

Machine Learning (2017, Fall)

Assignment 5 – Movie Recommendation

學號：b03901086 系級：電機四 姓名：楊正彥

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

我使用的 normalization method 為對 rating 做 standard normalization，得到的 mean = 3.582、std = 1.117，之後在 testing 時再將 predict 的結果乘上標準差並加上平均還原成原來的 rating 值。

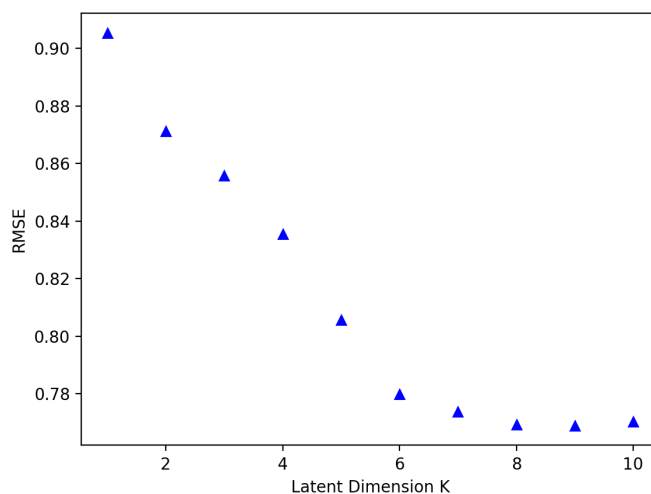
	Val_loss	Private Score	Public Score
w/o Normalization	0.7738	0.86889	0.86573
w/ Normalization	0.7677	0.86742	0.86296

從上表可以看出來對 rating 做了 normalization 之後雖然 rmse 降低了，但是得到的效果相當有限。

2 比較不同的 latent dimension 的結果(1%)

以我一共測試了 latent dimension=1,2,3,...,10 的 matrix factorization，而下表所顯示的 rmse error 值都是在沒有添加 bias 項與 normalization 狀態下以 epoch=200 的參數 train 出來的。可以觀察到當 latent dimension 約在 7,8,9 附近時可以得到相當不錯的 rmse error（最後本次 report 其他題所使用的皆為 latent dimension=10 的 MF）。

Latent dimension	rmse
1	0.9054
2	0.8712
3	0.8657
4	0.8355
5	0.8056
6	0.7798
7	0.7738
8	0.7693
9	0.7689
10	0.7702



3 比較有無 bias 的結果(1%)

我實作了 b_{users} 和 b_{movies} 兩個 bias 的 matrix factorization，但是結果如下表所示只有在 Val_loss 有比較顯著的差別，而在 kaggle 上的表現幾乎是差不多的，我也試著調整其他訓練參數如 epoch、learning rate 等，但是有無 bias 對於 rating 的 prediction 效果都相當有限。

	Val_loss	Private Score	Public Score
w/o Bias	0.7738	0.86889	0.86573
w/ Bias	0.7530	0.86648	0.86518

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。(1%)

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 1)	0	
input_2 (InputLayer)	(None, 1)	0	
embedding_1 (Embedding)	(None, 1, 50)	197650	input_1[0][0]
embedding_2 (Embedding)	(None, 1, 50)	302050	input_2[0][0]
flatten_1 (Flatten)	(None, 50)	0	embedding_1[0][0]
flatten_2 (Flatten)	(None, 50)	0	embedding_2[0][0]
dropout_1 (Dropout)	(None, 50)	0	flatten_1[0][0]
dropout_2 (Dropout)	(None, 50)	0	flatten_2[0][0]
merge_1 (Merge)	(None, 100)	0	dropout_1[0][0] dropout_2[0][0]
dense_1 (Dense)	(None, 512)	51712	merge_1[0][0]
dropout_3 (Dropout)	(None, 512)	0	dense_1[0][0]
batch_normalization_1 (BatchNor	(None, 512)	2048	dropout_3[0][0]
dense_2 (Dense)	(None, 256)	131328	batch_normalization_1[0][0]
dropout_4 (Dropout)	(None, 256)	0	dense_2[0][0]
batch_normalization_2 (BatchNor	(None, 256)	1024	dropout_4[0][0]
dense_3 (Dense)	(None, 128)	32896	batch_normalization_2[0][0]
dropout_5 (Dropout)	(None, 128)	0	dense_3[0][0]
batch_normalization_3 (BatchNor	(None, 128)	512	dropout_5[0][0]
dense_4 (Dense)	(None, 64)	8256	batch_normalization_3[0][0]
dropout_6 (Dropout)	(None, 64)	0	dense_4[0][0]
dense_5 (Dense)	(None, 1)	65	dropout_6[0][0]
Total params: 727,541			
Trainable params: 725,749			
Non-trainable params: 1,792			

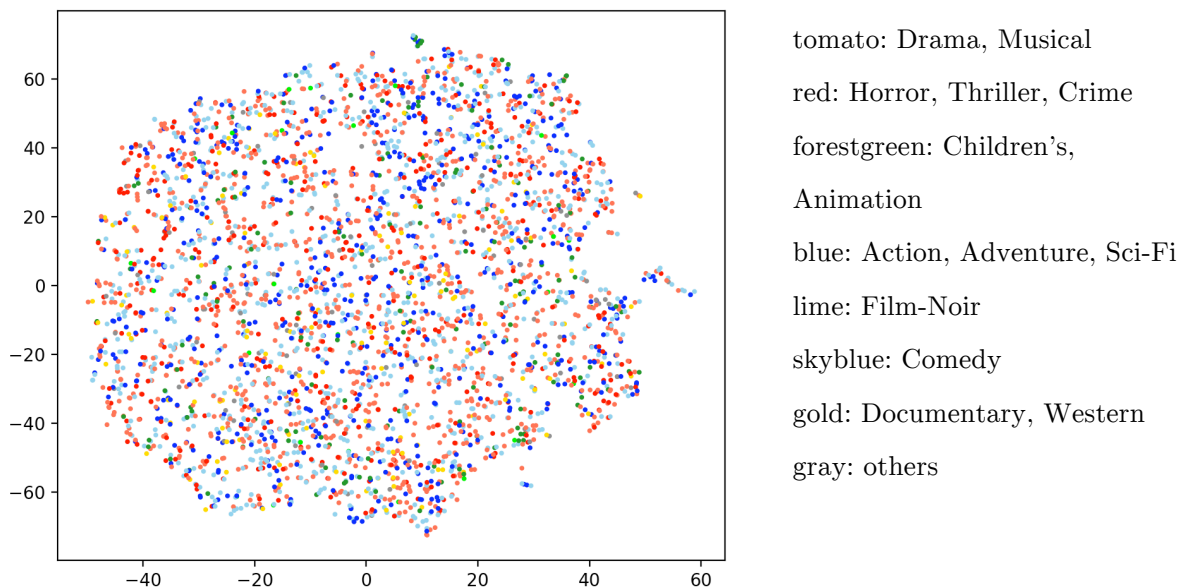
上圖是我這次實作 DNN 的 model 架構，主要可以看成兩個 embedding layers 以及一個相當基本的 deep neural network，其中 embedding layers 先把 users 和 movies 弄成一個 50 維的 embedding vector，再透過 concatenate 後直接丟進 DNN 中去經過多層 dense layers 去 train 出一個單一 output。DNN 模型相關參數為：optimizer=Adam()、learning rate = 0.001、loss=rmse 以及 epoch=200，總共使用了 727,541 個 parameters。

從下表可以看出，DNN 的 performance 表現得比 MF 的 performance 要好的許多，但是其他同學的 MF 表現幾乎都比我的要來得好，所以也有可能純粹是我的 MF 沒有 tune 好參數導致在這個 task 上 DNN 表現比 MF 表現好。

	Val_loss	Private Score	Public Score
Matrix Factorization	0.7738	0.86889	0.86573
DNN	0.7234	0.85431	0.85335

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

由於總共有十八種 movie categories，我先用 tsne 降維做出 18 張以單一類別電影比上其他電影的 scatter plot 來判斷哪些 movie categories 有比較相近的特徵。而分類後實作的圖如下所示：



可以看出來我這次實作的 embedding matrix 用 tsne 降維後看不太出來各個 movie categories 的差別，推測原因可能有以下幾點：

- (1) 實做的 latent dimension 選用太小，無法看出各電影經過 embedding 後的差別。
- (2) 沒有處理單一 movie 多個 labels 的情況。
- (3) movie categories 分群分得不夠好，可能可以參考其他 labels。

BONUS: 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。(1%)

我所嘗試的其他方法皆是基於 Q4 的 DNN 模型下去改的, 主要就是把 users.csv 和 movies.csv 中的某些特定 features 弄成 one-hot encoding 後加進 DNN 的 input vector 中一起 train, 但是如下表所示, 效果其實相當有限, 在 kaggle 上的表現也看不出太大的差異。

	Val_loss	Private Score	Public Score
Matrix Factorization	0.7738	0.86889	0.86573
DNN	0.7234	0.85431	0.85335
DNN w/ users.csv	0.7623	0.86153	0.86079
DNN w/ movies.csv	0.7442	0.85757	0.85660

囿於時間, 沒有再做其他嘗試, 但是回頭想想其實應該把 users 和 movies 之間的關係考慮進去, 例如把這個 users 看過哪些 movies 當成 feature (或者這部 movies 被哪些 users 看過), 用 embedding layer 來表示, 可能對於預測 movie ratings 會有比較顯著的效果。