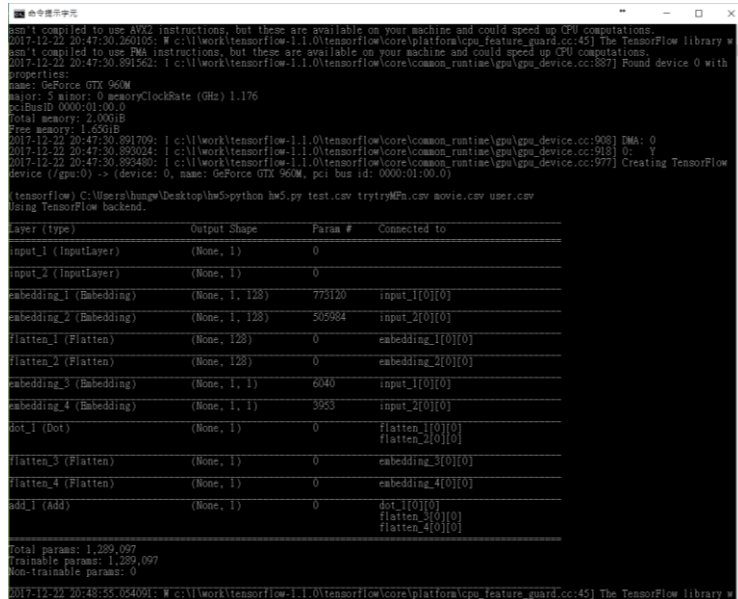


(Q1,Q2,Q3,Q5 都限定使用 MF 的 model)

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



```
bash compiled to use AVX instructions, but these are available on your machine and could speed up CPU computations.
2017-12-22 20:47:30.260105: W c:\work\tensorflow-1.1.0\tensorflow\core\platform\cpu_feature_guard.cc:45] The TensorFlow library w
bash compiled to use FMA instructions, but these are available on your machine and could speed up CPU computations.
2017-12-22 20:47:30.891562: I c:\work\tensorflow-1.1.0\tensorflow\core\common_runtime\gpu\gpu_device.cc:887] Found device 0 with
Properties:
name: GeForce GTX 960M
major: 5 minor: 0 memoryClockRate (GHz) 1.176
pciBusID 0000:01:00:0
Free memory: 1.6501B
2017-12-22 20:47:30.891709: I c:\work\tensorflow-1.1.0\tensorflow\core\common_runtime\gpu\gpu_device.cc:908] DMA: 0
2017-12-22 20:47:30.893024: I c:\work\tensorflow-1.1.0\tensorflow\core\common_runtime\gpu\gpu_device.cc:918] 0: Y
2017-12-22 20:47:30.893460: I c:\work\tensorflow-1.1.0\tensorflow\core\common_runtime\gpu\gpu_device.cc:977] Creating TensorFlow
Device (/gpu:0) => device: 0, name: GeForce GTX 960M, pci bus id: 0000:01:00:01
(tensorflow) C:\Users\hungs\Desktop\hw5>python hw5.py test.csv tryMFa.csv movie.csv user.csv
Using TensorFlow backend.
```

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 1)	0	
input_2 (InputLayer)	(None, 1)	0	
embedding_1 (Embedding)	(None, 1, 128)	773120	input_1[0][0]
embedding_2 (Embedding)	(None, 1, 128)	505984	input_2[0][0]
flatten_1 (Flatten)	(None, 128)	0	embedding_1[0][0]
flatten_2 (Flatten)	(None, 128)	0	embedding_2[0][0]
embedding_3 (Embedding)	(None, 1, 1)	6640	input_1[0][0]
embedding_4 (Embedding)	(None, 1, 1)	3953	input_2[0][0]
dot_1 (Dot)	(None, 1)	0	flatten_1[0][0] flatten_2[0][0]
flatten_3 (Flatten)	(None, 1)	0	embedding_3[0][0]
flatten_4 (Flatten)	(None, 1)	0	embedding_4[0][0]
add_1 (Add)	(None, 1)	0	dot_1[0][0] flatten_3[0][0] flatten_4[0][0]

```
Total params: 1,289,097
Trainable params: 1,289,097
Non-trainable params: 0
2017-12-22 20:48:35.054098: W c:\work\tensorflow-1.1.0\tensorflow\core\platform\cpu_feature_guard.cc:45] The TensorFlow library w
```

normalize ,kaggle_rmse = 0.89807

Un-normalize ,kaggle_rmse = 0.86551

我的作法是，採用 scaling，1-5，先同減一，再除以 4，最後再把他加回去，根據 kaggle 上面顯示，沒有做 normalize 的效果比較好!

2. (1%)比較不同的 latent dimension 的結果。

latent_dim = 1024, val_rmse = 0.8863

latent_dim = 128, val_rmse = 0.8654

可能 dim 比較少的情況下，資訊比較不會稀疏，所以效果比較好!

3. (1%)比較有無 bias 的結果。

有加上 bias 項，kaggle_rmse = 0.86551

沒有加上，kaggle_rmse = 0.86454

我覺得沒有顯著的差異，可能是因為，我沒有跑太多次，但是有加 bias 的收斂比較快!

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。

MF, kaggle_rmse = 0.86454

DNN, kaggle_rmse = 0.83796

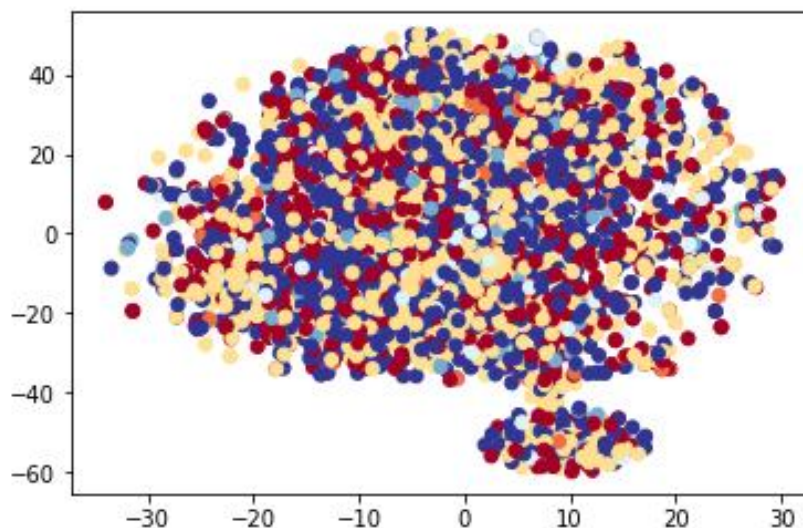
我在 DNN 上面除了 MF 的資訊又加了 2 個 embedding，而且還能使用 drop、hidden layer 等工具，覺得 DNN 在使用上比較靈活!

DNN_Model

tensorflow C:\Users\hung\Desktop\hw5>python hw5_test.py
Using TensorFlow backend.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 1)	0	
input_2 (InputLayer)	(None, 1)	0	
embedding_1 (Embedding)	(None, 1, 1024)	6184960	input_1[0][0]
embedding_2 (Embedding)	(None, 1, 1024)	4047872	input_2[0][0]
flatten_1 (Flatten)	(None, 1024)	0	embedding_1[0][0]
flatten_2 (Flatten)	(None, 1024)	0	embedding_2[0][0]
embedding_5 (Embedding)	(None, 1, 1)	6040	input_1[0][0]
embedding_6 (Embedding)	(None, 1, 1)	3953	input_2[0][0]
embedding_3 (Embedding)	(None, 1, 1024)	6184960	input_1[0][0]
embedding_4 (Embedding)	(None, 1, 1024)	4047872	input_2[0][0]
dot_1 (Dot)	(None, 1)	0	flatten_1[0][0] flatten_2[0][0]
flatten_5 (Flatten)	(None, 1)	0	embedding_5[0][0]
flatten_6 (Flatten)	(None, 1)	0	embedding_6[0][0]
flatten_3 (Flatten)	(None, 1024)	0	embedding_3[0][0]
flatten_4 (Flatten)	(None, 1024)	0	embedding_4[0][0]
add_1 (Add)	(None, 1)	0	dot_1[0][0] flatten_5[0][0] flatten_6[0][0]
concatenate_1 (Concatenate)	(None, 2049)	0	flatten_3[0][0] flatten_4[0][0] add_1[0][0]
dense_1 (Dense)	(None, 64)	131200	concatenate_1[0][0]
dropout_1 (Dropout)	(None, 64)	0	dense_1[0][0]
dense_2 (Dense)	(None, 1)	65	dropout_1[0][0]
Total params: 20,606,922			
Trainable params: 20,606,922			

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



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

我把電影的類別做 one-hot，然後再去加入 DNN 裡面，但是 RMSE，沒有比我原先的 DNN 好。