學號：B03505031 系級： 工海四 姓名：邱昱軒

1. (1%)請比較有無 normalize(rating)的差別。並說明如何 normalize.
   (collaborator:)
   在同樣的條件下(DIM=75,earlystop patience=15)，將 rating 做了
   rating=(rating-np.mean(rating))/np.std(rating)
   再丟進 MF 訓練，並將取得的 mean 及 std 放入 predict 以還原分數。

	Kaggle	收斂(earlystop=15 時的 epoch)
Normalize 前	0.85801	163
Normalize 後	0.85892	144

   可以發現在 normalize 後，收斂較快且 rmse 較小

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

DIM	50	75	100	200	300	400
Val_rmse	0.85718	0.85713	0.8583	0.86238	0.869	0.87525

   原本以為 dim 越高越好，但實驗過後發現，在同樣的 dropout 下，大約在 dim=75 時有最好的效果。可能是因為 dim 太大反而容易對 training data 過擬合。

3. (1%)比較有無 bias 的結果。
   (collaborator:)
   以 DIM=75 及 200 做實驗

DIM	Bias	Val_acc	收斂(earlystop=15 時的 epoch)
75	0	0.8570	179
75	1	0.85713	163
200	0	0.86285	207
200	1	0.86238	178

   發現 val_acc 似乎不論有沒有加 bias 都差不多，但加上 bias 後收斂較快。推測是因為可訓練參數變多的關係。

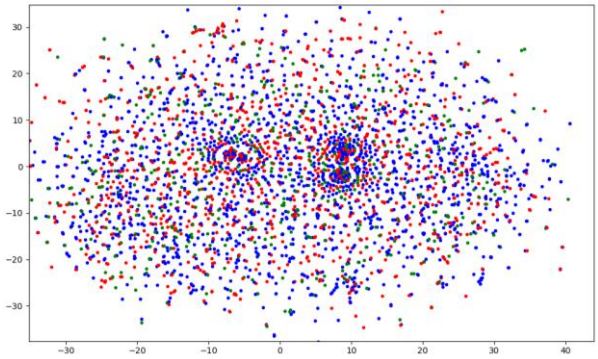
4. (1%)請試著用 DNN 來解決這個問題，並且說明實做的方法(方法不限)。並比較 MF 和 NN 的結果，討論結果的差異。
   (collaborator:)
   將 userID 的 embedding vector 和 movieID 的 embedding vector concatenate 後，進入一層 Dene(256,activation='elu')和一層 Dene(64,activation='elu')和一層 Dene(,activation='linear')後得到 rating。
   以下比較 embedding dim=75 時的 MF 及 NN

	Val_rmse	收斂(earlystop=15 時的 epoch)
MF	0.85713	163
NN	0.87161	67

在準確率方面，MF 的表現較佳，應該是因為 MF 隱含的是 user 及 movie 間的關係(該 user 對不同的電影的偏好)，但 NN 只是強行帶入使之趨近結果。但 NN 的收斂速度比 MF 快許多，應該是因為可訓練參數較多的關係。

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

   紅色: Animation / Comedy
   綠色: Action / Crime / War
   藍色: Triller / Horror



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:)

```
ui = Input(shape=[1])
uv=Embedding(MAX_USER_ID,embedding_dim,embeddings_initializer='ones')(ui)
uv=Flatten()(uv)

mi = Input(shape=[1])
mv=Embedding(MAX_MOVI_ID,embedding_dim,embeddings_initializer='zeros')(mi)
mv=Flatten()(mv)

msi = Input(shape=[1])
msv=Embedding(2,embedding_dim,embeddings_initializer='zeros')(msi)
msv=Flatten()(msv)

fsi = Input(shape=[1])
fsv=Embedding(2,embedding_dim,embeddings_initializer='zeros')(fsi)
fsv=Flatten()(fsv)

ai = Input(shape=[1])
av=Embedding(MAX_AGE,embedding_dim,embeddings_initializer='zeros')(ai)
av=Flatten()(av)

oi = Input(shape=[1])
ov=Embedding(MAX_MOVI_ID,embedding_dim,embeddings_initializer='zeros')(oi)
ov=Flatten()(ov)

fv = Concatenate()([uv,mv,msv,fsv,av,ov])
#dnn = Dense(512, activation='elu')(fv)
#dnn = Dropout(0.5)(dnn)
dnn = Dense(256, activation='elu')(fv)
dnn = Dropout(0.4)(dnn)
dnn = Dense(64, activation='elu')(dnn)
dnn = Dropout(0.3)(dnn)
dnn = Dense(1, activation='linear')(dnn)
```

除了 userID 和 movieID，我將性別為男、性別為女、年齡、職業也經過 embedding ，再將所有 vector concatenate 後進 NN 訓練，大約在第 55 個 epoch 時，val_rmse 收斂至 0.8635，比原本只用 userID 和 movieID 當作訓練資料的 NN 還好(val_rmse=0.87161)，但仍比 MF 的 0.85713 還差一點。