**Machine Learning 2017 HW6 Report**

*Matrix Factorization*

B03901156 Yu Xuan Huang

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

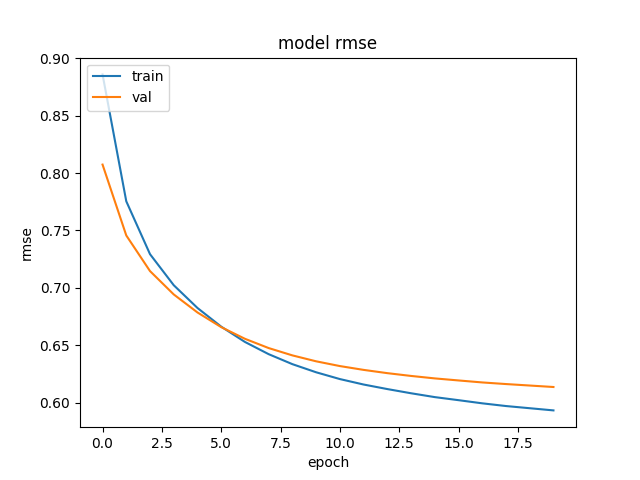
* How to normalize

以 matrix factorization的model為例，首先利用numpy.mean 及 numpy.std 算出training data 中rating 的平均值及標準差；再利用以下公式將training data 的rating 做normalization:

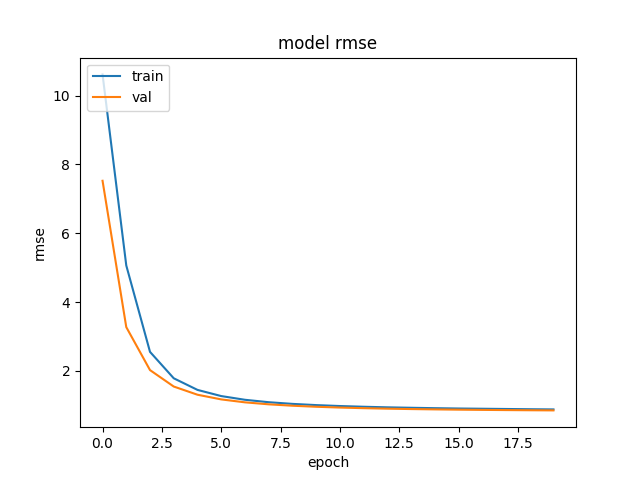
而在最後predict testing data時，需將所predict出的值以下列公式還原成原先的rating值:

* Difference of two models (model parameters: epochs = 20, validation\_split = 0.1, batch\_size = 128, latent dimension = 5)

Implementation of normalization: (Kaggle rmse=**0.87801**)



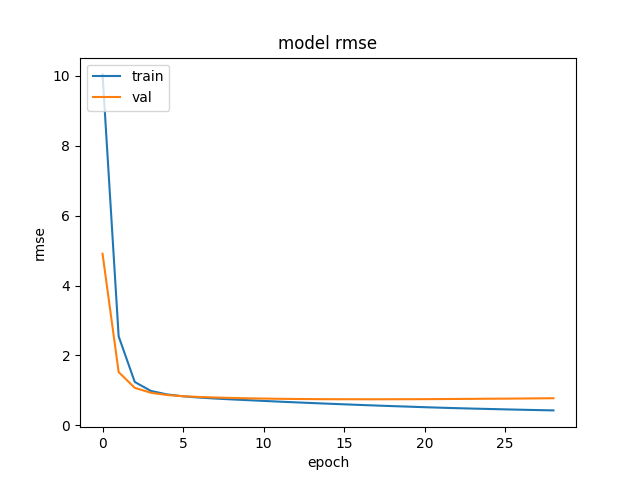
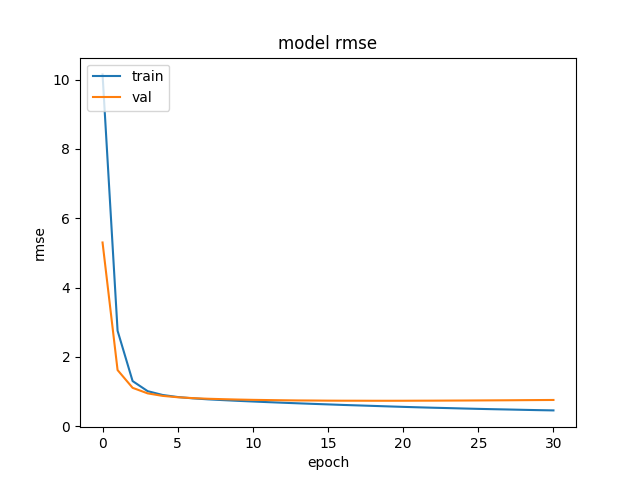
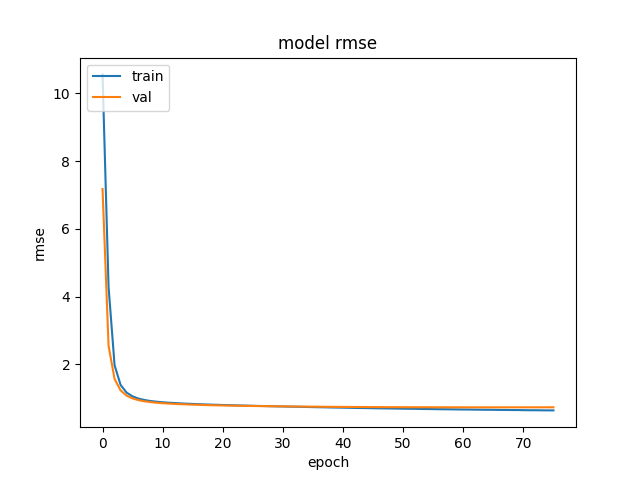
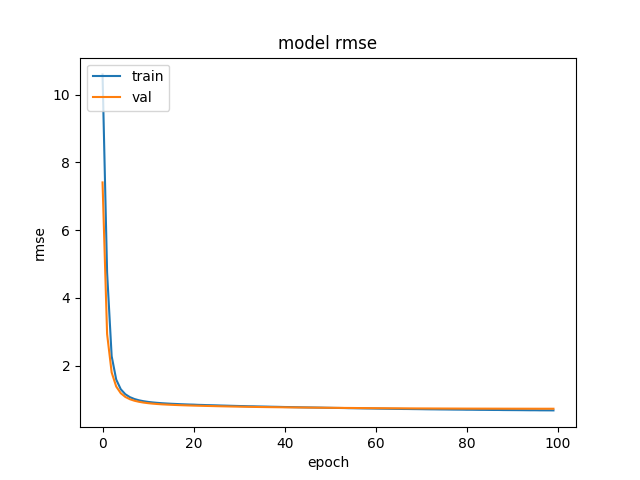
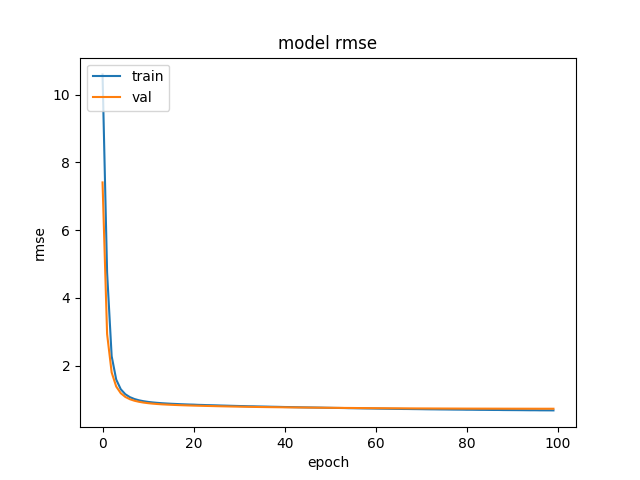
Implementation of not performing normalization: (Kaggle rmse=**0.92957**)



由上列結果觀察可知: rating normalization 可加速matrix factorization model的rmse loss收斂速率，在相同epoch數的training 下，能獲得較好的model，kaggle上分數也較為優越。

1. 比較不同的latent dimension的結果。

(model parameters: epochs = 100, validation\_split = 0.1, batch\_size = 128, optimizer = ‘adamax’)



Latent dimension = 120

Latent dimension = 100

Latent dimension = 20

Latent dimension = 10

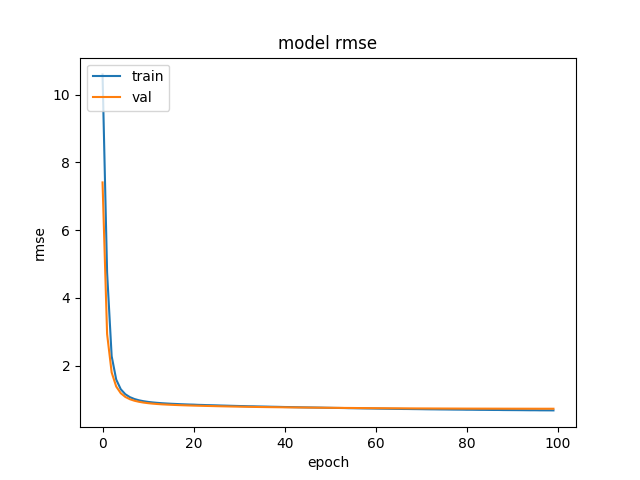
Latent dimension = 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Latent dimension** | 5 | 10 | 20 | 100 | 120 |
| **RMSE** | 0.87093 | 0.85930 | 0.85752 | 0.86663 | 0.86590 |

由上述結果觀察可知: latent dimension越大雖會增加單次epoch下training所需花費的時間，但其訓練epoch數結束較早；而當latent dimension = 5~20時，latent dimension越大所得到的結果(RMSE)較好，但當latent dimension >=20時，latent dimension越大所得結果(RMSE)卻較當latent dimension = 5~20時的model差。因此從實驗數據表中可推論當latent dimension在5~20時，training rate 雖會隨著latent dimension的加大而降低，但RMSE明顯變小，確實對結果能有所改善。

1. 比較有無bias的結果。

(model parameters: epochs = 100, validation\_split = 0.1, batch\_size = 128, latent\_dimension = 5, optimizer = ‘adamax’)



With bias

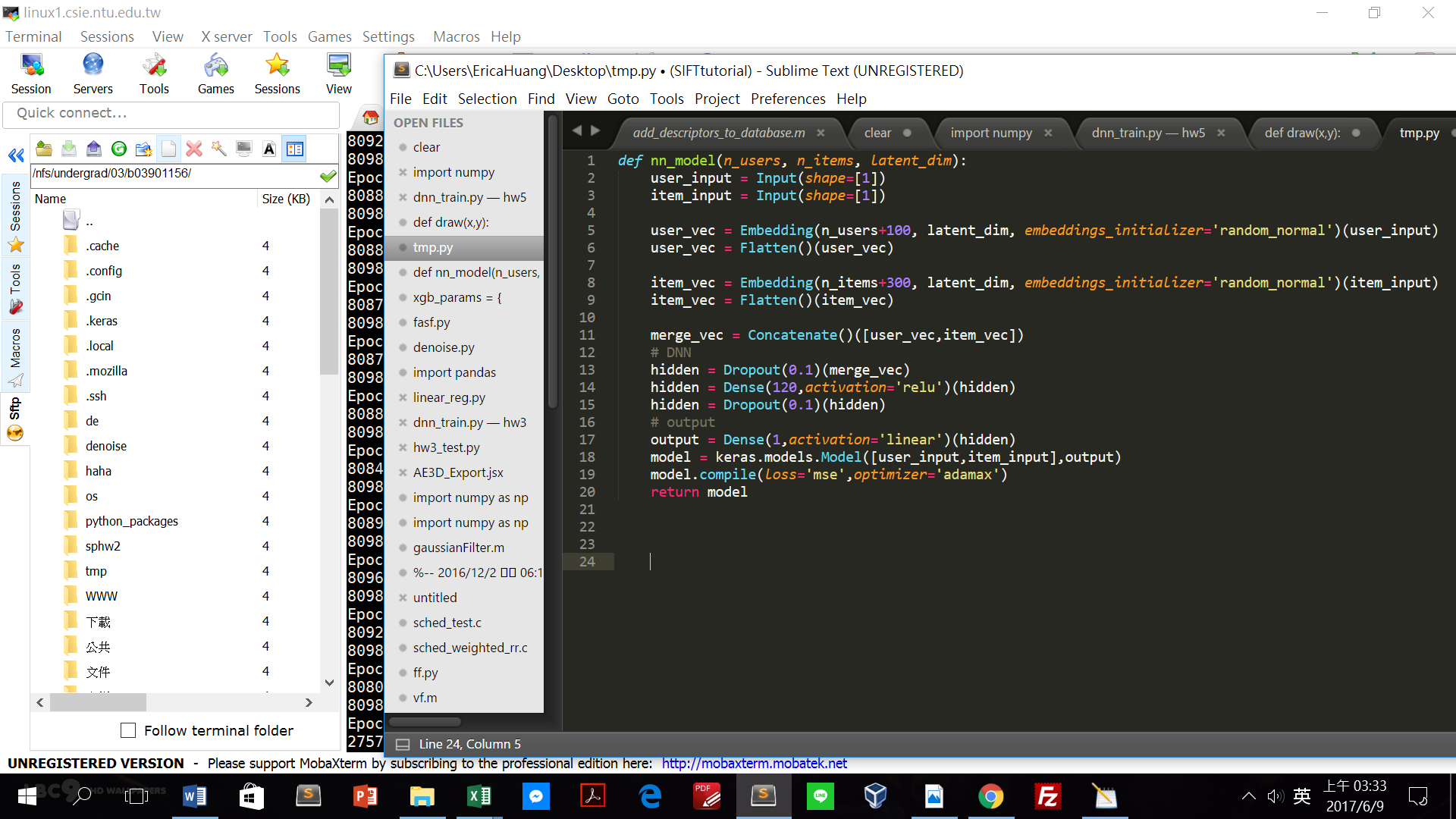
|  |  |
| --- | --- |
| **Matrix Factorization Model** | **RMSE** |
| With bias | 0.87093 |
| Without bias |  |

因每個user 可能都會有各自評分的取向，例如傾向於將每部電影都評得較為高分或較為低分；而相同地，每部電影所得評分也可能具有類似的偏差情形，故為了降低這種現象對於結果的影響，在training 的model 中加入bias項: .

而從上述實驗結果觀察可知: 加入bias所train出的model確實能有效降低RMSE，顯示bias確實能抑制評分傾向過高或過低對test結果可能造成的影響。

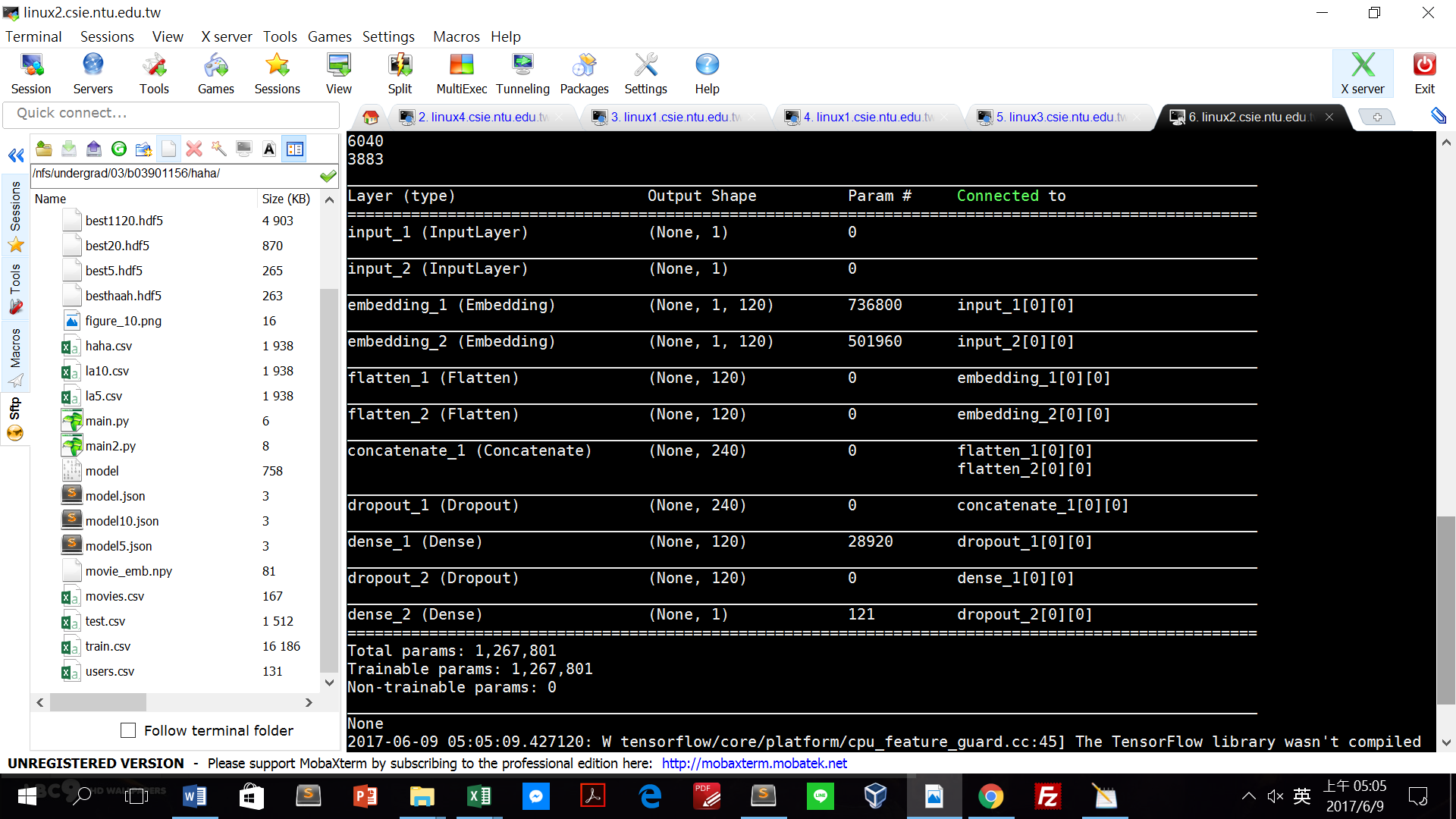
1. 請試著用DNN來解決這個問題，並且說明實做的方法(方法不限)。並比較MF和NN的結果，討論結果的差異。

Implementation:



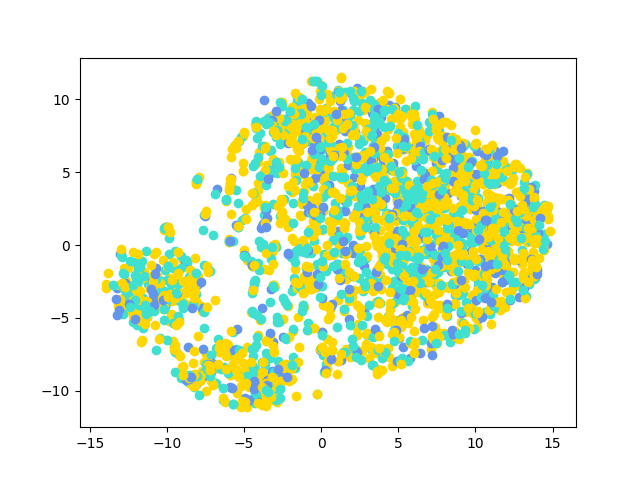
將user embedding 及movie embedding concatenate 在一起後再過DNN得出rating，而output則將其視為regression問題。

Model structure:



|  |  |
| --- | --- |
| **Model** | **RMSE** |
| Matrix Factorization model | 0.86202 |
| DNN model | 0.86663 |

在上述的Matrix Factorization model 中其參數為(epochs = 1000, validation\_split = 0.1, batch\_size = 128, latent\_dimension = 5, optimizer = ‘adamax’)，而DNN model的參數為(epochs = 1000, validation\_split = 0.1, batch\_size = 128, latent\_dimension = 120, optimizer = ‘adamax’)；經實驗後得知Matrix Factorization model所得RMSE較DNN model來得低，但其訓練過程較為緩慢，而DNN model 則有較佳的訓練速度。

1. 請試著將movie的embedding用tsne降維後，將movie category當作label來作圖。  
   
2. 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。