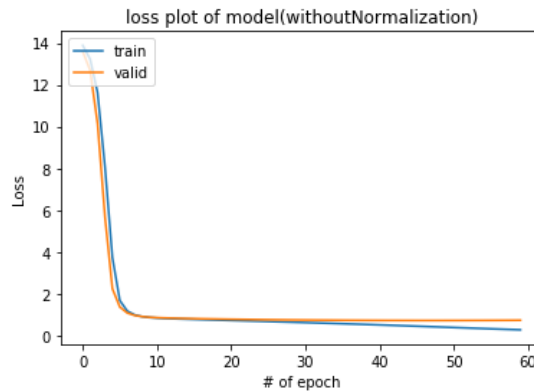


1. 請比較有無 **normalize(rating)** 的差別。並說明如何 **normalize(collaborator:自己)**

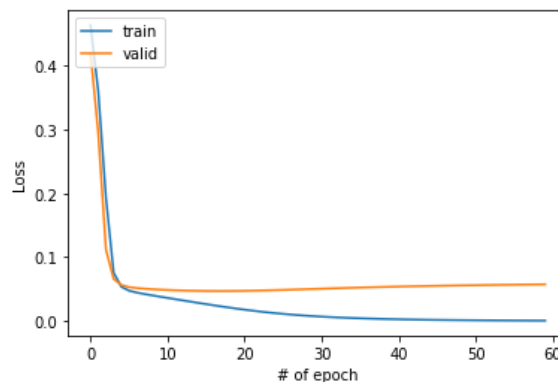
a. 沒有 normalization：



Kaggle RMSE : 0.84979

Val loss : 0.7079

B. 有 normalization：



Kaggle RMSE : 0.87376

Val loss : 0.04700

(上圖是 normalization 的 loss，所以 loss 會比較小)

由上兩圖所示，沒有 normalization 的 Kaggle 分數較好。而我的作法是將 rating normalize 到 [0,1] 之間。先將 rating 扣掉最小值 1，在除上最大(5) – 最小(1)。

2. 比較不同的 **latent dimension** 的結果。(collaborator: 自己)

左圖：

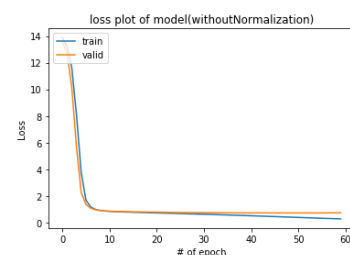
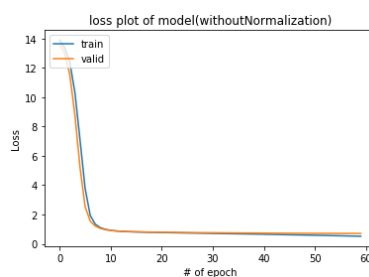
latent_factor = 256

Val loss(mse) : 0.71117

右圖：

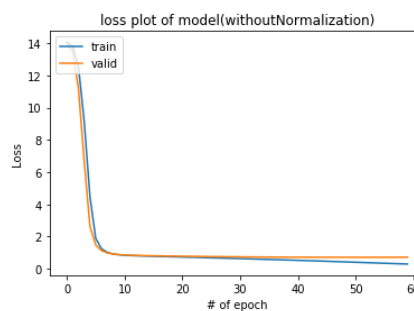
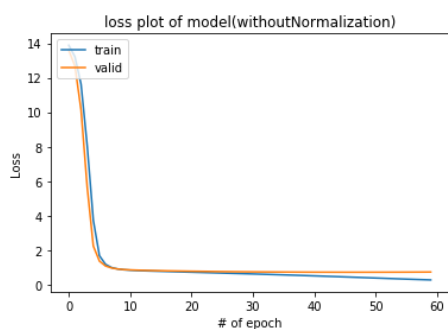
latent_factor = 512

Val loss(mse) : 0.7079



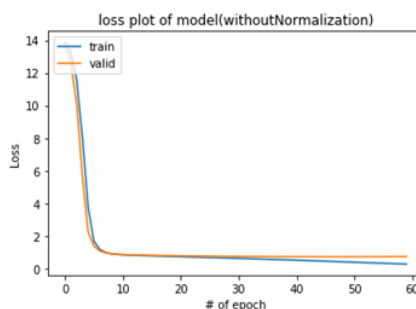
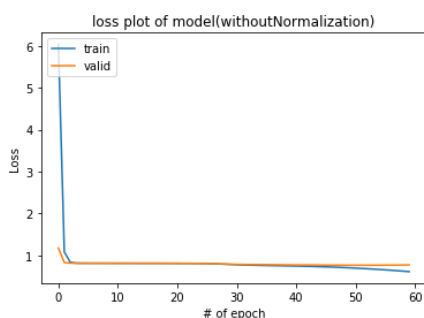
由上述所示，在同樣的 epoch(60)及 batch_size(40000)，latent_factor 較大，有較佳的效果。

3. 比較有無 bias 的結果。(collaborator: 自己)



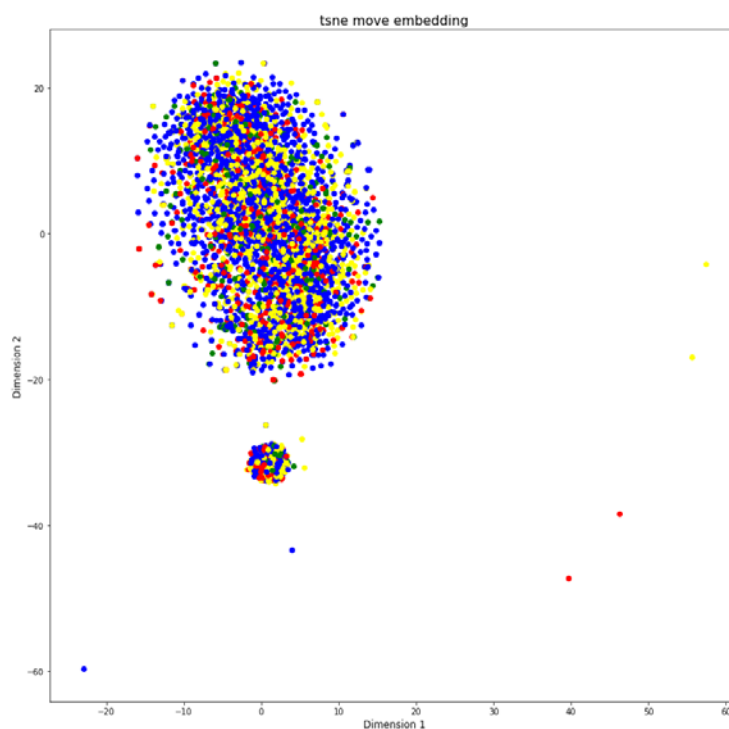
左圖為有加 **bias** 的結果，Val loss：0.7079；右圖為沒加 bias 的結果，Val loss：0.70970，兩者都有相同的 `latent_factor(512)`、`epoch(60)`及 `batch_size(40000)`。兩者由結果來看效果差異並不大。

4. 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: 自己)



左圖為 NN 的結果，Val loss：0.76588；右圖為沒加 bias 的 MF 結果，Val loss：0.7079，兩者都有相同的 `latent_factor(512)`、`epoch(60)`及 `batch_size(40000)`。我的作法是將 `user` 及 `movie` 透過 `concatenate` 再一起，再透過三層的 `Dense` 層，分別是 `unit=256`、`128` 及 `1`。而比較的結果為單純的 MF 比 NN 的效果好，可能是我的參數沒能調的很好，才有比較大的差距。

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



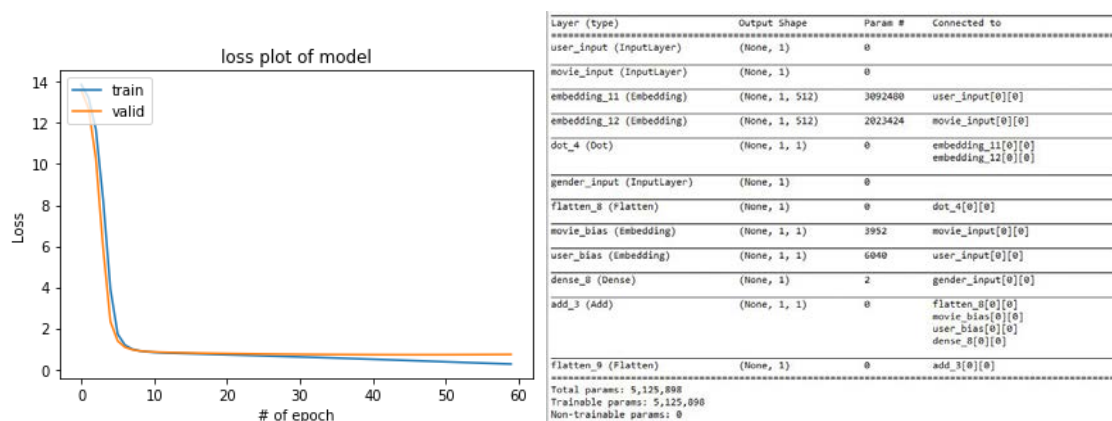
'Thriller'、'Horror'及'Crime'一類: red

'Drama'及'Musical'一類: yellow

'Adventure'、'Animation'及 Children's 一類 : green

其他一類 : blue

6. (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: 自己)



上圖是我加上 user gender 這個 feature 的結果，val loss= 0.71055，latent_factor(512)、epoch(60)及 batch_size(40000)，效果比沒加上這個 feature 的結果還差。我的作法是將 user 的 gender 男生設為 1，女生設為 0，並把 user gender add 到 user embedding 和 movie embedding 中。