學號：B03902101 系級： 資工四 姓名：楊力權

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

   (collaborator: No)

   用相同架構(無bias latent dimension=128)，將rating除以5，讓rating介於0-1之間來normalize，predict後再乘上5輸出，並比較其結果如下：

	public score	private score
有normalize	0.89376	0.89486
無normalize	0.87196	0.87555

   發現不做normalize的結果好很多。

2. (1%)比較不同的latent dimension的結果。

   (collaborator: No)

   我嘗試用相同架構(有加入bias)使用latent dimension 32, 64, 128與256在validation data loss收斂後predict得到結果如下：

latent	public score	private score
32	0.85979	0.86001
64	0.86039	0.86352
128	0.85926	0.86005
256	0.85533	0.85520

   發現latent dimension 256是最好大小。

3. (1%)比較有無bias的結果。

   (collaborator: No)

   我嘗試在相同架構下(latent dimension=128)，使用有在矩陣相乘之後加上bias與沒有，得到結果如下：

	public score	private score
有bias	0.85926	0.86005
無bias	0.87196	0.87555

   發現有加入bias的結果好非常多，因為每個獨立使用者給的評價顯然是會比較一致，而電影得到使用者的評價也是，所以在MF之後加上bias有很顯著的幫助。第2.題的數據也是建立在有bias之上。

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: No)

NN做法：將user與movie都做一個dimension 64的embedding，並將兩個concatenate成一個128 dim的tensor，再過兩層256的dense就輸出。

MF做法：同樣user與movie都是latent dimension 64的embedding，並dot起來後加入bias。

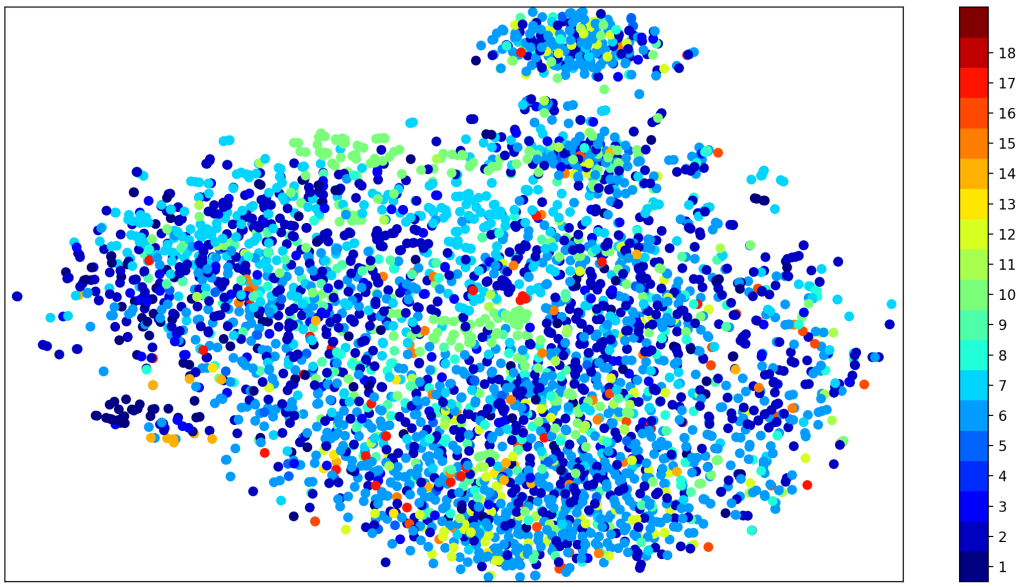
	public score	private score
MF	0.86039	0.86352
NN	0.89691	0.89809

使用MF的效果比單純的NN來得好，或許是concatenate的NN會喪失較MF更多相互關係的資訊。

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

(collaborator: No)

將每個movie的第一個類別當作label做圖。



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。

(collaborator: No)

(a.)使用額外feature：movie category, user sex, user occupation

(b.)作法：

將user, movie, user sex, user occupation做dim=64的embedding

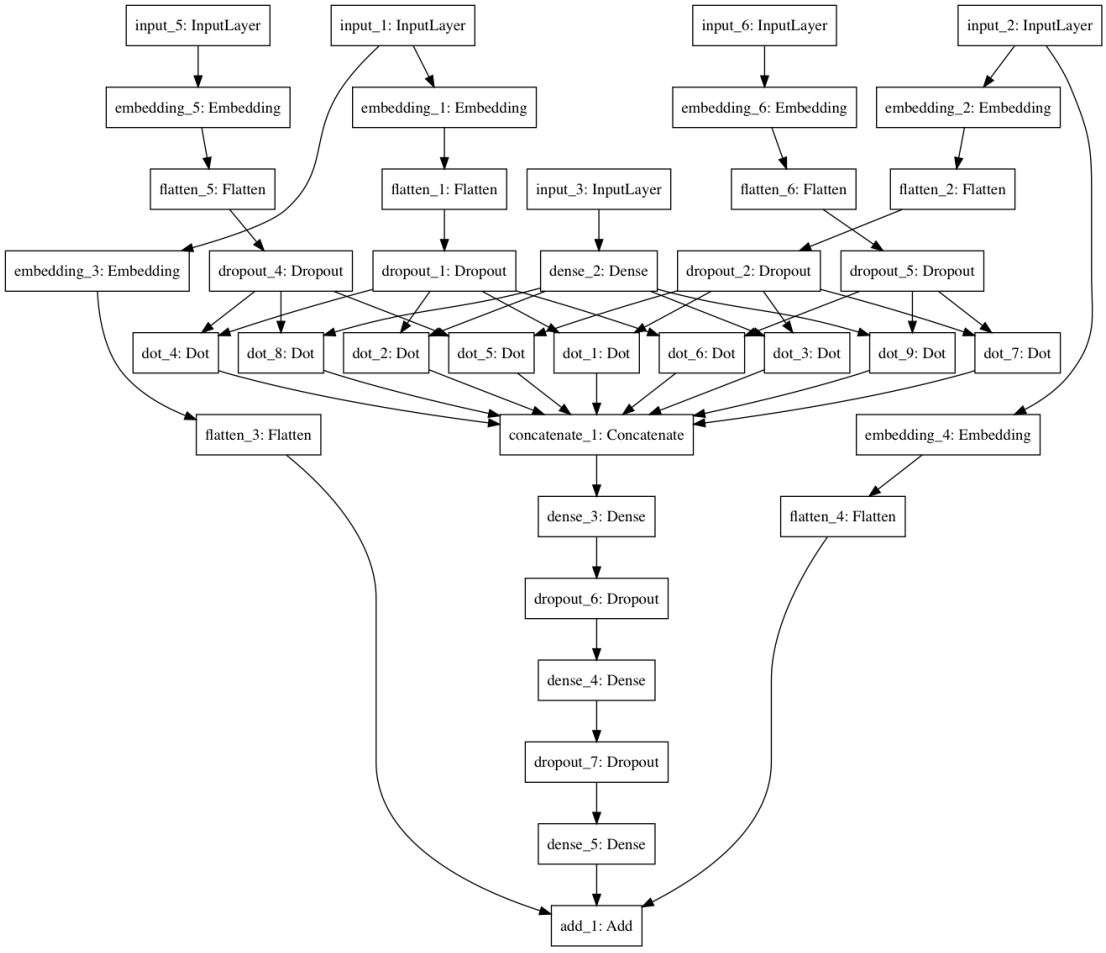
movie category則是用透過dense的方式輸出成一個dim=64的tensor

把user, movie, movie category與user sex, user occupation相互dot出6個tensor

user, movie, movie category內部相互dot出3個tensor

concatenate成9個tensor並做DNN

(c.)架構：



(d.)結果：

kaggle public score:0.85378 private score:0.85653