

Data Science Practice Assignment 3 report

SID: 12110821

Name: Zhang Chi

Pull Request: [12110821 ZhangChi](#)

Introduction

In this assignment, we are asked to implement three methods introduced in the paper [Neural collaborative filtering](#), including **GMF**, **MLP**, and **NeuMF**. All of models should be evaluated by metrics HR@10 and NDCG@10. Additionally, the study of impact of layer's number on MLP should be taken.

Experiments

The pytorch version I used is 2.3.0, which is the latest version at the time of writing this report and it can be checked by running the following code:

```
1 import torch
2 torch.__version__ # '2.3.0a0+6ddf5cf85e.nv24.04'
```

The data used for this assignment is the **MovieLens**. By default, the learning rate is set to 0.001, the batch size is set to 256, and the number of epochs is set to 40. Moreover, the number of negative samples I trained for each positive sample is 4 uniformly.

To start a training session, I use a function `model_train()` to wrap the whole process so that I can only input `model_config` by a dictionary. Here is an example:

```
1 model_config = {
2     "model_mark": "mlp(mlp_layer=2)", # A string to identify model when saving
    training parameter
3     # 'embedding_dim_mf': 8, # only for GMF and NeuMF
4     "mlp_layers(X)" : 3,
5     'mlp_layers': [64, 32, 16, 8],
6     'model_type': 'MLP' # MLP, NeuMF
7 }
8 model_train(model_config, seed = 42, num_of_negatives=4, num_of_epochs=40)
```

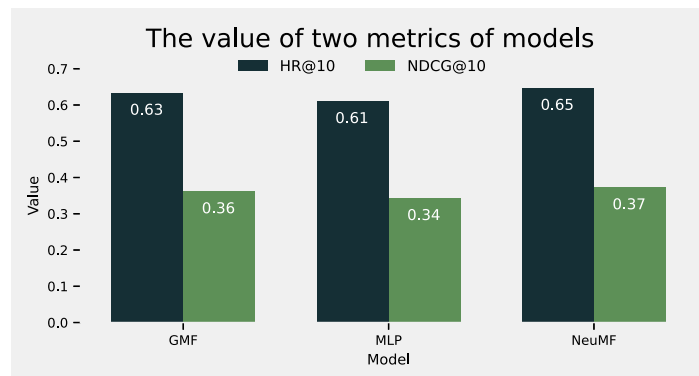
To ensure the reproducibility of my results, I set a `seed` parameter. And during all the training, I used 42 as my seed.

Results

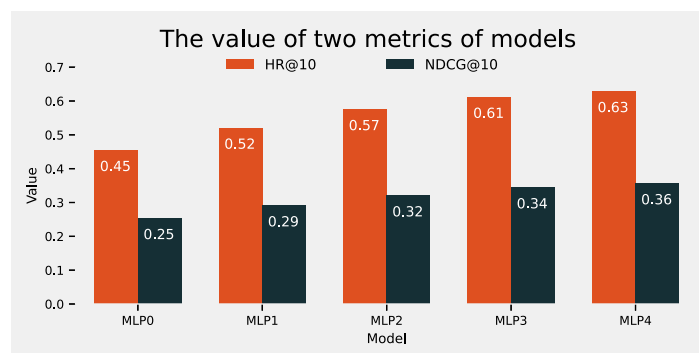
For the first task, I trained 3 models, and configurations were set as follows:

- GMF: embedding size 8
- MLP: embedding size 32, hidden layer size 3.
- NeuMF: embedding size 32, hidden layer size 3.

As we can see, **the performance of NeuMF is better than GMF and MLP**. All models' values of HR@10 is higher than 0.6 while values of NDCG@10 are higher than 0.3, and the best result is 0.65 and 0.37 for HR@10 and NDCG@10, respectively.



For the second task, I train MLP with different hidden layer size, and the results are shown in the following figure.



Based on the information provided, with the number of layers increasing, the model's performance is improving. However, when the number of layers is larger than 3, the model's performance is not improving much anymore.

References

1. 推荐系统论文回顾：神经协同过滤理解与实现-CSDN博客
2. 神经协同过滤NCF|PyTorch实战篇 - 知乎 (zhihu.com)
3. 神经协同过滤NCF(二)之代码实战篇 - 简书 (jianshu.com)
4. neural-collaborative-filtering/src/data.py at master · yihong-chen/neural-collaborative-filtering (github.com)