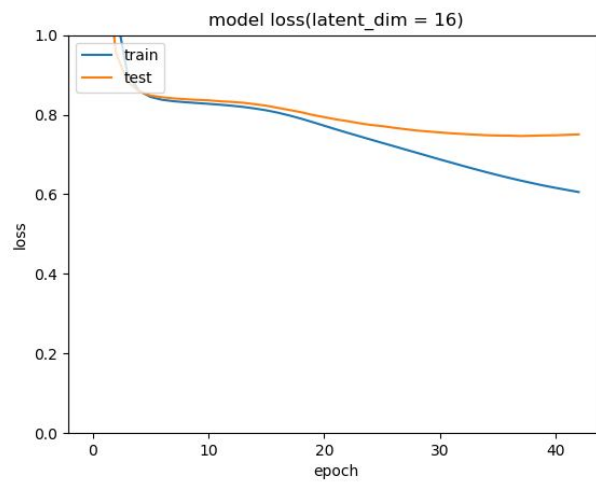
2. (1%)比較不同的latent dimension的結果。
(collaborator: r05229016 羅章碩)

這裡比較了latent dimension在 16, 32, 64, 128, 256, 512, 1024的差異，並且使用的MF_model是有加入bias的。

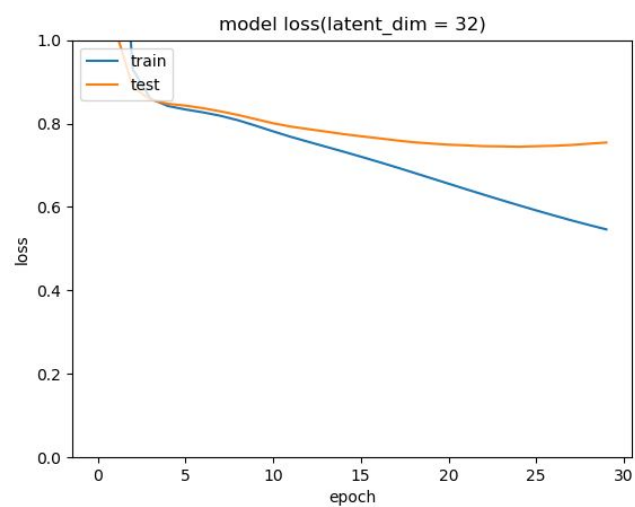
模式參數：batch_size = 2000, epochs=200, validation_split=0.1, dropout_rate=0,

earlystopping(patience=5)

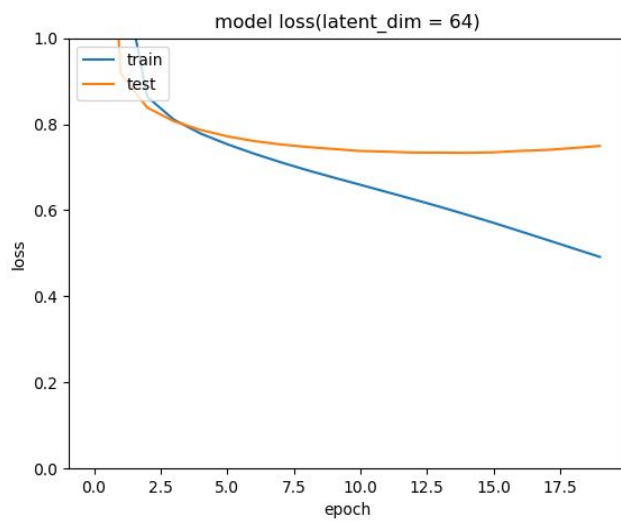
latent dim = 16 kaggle:(pri: 0.86358 pub: 0.86392)



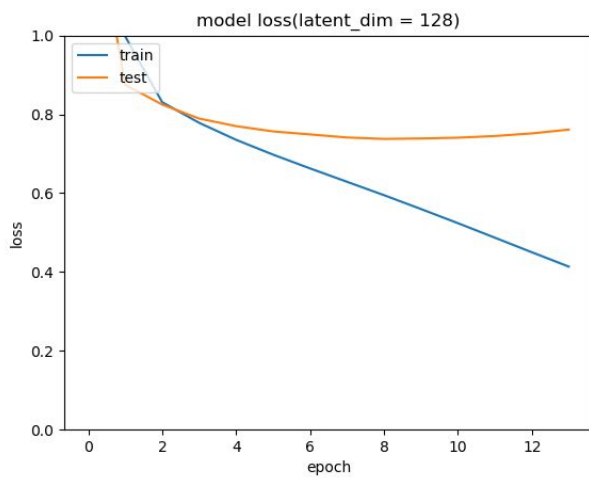
latent dim = 32 kaggle:(pri: 0.86099 pub: 0.85955)



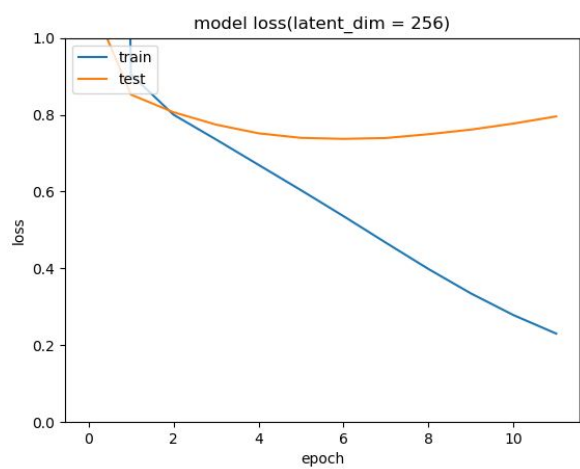
latent dim = 64kaggle:(pri: 0.85679 pub: 0.85624)



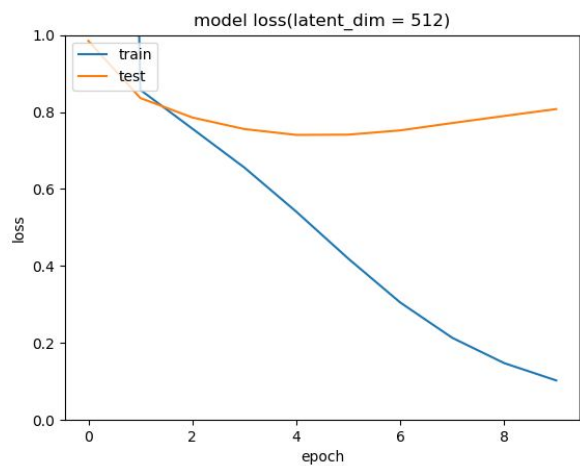
latent dim = 128kaggle:(pri: 0.86099 pub: 0.85955)



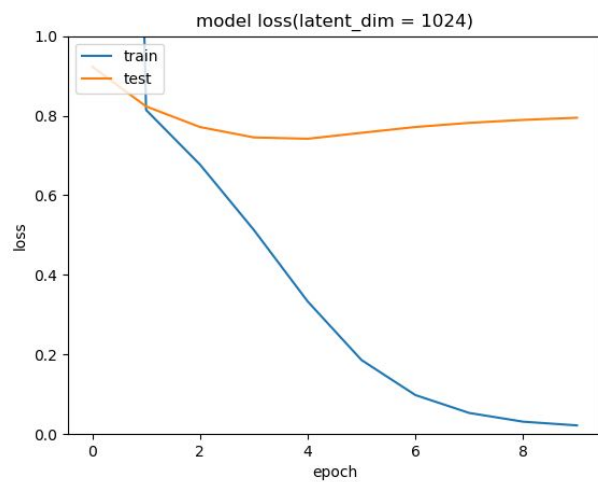
latent dim = 256kaggle:(pri: 0.85729 pub: 0.85591)



latent dim = 512kaggle:(pri: 0.85928 pub : 0.85928)



latent dim = 1024kaggle:(pri: 0.86290 pub: 0.86179)



以上幾張圖是各個不同latent dimension的訓練過程，而在這些訓練過程中我發現；

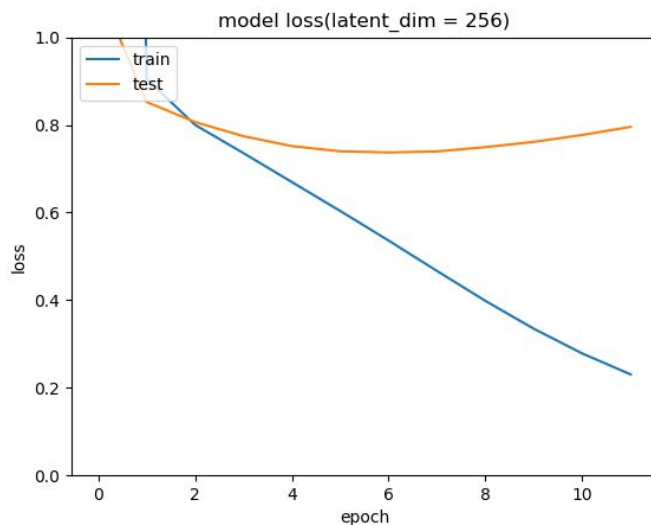
1. 低latent dim的狀況需要比較久的訓練來收斂
2. 高latent dim的狀況很容易overfitt
3. 我測試最好的結果是latent dim = 256(當然更高維度的也有可能分數一樣高，但還需要再調整其他參數)

3. (1%)比較有無bias的結果。

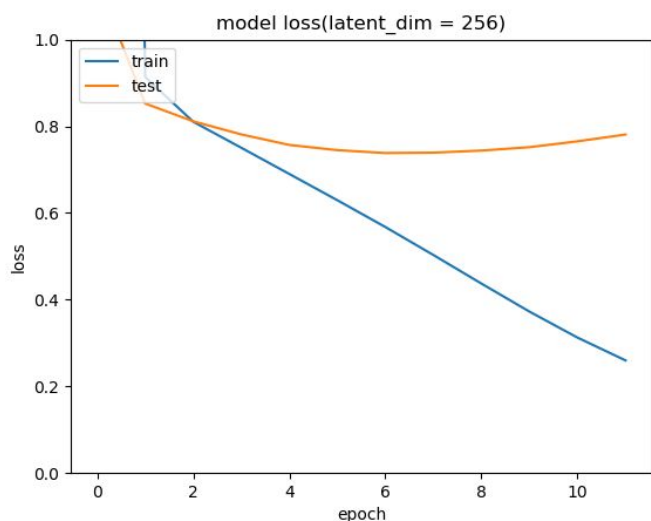
(collaborator: r05229016 羅章碩)

在有測試無BIAS的狀況下，我都將MODEL的latent dimension設在256，發現在這個狀況下，有沒有加入BIAS都讓vaildataion set的表現是差不多的，而在kaggle上的表現則是有加入BIAS的表現要好一點，加入BIAS讓public與private的loss都下降了0.003左右，算是個不錯的進步，顯然這組資料下，每個user可能都會有自己評分的傾向，像是傾向於把每部電影都評得很高分或者很低分；同樣的電影也會有這樣的趨勢。

有BIAS



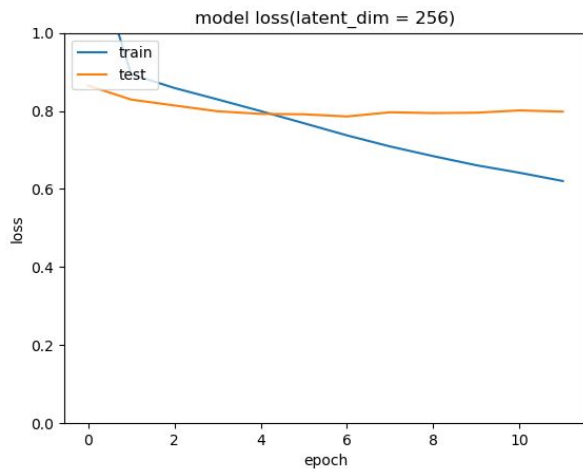
無BIAS



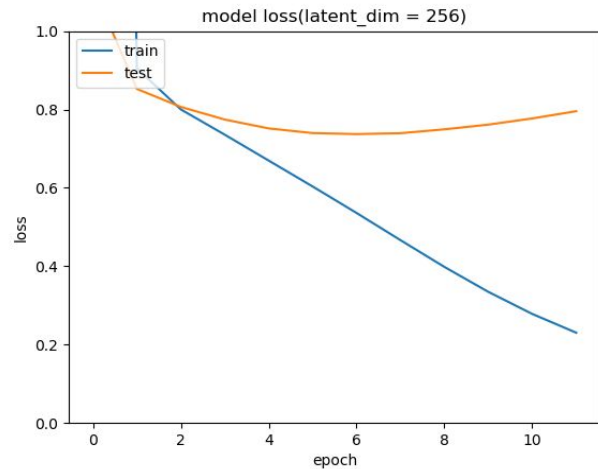
4. (1%)請試著用DNN來解決這個問題，並且說明實做的方法(方法不限)。並比較MF和NN的結果，討論結果的差異。

(collaborator: r05229016 羅章碩)

我將user embedding以及movie embedding concatenate在一起再通過DNN得出rating
跟MF的方法很像都是預測出一個rating。



NN



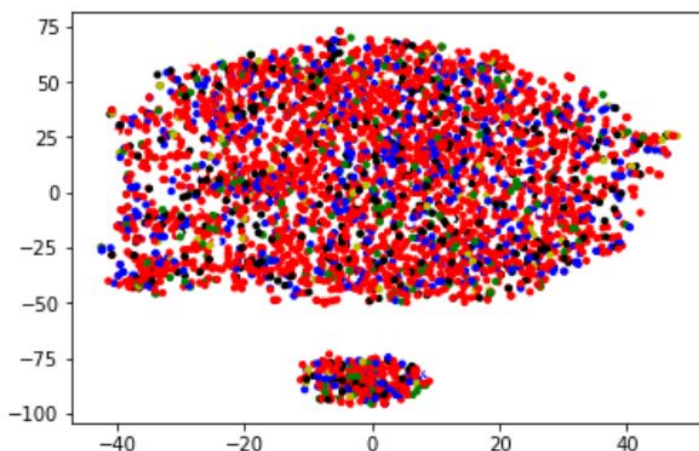
MF

(NN只用了3層 參數分別為256, 128,1)

發現NN在收斂速度上比較快，幾乎在第二個epoch就收斂了，而且比較不容易overfitting，但在NN神經元不多的情況下(我的測試中)，還沒有最後結果比MF好的。而MF的優點大概是我覺得我目前做過的作業中，參數最好調整的一個model不太需要調整就會有很不錯的表現。

5. (1%)請試著將movie的embedding用tsne降維後，將movie category當作label來作圖。

(collaborator: r05229016 羅章碩)



我用latent dim=16來降維 並把類似的電影把它分成同一種顏色，但似乎這樣並沒有使全部的資料分得很開。

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: r05229016 羅章碩)