學號：B04902105 系級： 資工三 姓名：戴培倫

1. (1%)請比較有無normalize(rating)的差別。並說明如何normalize.
   我將rating減平均後除以標準差。

   normalize後RMSE與原本差不多，並沒有什麼顯著的進步，但是收斂速度快很多。沒有normalize的情況val_loss大約要35個epoch才收斂，但normalize後大約10個epoch就收斂了。

2. (1%)比較不同的latent dimension的結果。

   整體設定如下：

   - 沒有normalize
   - 有加bias
   - Batch size = 1000
   - Epochs = 30
   - Optimizer = 'adam'
   - Loss = 'mse'
   - Validation set為shuffle過的90000筆data，約佔10%

Dimension	16	32	64	128	256
loss	0.8062	0.7864	0.7089	0.5735	0.4296
val_loss	0.7925	0.7581	0.7367	0.7412	0.7668

   從結果可以看到latent dimension愈大，model fit的速度愈快，128 dim的在13個epoch時便開始overfit，256 dim的在第6個epoch就開始overfit了。

3. (1%)比較有無bias的結果。

   best model: latent dimension=96, 有加bias, 沒有normalize

   使用跟best model同樣的條件，只拿掉bias，比較準確率。發現有加bias的情況下，fit的速度快非常多，大約75個epoch val_loss就收斂了，但沒有bias到約300 epoch才收斂。

Keras public score:

有加bias: 0.84979

沒加bias: 0.85921

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。

我是將原本MF Embedding, Flatten後的兩個vector concatenate傳進dense裡，有試過多加幾層，發現很快就overfit且loss都降不太下去，無論是val_loss或public score都比MF差。

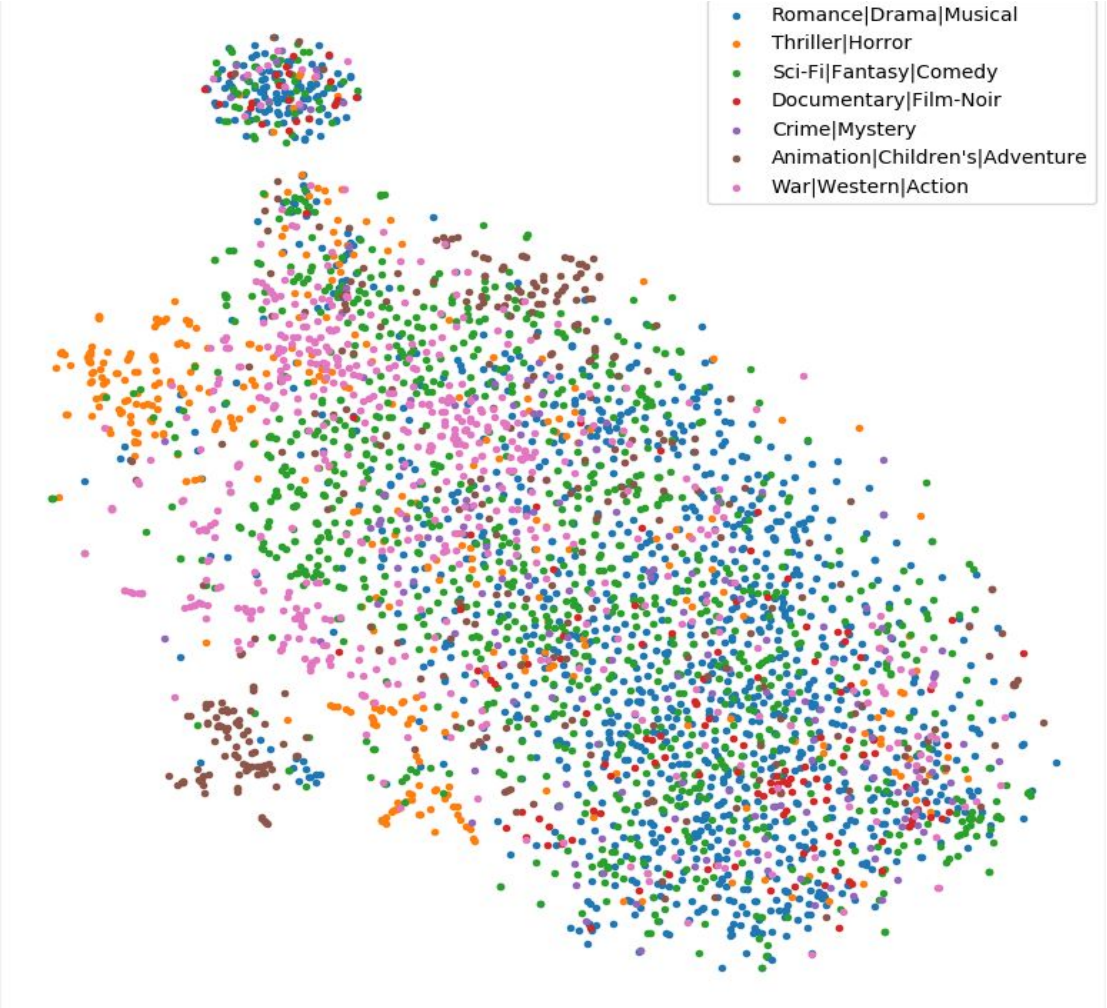
Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 1)	0	
input_2 (InputLayer)	(None, 1)	0	
embedding_1 (Embedding)	(None, 1, 96)	579936	input_1[0][0]
embedding_2 (Embedding)	(None, 1, 96)	372864	input_2[0][0]
flatten_1 (Flatten)	(None, 96)	0	embedding_1[0][0]
flatten_2 (Flatten)	(None, 96)	0	embedding_2[0][0]
dropout_1 (Dropout)	(None, 96)	0	flatten_1[0][0]
dropout_2 (Dropout)	(None, 96)	0	flatten_2[0][0]
concatenate_1 (Concatenate)	(None, 192)	0	dropout_1[0][0] dropout_2[0][0]
dense_1 (Dense)	(None, 128)	24704	concatenate_1[0][0]
dropout_3 (Dropout)	(None, 128)	0	dense_1[0][0]
dense_2 (Dense)	(None, 1)	129	dropout_3[0][0]

Total params: 977,633

	Kaggle public score:	Training val_loss:
MF	0.84979	0.7461
NN	0.86737	0.7367

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

我將18個category中幾個較類似的合併，最後取了7個分類，從圖中可以看出大部份顏色都看得出有聚在一起，其中Thriller | Horror（橘）、War | Western | Action（粉）、Animation | Children's | Adventure（棕）比較明顯。而Sci-Fi | Fantasy | Comedy（綠）、Romance | Drama | Musical（藍）則四散各地。



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。

我使用了user data中的gender, age, occupation，以及movie data的genres並將其轉成one-hot vector，總共為3+18 = 21個feature。

Training loss: 1.1519, val_loss: 1.1571

Kaggle public score: 1.07534

訓練時loss下降的幅度非常小，有試過調小batch_size和改變optimizer，結果都差不多，訓練了100個epoch，收斂在1.152。