

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

(collaborator:)

用numpy.mean跟numpy.linalg.norm 算出平均、標準差，再減去平均除以標準差。

使用normalization 準確率：private: 0.85569 public: 0.85440

不使用normalization 準確率：private: 0.85771 public: 0.85748

2. (1%)比較不同的latent dimension的結果。

(collaborator:)

latent dimension=256 準確率：private:0.85683 public:0.85562

latent dimension=128 準確率：private: 0.85771 public: 0.85748

3. (1%)比較有無bias的結果。

(collaborator:)

沒bias 準確率：private:0.86121 public:0.85815

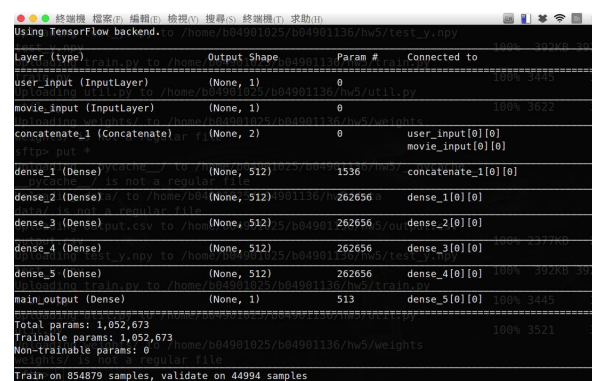
有bias 準確率：private: 0.85771 public: 0.85748

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:)

DNN 準確率：private:1.11899 public:1.11964

MF 準確率：private: 0.85771 public: 0.85748

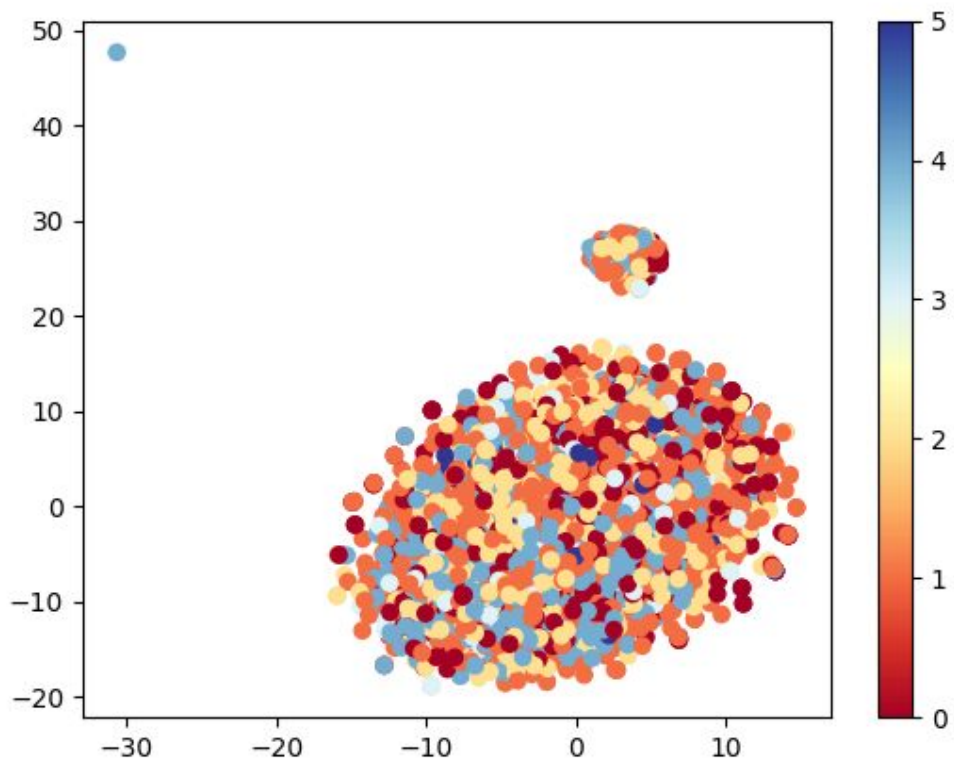


```
Using TensorFlow backend.
Layer (type) Output Shape Param # Connected to
-----
user_input (InputLayer) (None, 1) 0 /home/b04901136/hw5/train.py
movie_input (InputLayer) (None, 1) 0 100% 3622 3
concatenate_1 (Concatenate) (None, 2) 0 user_input[0][0]
movie_input[0][0]
dense_1 (Dense) (None, 512) 1536 concatenate_1[0][0]
dense_2 (Dense) (None, 512) 26256 dense_1[0][0]
dense_3 (Dense) (None, 512) 26256 dense_2[0][0]
dense_4 (Dense) (None, 512) 26256 dense_3[0][0]
dense_5 (Dense) (None, 512) 26256 dense_4[0][0]
main_output (Dense) (None, 1) 513 dense_5[0][0]
Total params: 1,052,673
Trainable params: 1,052,673
Non-trainable params: 0
Train on 854879 samples, validate on 44994 samples
```

因為DNN中可能參數太多，需要更多的training data 才可以有比較好表現。

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

(collaborator:)



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:)

我把age、gender、都拿來考慮, 直接跟dot後的結果concat在一起然後在直接DNN, 可是這樣做之後準確率完全沒有上升。

準確率: private:0.86927 public:0.86235