

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

normalize的方式是先算出 training data 之 rating 的 mean及 std，然後將所有training data 先減去 mean 再除以 std，而 predict 出的結果再乘上 std 並加回 mean 得到最後的結果。

將 epochs 固定為 18，latent dimension 設定為 120，optimizer使用 adamax，並且 training data 經過相同的方式 shuffle，得出之 prediction 結果在 Public Leaderboard上之 loss(rms)分別如下：沒有normalize：0.86191，normalize後：0.90884，normalize過後之結果反而變差，這可能是由於 testing data之 mean 與 std 和 training data，導致再將 predict 的結果轉換回真正的 rating時，產生了偏差。

2. (1%)比較不同的latent dimension的結果。

dim.	1	3	5	10	20	40	80
loss	0.8483	0.7909	0.7791	0.7778	0.7776	0.7831	0.7930

上表為使用不同的 latent dimension 在 validation set 上所得到的 loss(mean squared error)，由此結果可知，當 latent dimension 太小時，embedding layer 會喪失足夠的訊息，因此 loss 較大，而當 latent dimension 增大到一定程度後(在這個例子大約為 5)，loss 約略維持定值，但當latent dimension 太大後，又會因為參數過多而容易導致 overfitting 的現象。

3. (1%)比較有無bias的結果。

固定 latent dimension 為 10，optimizer 為 adam，比較有無 bias 在validation set 上之 loss(mse) 分別如下：無 bias：0.7778，有 bias：0.7720。

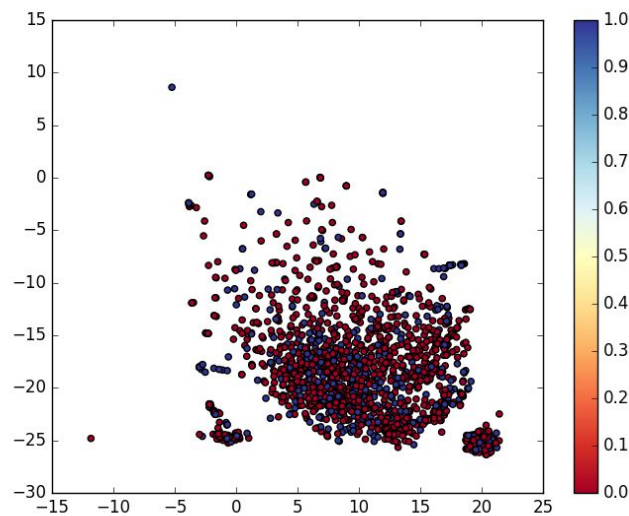
結果顯示，有bias的performance 略好，這是一個符合預期的結果，但是使用 bias 顯然會使總共的參數量增加，因此更容易造成 overfitting。

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。

固定 latent dimension 為 10，將 users 對應的 embedding layer 和 movies 對應的 embedding layer 作 Concatenate，並將此結果作為 DNN之 input，DNN的架構為經由一層 hidden layer(dimension: 200) 到達output layer(即為 rating)。

此簡單的 DNN 在 validation set 上之loss (mse)為 0.7658，相較於在相同的latent dimension 下之MF的 loss(0.7778)，performance更好。這樣的結果並不意外，因為 MF 限定了模型為兩個 latent vecotor 的內積，但反之，DNN的模型可以更為複雜，故DNN 模型的 bias 會比 MF，但相對的，其 variance會比MF大。

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



上圖之紅色點對應到 Children's、Comedy 這兩類的 movies、藍色點對應到 Crime、Horror、Film-Noir 這三類的 movies。這兩種對應到的 movies 在客觀上有著極大的差距，前者偏向正面情感、後者則較為灰暗。然而意想不到的，tsne降維後的結果十分的差，並無法將兩個種類做出區分。很有可能的是因為我的 MF model 本身就有蠻大的 loss，以致於由該 model 取出之電影的 vector 無法有效代表該電影的抽象意義。

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。

使用 users 的性別作為額外的 feature，0 代表男性，1 代表女性。在 DNN 的 model 中，在形成 users 的 embedding vector 時所用到的 input 除了原本的 user id 外，再加上性別這項 feature，其餘任何 model 都與第四題的模型相同。結果在 validation set 上之 loss 為 0.9597，比第四題的結果差了更多，可能的原因是性別與 rating 本身的 correlation 並不是非常高，此外多了性別這項 feature 反而使得原本每個 user 的特性在形成 user 的 embedding vector 時的效果被減弱，因此使得 train 出來的結果變差。