學號：B04501002 系級： 土木三 姓名：謝至宥

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

   我的測試使用的model都大致相同，是4個embedding將user和movie轉成vector和bias，embedding後面都接dropout，output就是embedding內積再加bias，loss使用mse，optimizer使用adam，latent dimension約為500，再根據題目微調。
   做normal時，在train計算出平均值和標準差，將rating = (rating-平均)/標準差。在test時，將預估結果*training的標準差再加平均值。在kaggle上將public和private取平均，normalize過的rmse為0.89391，沒normal的為0.90532，根據實驗結果，有normalize的表現比較好，可能是做normalize後收斂的速度比較快，所以結果比較好。

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

   我測試時都跑15個epochs並上傳kaggle，測試了128、256、512、1024，得到的rmse分別為0.898、0.87743、0.87629、0.88744，好像沒有明顯的趨勢，其中1024的訓練時間長、表現又不如512的好、可能是overfit，128則是太簡單，誤差較大，參數太多或太少對模型都是不好的。

3. (1%)比較有無bias的結果。
   (collaborator:)

   我使用dim 512做測試，將bias的layer拿掉，在kaggle上有bias的rmse為0.89685，沒有的則是0.8992，確實符合預期，有bias的誤差較小，使用者或電影可能會因對象不同，而有較高或較低的偏差。

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:)

   我是了幾個dnn，都是過embedding後接dense layer，不過沒有試到比MF還要更好的model，有個是接512、256、128、64的dense，loss是0.9024，有個比較簡單的只有兩層，分別為150、50，得到loss為0.8992，表現較好，不過都沒到MF的效果，而且train時發現比MF容易overfit。

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

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:)

   我將movies.csv中的category中選第一個當作標籤，用1 of N encoding concatenate在資料後面，user也是類似，最後結果loss約0.87，可能是因為我train的epochs比較少，所以來不及看到效果。