Q5 Recommendation Systems

Author: Jiyao Wang 20797324

Date: 2021.12.13

For this task, firstly, I get 4 datasets: rating_train, rating_test, movie and user. I use movie and user to get their features to pre-train by matrix factorization. Then I use their pre-trained feature vectors to do a neural collaborative filtering on rating data. I test performance by split train data into train and validation data.

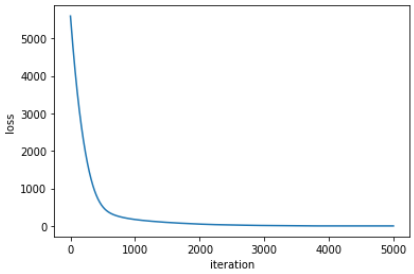
Firstly, for user and movie data, I do prepocessing and feature engineer on them to transfer their descrete features into one-hot which is better for matrix factorization and drop some useless features. Following is the presentation of user data after processing:

ueeza

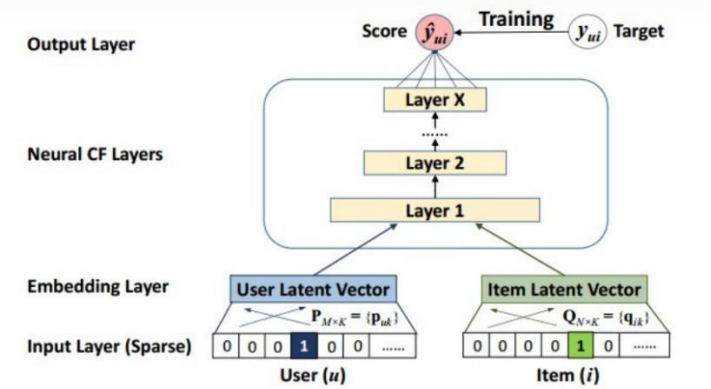
	2_F	2_M	7_1	7_18	7_25	7_35	7_45	7_50	7_56	21_0	21_1	21_2	21_3	21_4	21_5	21_6	21_7	21_8	21_9	21_1
0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
2	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	
4	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
6035	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6036	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	
6037	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	
6038	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	
6039	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	

6040 rows × 30 columns

Then I use pytorch to speed up the matrix factorization on user and movie matrix at the same time, and compute the loss by sum them up. The learning curves is following:



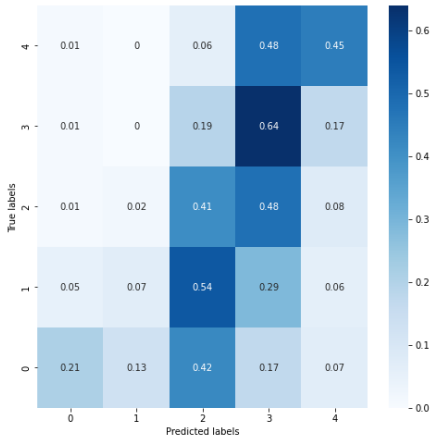
After it, I get two feature matrixes of user and movie. Then I utilize a model named NCF (neural collaborative filtering) which was published on WWW by Dr. He Xiangnan. Compared to traditional CF models, MF is mostly used to model the interaction between user and item, and inner product calculation is used for hidden features of user and item, which is a linear method. The introduction of user/item feature learning by NCF to improve the MF effect also shows that the inner product is not sufficient to capture the complex structure information in the user interaction data.



Following is the structure of my NCF and results:

```
NCF(
  (U): Embedding(6040, 32, padding_idx=6039)
  (V): Embedding(3883, 32, padding_idx=3882)
  (L1): Linear(in_features=64, out_features=64, bias=True)
  (L2): Linear(in_features=64, out_features=5, bias=True)
  (dropout): Dropout(p=0.2, inplace=False)
)
```

	precision	recall	f1-score	support
0	0.50	0.21	0.30	4496
1	0.36	0.07	0.11	8580
2	0.40	0.41	0.41	20984
3	0.45	0.64	0.53	27921
4	0.53	0.45	0.49	18039
accuracy			0.45	80020
macro avg	0.45	0.36	0.37	80020
weighted avg	0.45	0.45	0.43	80020



Meanwhile, to test the affect of pre-train, I do a compare experiment. Firstly, I train NCF after pre-train and test it. Then I train NCF without pre-train and initiate its two embedding layers randomly. Following are the results, left one is pre-trained one and right is without pre-trained, we can see pre-train can speed up the training process.

```
Epoch: 0, loss: 1.27586
Epoch: 5, loss: 1.23613
Epoch: 10, loss: 1.23536
Epoch: 15, loss: 1.23522
Epoch: 20, loss: 1.23547
Epoch: 25, loss: 1.23530
```

```
Epoch: 0, loss: 1.43631
Epoch: 5, loss: 1.23863
Epoch: 10, loss: 1.23652
Epoch: 15, loss: 1.23506
Epoch: 20, loss: 1.23522
Epoch: 25, loss: 1.23532
Epoch: 30, loss: 1.23513
```

Reference:

[1] He X, Liao L, Zhang H, et al. Neural Collaborative Filtering[C]// International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2017:173-182.