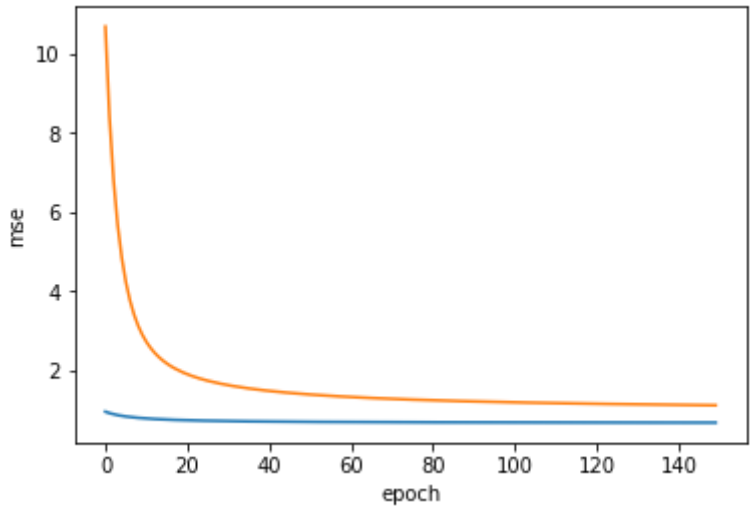
學號：r05922135 系級： 資工碩一 姓名：蔡慶源

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



上圖紅色是沒做normalize，可以看的出來收斂速度較慢，而且前幾個epoch的mse大很多，不過最後會收斂得差不多。

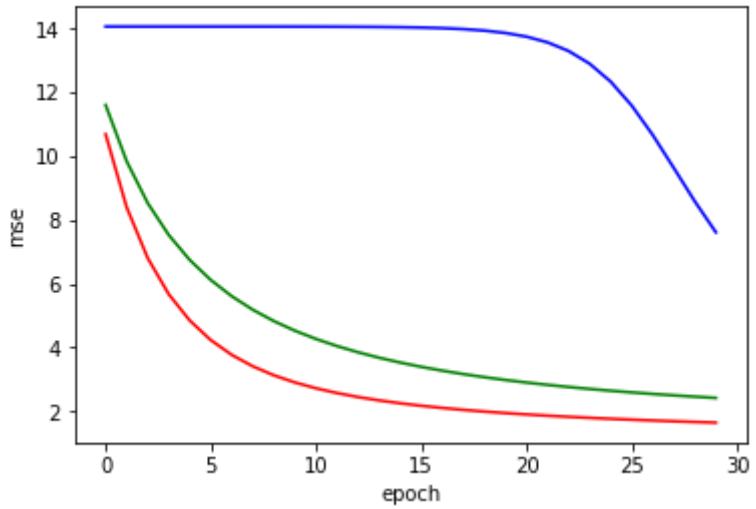
normalize method :

(target - np.mean(target)) / np.std(target)

2. (1%)比較不同的latent dimension的結果。

我實驗4種不同的維度，分別是2, 10, 50, 100，得到的結果都差不多，不管是收斂速度還是mse，但維度越低訓練時間明顯降低，所以我認為做 mf 的時候不需要使用太高的維度。

3. (1%)比較有無bias的結果。



紅色:有 bias，綠色:去掉一個 bias，藍色:去掉所有 bias

可以明顯的看出來有加bias能讓結果變好，且沒有 bias 和有 bias 的模型收斂趨勢不一樣，有bias的模型前幾個epoch就會開始明顯減少 loss。

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。
   DNN實作:

   使用助教的範例改寫，把user id 和 movie id 通過 embedding 層轉換成向量，並在embedding層和dense層中間加入epoch normalize。

   討論:

   如果不使用額外training data(users.csv, movies.csv)，mf的結果會稍微優於 dnn，這個結果很合理，因為我們是拿 dnn 來模擬 mf， 如果不使用其他 training data ，dnn 不會比 mf 結果好。

   Training 速度兩者相差不多，收斂速度也是。如果不normalize， dnn 的收斂速度快很多，可能是因為 mf 在 embedding 的時候有做 normalize 導致 user vector 和 movie vector 內積的結果遠小於 rating，這樣要更新很多次參數才會到最佳解，dnn 雖然也有這個問題，但 dense layer 在 activation function 會額外多一個bias來改善這個問題。

   mf 的 mse 會穩定下降，但 dnn 有時候會突然上升，之後又慢慢降回來，其原因可能是因為 mf 的 loss function 是碗型，而 dnn 的 loss function 有局部最小值。

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

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。

我實作 dnn 額外使用 users.csv 和 movies.csv。

User 使用 gender、age、occupation、zip-code。Gender 把 F 定為 0，M 定為 1，age 直接使用並做normalize，occupation 和 zip-code 視作字串，分別先做word index 再把他們分別用 embedding layer 轉成向量。

Movie 只使用分類，總共有18種分類，每一種 movie 維護一個 18 維的向量，這個 movie 有哪種分類，相應的位置就設為 1，其餘設 0。

之後把上述的這些和 mf 產生的embedding vector 串起來，跑一個三層 dense，即是使用額外 data 的 dnn 實作。