

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

減掉mean再除以std，使得mean為零，std=1，實際跑發現，有做normalize效果會變差，而且差蠻多的，達到4%左右，我想可能是因為做了反而使得原本rating間的差異改變，以下是用dnn模型，維度為120來做的結果。

	private	public
有	0.85699	0.86384
無	0.89361	0.90062

2. (1%)比較不同的latent dimension的結果。

使用dnn的模型，嘗試了維度90、120、150的結果，發現維度越高，效果越好，也許可以再嘗試更高的維度看看，但也表示訓練資料中包含了許多資訊，需要用到更多的特徵來表示，實驗結果如下，90和120差距約2%，但120和150之間其實差距就不大了，顯見維度再增加，對正面效果的影響應該也會逐漸變少。

	90	120	150
private	0.87226	0.85699	0.85371
public	0.87963	0.86384	0.86193

3. (1%)比較有無bias的結果。

使用mf模型，維度150來做比較，發現沒有bias會影響效果蠻多的，約2%左右，表示的確需要bias來幫助記住每個人和每部電影本身的偏差值，實驗結果如下。

	private	public
有	0.85760	0.86137
無	0.88128	0.88726

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:r06725053 資管碩一 韓宏光)

下圖是dnn的模型，把user、movie的embedding結果Concatenate在一起後，作為input，再透過兩層dense去學習，最後輸出一維的數值結果，即為最後的分數，實際跑發現，其實mf模型的效果會比dnn好上一點，以下是實驗結果，發現差距

非常小，但dnn的訓練速度比較快。

	private	public
mf	0.85760	0.86137
dnn	0.85371	0.86193

```
def nn_model(n_users, n_items, latent_dim=120):
    user_input = Input(shape=[1])
    item_input = Input(shape=[1])

    user_vec = Embedding(n_users, latent_dim, embeddings_initializer='random_normal')(user_input)
    user_vec = Flatten()(user_vec)
    user_vec = Dropout(0.5)(user_vec)

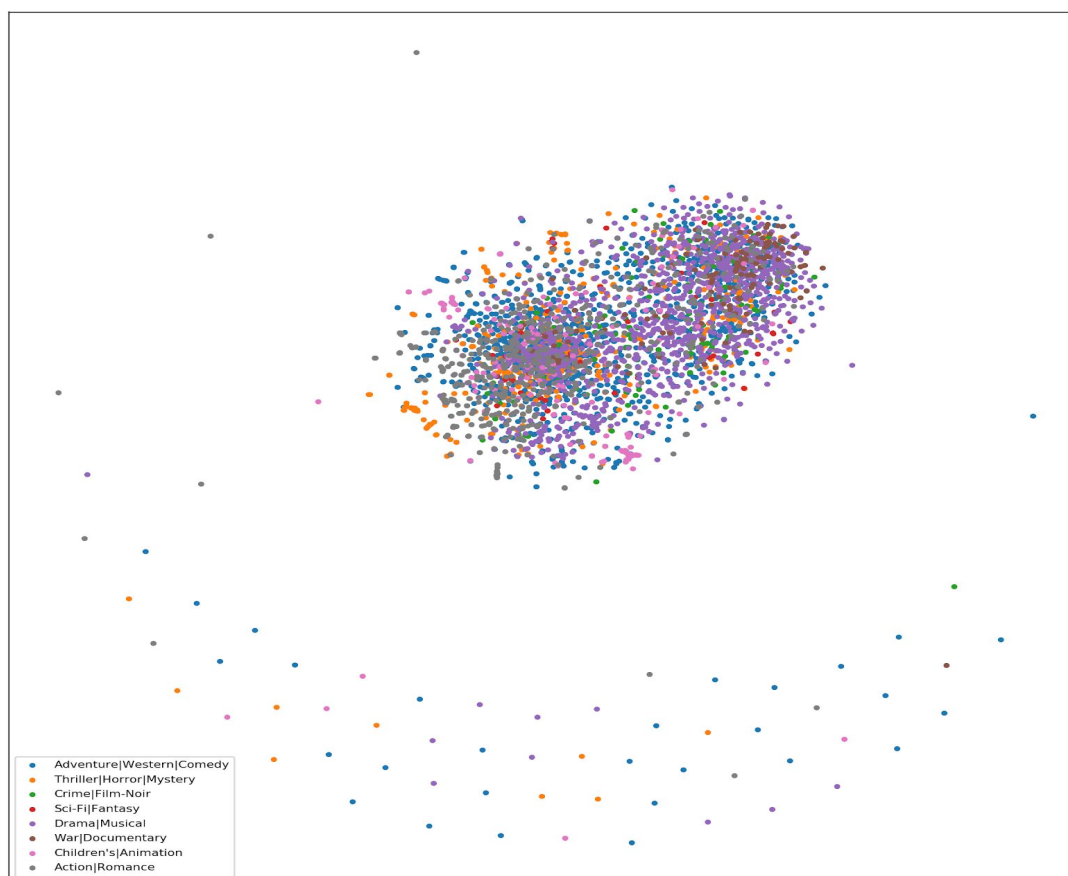
    item_vec = Embedding(n_items, latent_dim, embeddings_initializer='random_normal')(item_input)
    item_vec = Flatten()(item_vec)
    item_vec = Dropout(0.5)(item_vec)

    merge_vec = Concatenate()([user_vec, item_vec])

    #hidden = Dense(150, activation='relu')(merge_vec)
    hidden = Dense(150, activation='linear')(merge_vec)
    hidden = LeakyReLU(alpha=0.001)(hidden)
    hidden = Dropout(0.5)(hidden)
    #hidden = Dense(50, activation='relu')(hidden)
    output = Dense(1, activation='linear')(hidden)
    output = LeakyReLU(alpha=0.001)(output)

    model = Model([user_input, item_input], output)
    #model.compile(loss='mse', optimizer='sgd')
    model.compile(loss='mse', optimizer='adam')
```

5. (1%)請試著將movie的embedding用tsne降維後，將movie category當作label來作圖。(collaborator:r06725053 資管碩一 韓宏光)



如上圖，參考前人的分類來做圖，使用mf模型，維度150來做圖，可以發現旁邊有一圈發散的點，但中間又有以兩個中心為主集中的區域，其實看不太出來不

同類電影的label分佈。

<https://github.com/Jy-Liu/ML2017/blob/master/hw6/draw.py>

https://github.com/qhan1028/Machine-Learning/blob/master/hw6/plot_model.py

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:)