NEURAL COLLABORATIVE FILTERING

Research Project

DADS6003
Applied Machine Learning

PROJECT SUMMARY

01

Overview

This section outlines the context of the assigned research paper., the necessary steps, and the goals of the research.

02

Methodology

Explain mathematical models used in the paper to achieve collaborative filtering of given datasets and their results.

03

Reproduce Result

Try to replicate the mehtod used in the research paper. Compares the results.

04

Suggestion

Suggest ways to improve the model for better performance.

01 OVERVIEW

arXiv:1708.05031v2 [cs.IR] 26 Aug 2017

Neural Collaborative Filtering*

Xiangnan He
National University of Singapore, Singapore
xiangnanhe@gmail.com

Lizi Liao
National University of Singapore, Singapore
liaolizi.llz@gmail.com

Hanwang Zhang
Columbia University
USA
hanwangzhang@gmail.com

Liqiang Nie
Shandong University
China
nieliqiang@gmail.com

Xia Hu
Texas A&M University
USA
hu@cse.tamu.edu

Tat-Seng Chua
National University of Singapore, Singapore
dcscts@nus.edu.sg

ABSTRACT

In recent years, deep neural networks have yielded immense success on speech recognition, computer vision and natural language processing. However, the exploration of deep neural networks on recommender systems has received relatively less scrutiny. In this work, we strive to develop techniques based on neural networks to tackle the key problem in recommendation — collaborative filtering — on the basis of implicit feedback.

Although some recent work has employed deep learning for recommendation, they primarily used it to model auxiliary information, such as textual descriptions of items and acoustic features of musics. When it comes to model the key factor in collaborative filtering — the interaction between user and item features, they still resorted to matrix factorization and applied an inner product on the latent features of users and items.

By replacing the inner product with a neural architecture that can learn an arbitrary function from data, we present a general framework named NCF, short for *Neural network-based Collaborative Filtering*. NCF is generic and can express and generalize matrix factorization under its framework. To supercharge NCF modelling with non-linearities, we propose to leverage a multi-layer perceptron to learn the user–item interaction function. Extensive experiments on two real-world datasets show significant improvements of our proposed NCF framework over the state-of-the-art methods. Empirical evidence shows that using deeper layers of neural networks offers better recommendation performance.

Keywords

Collaborative Filtering, Neural Networks, Deep Learning, Matrix Factorization, Implicit Feedback

*NExT research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its IRC@SG Funding Initiative.

WWW 2017, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052569

1. INTRODUCTION

In the era of information explosion, recommender systems play a pivotal role in alleviating information overload, having been widely adopted by many online services, including E-commerce, online news and social media sites. The key to a personalized recommender system is in modelling users' preference on items based on their past interactions (*e.g.*, ratings and clicks), known as collaborative filtering [31, 46]. Among the various collaborative filtering techniques, matrix factorization (MF) [14, 21] is the most popular one, which projects users and items into a shared latent space, using a vector of latent features to represent a user or an item. Thereafter a user's interaction on an item is modelled as the inner product of their latent vectors.

Popularized by the Netflix Prize, MF has become the *de facto* approach to latent factor model-based recommendation. Much research effort has been devoted to enhancing MF, such as integrating it with neighbor-based models [21], combining it with topic models of item content [38], and extending it to factorization machines [26] for a generic modelling of features. Despite the effectiveness of MF for collaborative filtering, it is well-known that its performance can be hindered by the simple choice of the interaction function — inner product. For example, for the task of rating prediction on explicit feedback, it is well known that the performance of the MF model can be improved by incorporating user and item bias terms into the interaction function[1]. While it seems to be just a trivial tweak for the inner product operator [14], it points to the positive effect of designing a better, dedicated interaction function for modelling the latent feature interactions between users and items. The inner product, which simply combines the multiplication of latent features linearly, may not be sufficient to capture the complex structure of user interaction data.

This paper explores the use of deep neural networks for learning the interaction function from data, rather than a handcraft that has been done by many previous work [18, 21]. The neural network has been proven to be capable of approximating any continuous function [17], and more recently deep neural networks (DNNs) have been found to be effective in several domains, ranging from computer vision, speech recognition, to text processing [5, 10, 15, 47]. However, there is relatively little work on employing DNNs for recommendation in contrast to the vast amount of literature

[1] http://alex.smola.org/teaching/berkeley2012/slides/8_Recommender.pdf

The paper presents an approach to collaborative filtering (CF) using neural networks, addressing limitations in traditional methods like Matrix Factorization (MF). Neural Collaborative Filtering (NCF) introduces a flexible framework that leverages neural architectures to learn these interactions, providing improved recommendation performance.

WORK WITH DATA

movielens

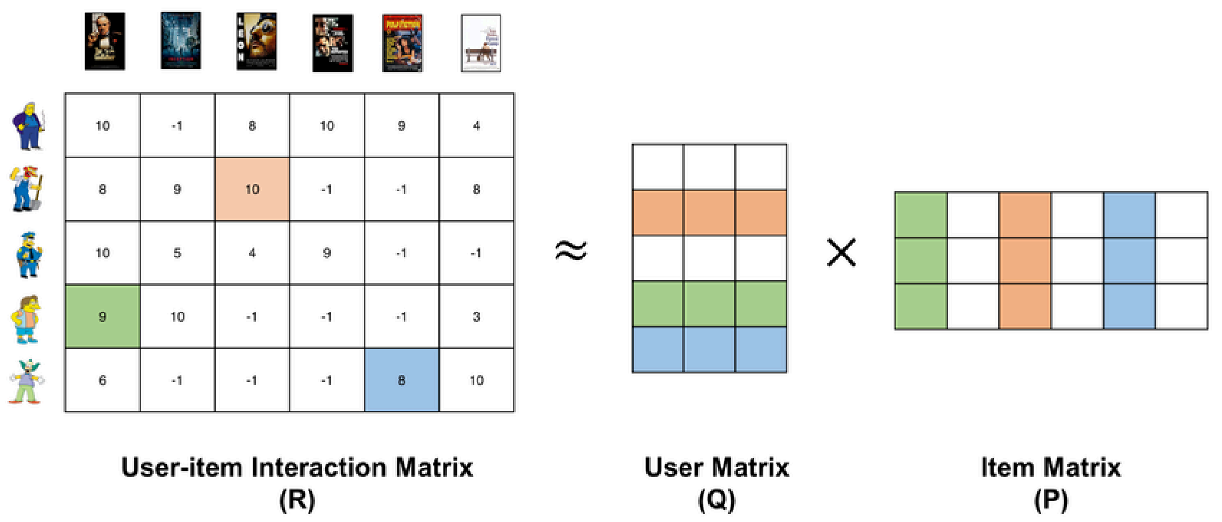
Non-commercial, personalized movie recommendations.

Pinterest

02 METHODOLOGY

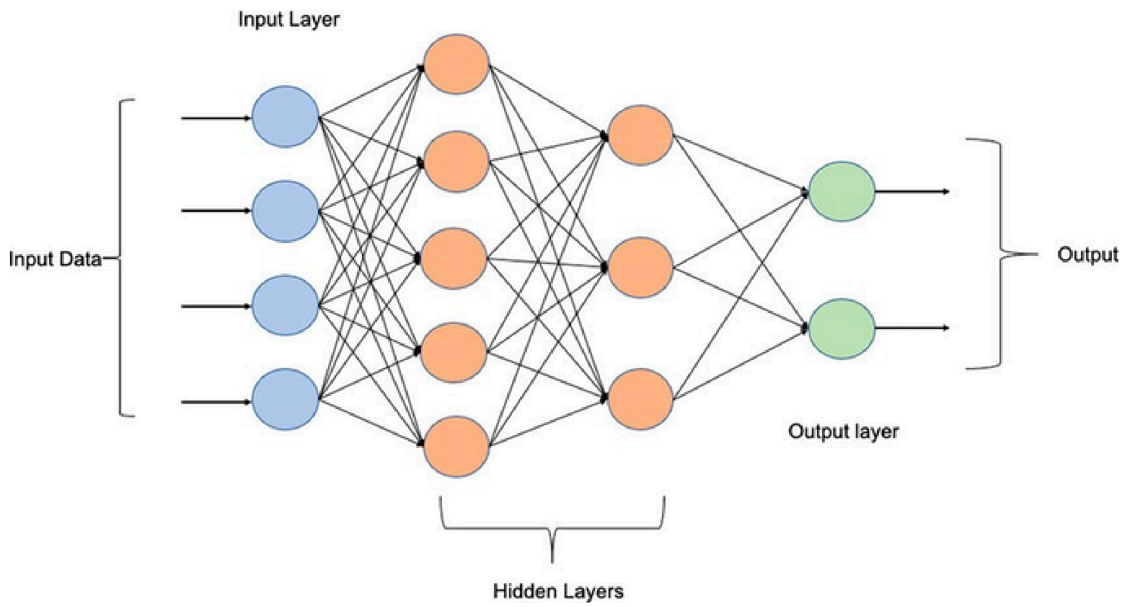
Model Training

Neural matrix factorization model (NeuMF)



Matrix Factorization

A class of collaborative filtering algorithms used in recommender systems. Matrix factorization algorithms work by decomposing the user-item interaction matrix into the product of two lower-dimensionality rectangular matrices.



Multilayer-perceptron

a neural network that's made up of multiple layers of neurons that are fully connected and use nonlinear activation functions. MLPs are a type of artificial neural network that is commonly used in machine learning for tasks like classification, regression, and pattern recognition.

user vector

0	0	0	1	0	...

item vector

0	0	1	0	0	...

02 METHODOLOGY

Model Evaluation

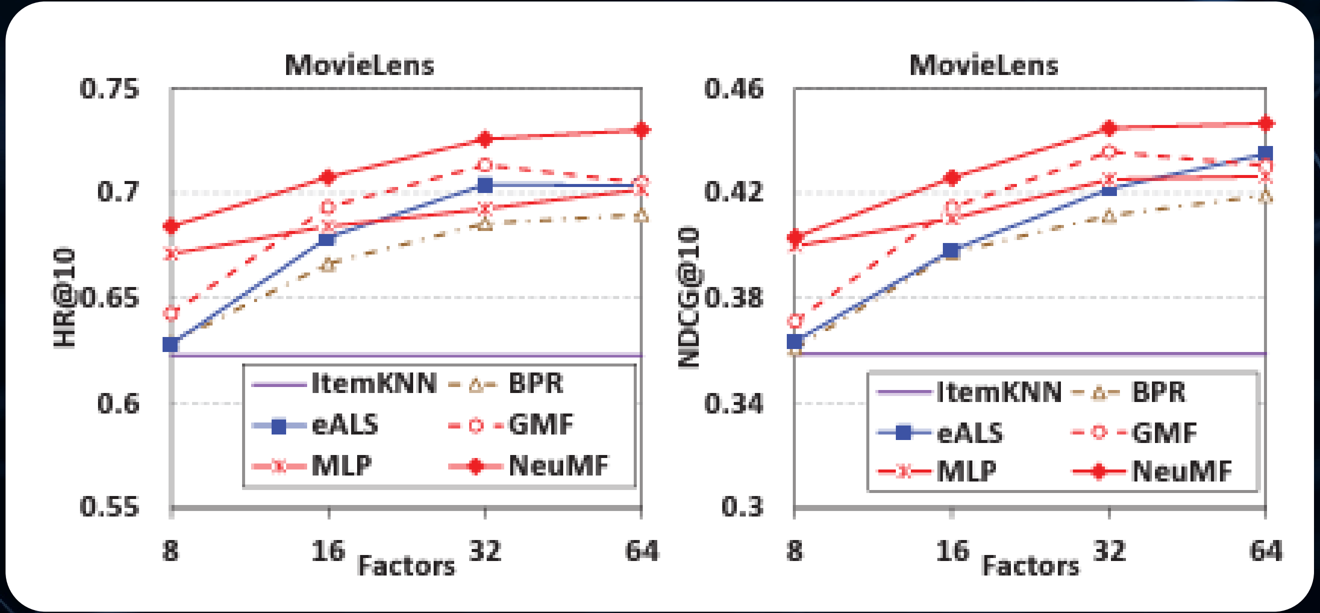
The evaluation of item recommendations was conducted using a leave-one-out strategy, where each user's most recent interaction is held out as the test set, and the remaining data is used for training. Due to the high computational cost of ranking all items for each user, 100 random items, not previously interacted with by the user, are sampled for evaluation. The test item is ranked among these 100 items. The performance of the recommendation system is measured using two of following metrics

Hit Ratio (HR)

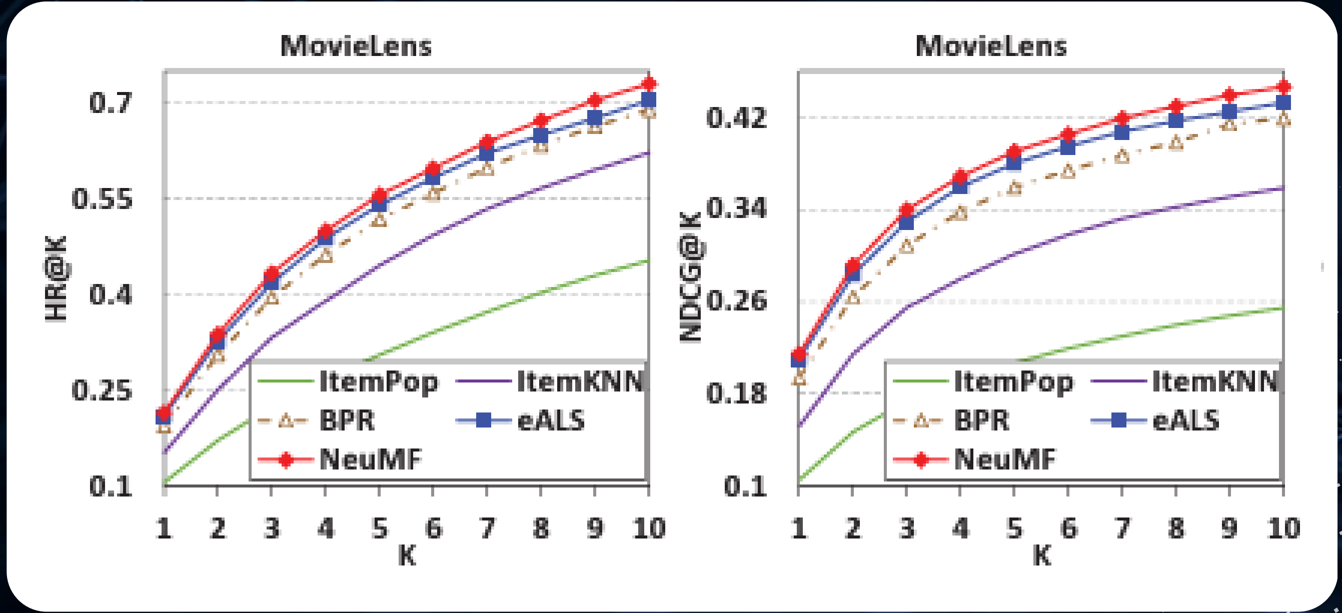
The Hit Ratio measures the proportion of times that the recommended item appears in the top-k list, where k is the number of items recommended. It's a simple binary metric that indicates whether the recommended item is relevant or not.

Normalized Discounted Cumulative Gain (NDCG)

NDCG is a more sophisticated metric that considers both the relevance of the recommended items and their positions in the ranked list.



Performance of Top10 HR (HR@10) and Top10 NDCG (NDCG@10) w.r.t. the number of predictive factors.



Evaluation of Top-K item recommendation where K ranges from 1 to 10

REPRODUCE RESULT

03

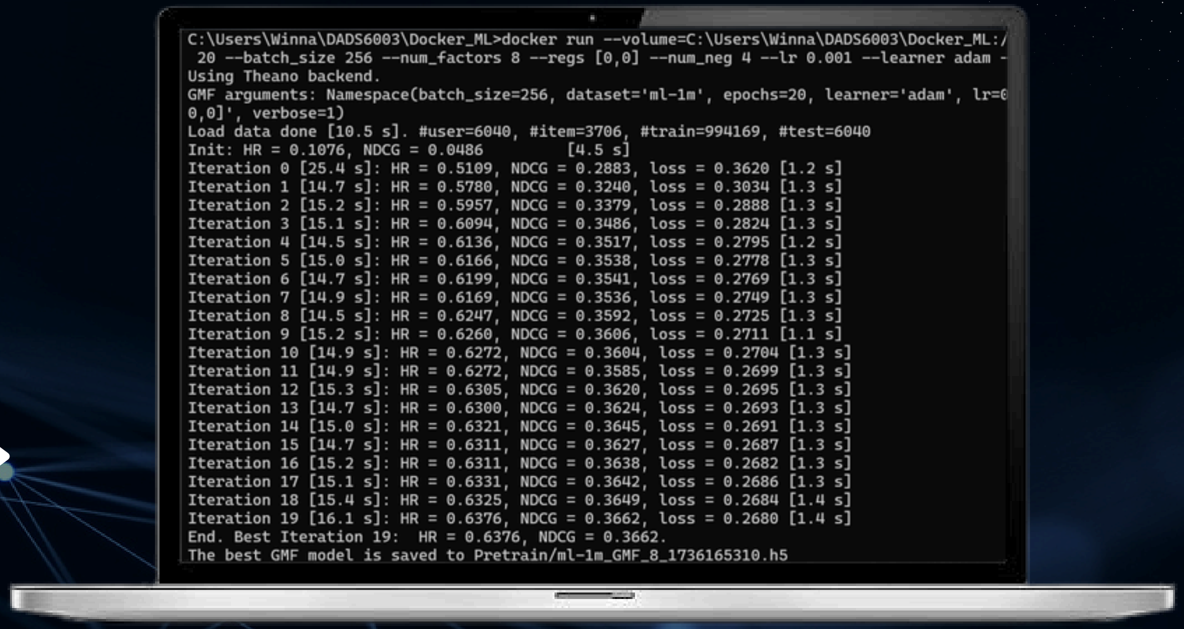
with Movielens datasets

movielens

Non-commercial, personalized movie recommendations.

Use the model from
hexiangnan/neural_collaborative_filtering
Neural Collaborative Filtering

Build Docker image prove to be more time efficient than Google collab



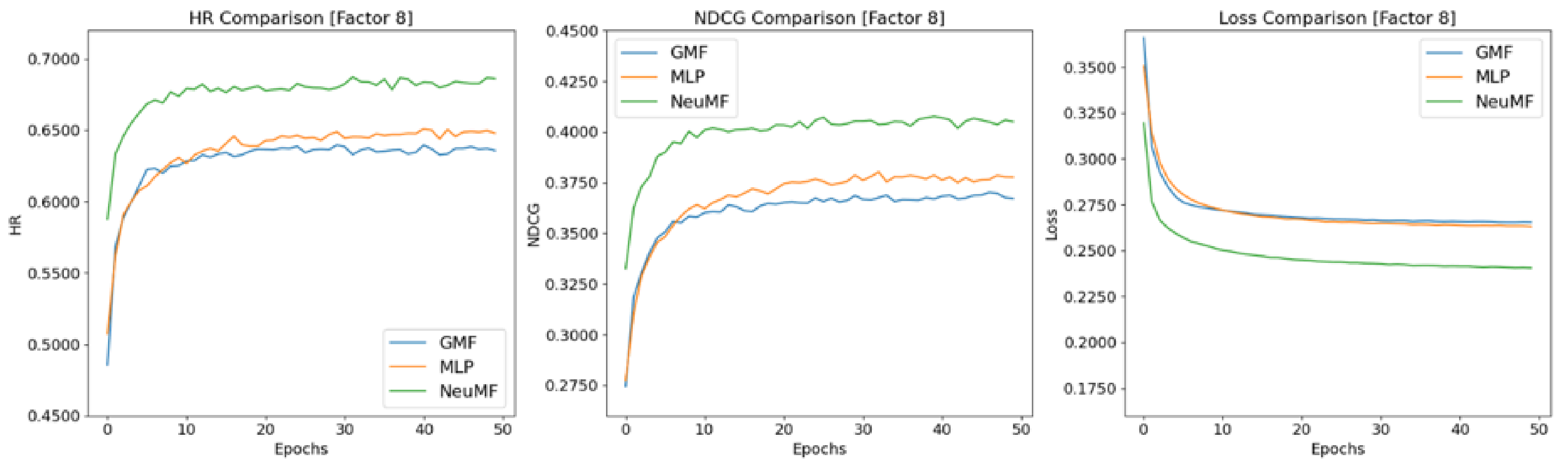
```
C:\Users\Winna\DADS6003\Docker_ML>docker run --volume=C:\Users\Winna\DADS6003\Docker_ML:/
 20 --batch_size 256 --num_factors 8 --regs [0,0] --num_neg 4 --lr 0.001 --learner adam -
Using Theano backend.
GMF arguments: Namespace(batch_size=256, dataset='ml-1m', epochs=20, learner='adam', lr=0
0,0]', verbose=1)
Load data done [10.5 s]. #user=6040, #item=3706, #train=994169, #test=6040
Init: HR = 0.1076, NDCG = 0.0486         [4.5 s]
Iteration 0 [25.4 s]: HR = 0.5109, NDCG = 0.2883, loss = 0.3620 [1.2 s]
Iteration 1 [14.7 s]: HR = 0.5780, NDCG = 0.3240, loss = 0.3034 [1.3 s]
Iteration 2 [15.2 s]: HR = 0.5957, NDCG = 0.3379, loss = 0.2888 [1.3 s]
Iteration 3 [15.1 s]: HR = 0.6094, NDCG = 0.3486, loss = 0.2824 [1.3 s]
Iteration 4 [14.5 s]: HR = 0.6136, NDCG = 0.3517, loss = 0.2795 [1.2 s]
Iteration 5 [15.0 s]: HR = 0.6166, NDCG = 0.3538, loss = 0.2778 [1.3 s]
Iteration 6 [14.7 s]: HR = 0.6199, NDCG = 0.3541, loss = 0.2769 [1.3 s]
Iteration 7 [14.9 s]: HR = 0.6169, NDCG = 0.3536, loss = 0.2749 [1.3 s]
Iteration 8 [14.5 s]: HR = 0.6247, NDCG = 0.3592, loss = 0.2725 [1.3 s]
Iteration 9 [15.2 s]: HR = 0.6260, NDCG = 0.3606, loss = 0.2711 [1.1 s]
Iteration 10 [14.9 s]: HR = 0.6272, NDCG = 0.3604, loss = 0.2704 [1.3 s]
Iteration 11 [14.9 s]: HR = 0.6272, NDCG = 0.3585, loss = 0.2699 [1.3 s]
Iteration 12 [15.3 s]: HR = 0.6305, NDCG = 0.3620, loss = 0.2695 [1.3 s]
Iteration 13 [14.7 s]: HR = 0.6300, NDCG = 0.3624, loss = 0.2693 [1.3 s]
Iteration 14 [15.0 s]: HR = 0.6321, NDCG = 0.3645, loss = 0.2691 [1.3 s]
Iteration 15 [14.7 s]: HR = 0.6311, NDCG = 0.3627, loss = 0.2687 [1.3 s]
Iteration 16 [15.2 s]: HR = 0.6311, NDCG = 0.3638, loss = 0.2682 [1.3 s]
Iteration 17 [15.1 s]: HR = 0.6331, NDCG = 0.3642, loss = 0.2686 [1.3 s]
Iteration 18 [15.4 s]: HR = 0.6325, NDCG = 0.3649, loss = 0.2684 [1.4 s]
Iteration 19 [16.1 s]: HR = 0.6376, NDCG = 0.3662, loss = 0.2680 [1.4 s]
End. Best Iteration 19:  HR = 0.6376, NDCG = 0.3662.
The best GMF model is saved to Pretrain/ml-1m_GMF_8_1736165310.h5
```

Run Docker container by adjusting the parameter and number of iteration based on experiment conducted in research paper

REPRODUCE RESULT

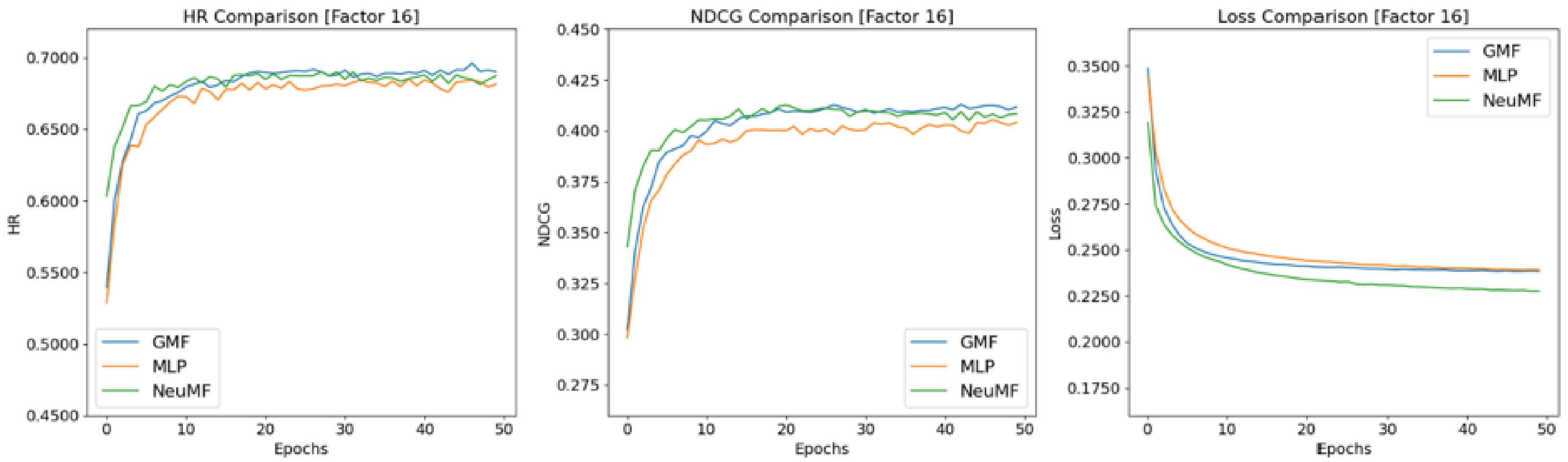
03

Evaluate NeuMF model using HR, NDCG, and Loss comparison for each number of predictive factors (8, 16, 32,64).
All of the results provide the similar trend. As the model reach certain number of epochs (K), the performance tend to not improve any further.



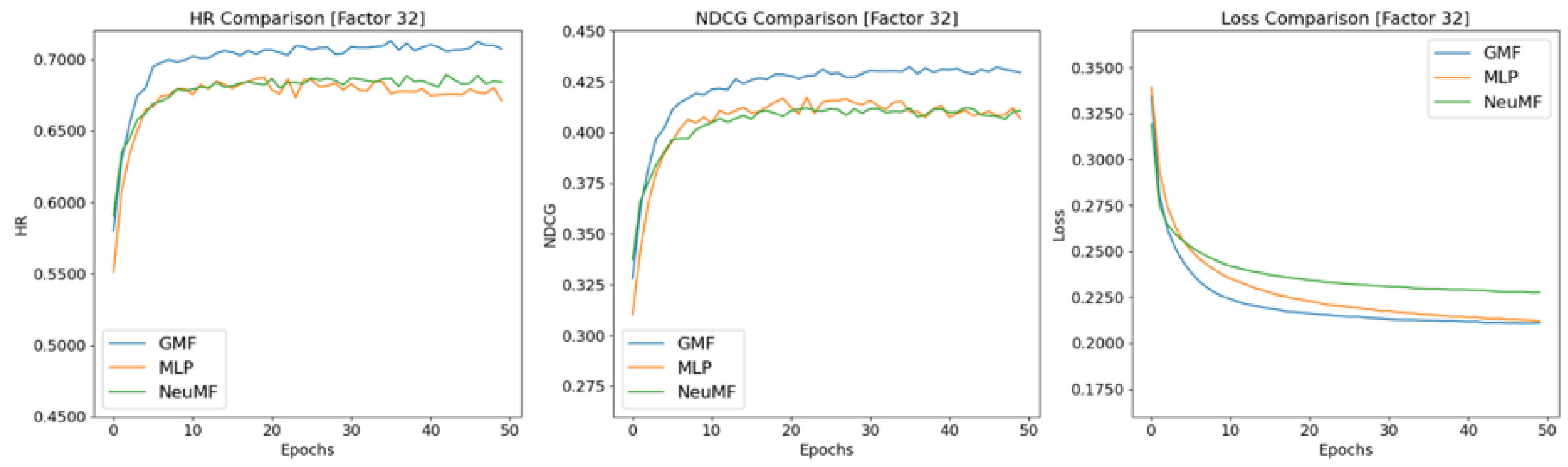
GMF: Best Iteration 29: HR = 0.6397, NDCG = 0.3685.
MLP: Best Iteration 40: HR = 0.6510, NDCG = 0.3762.
NeuMF: Best Iteration 31: HR = 0.6874, NDCG = 0.4056.

factor 8



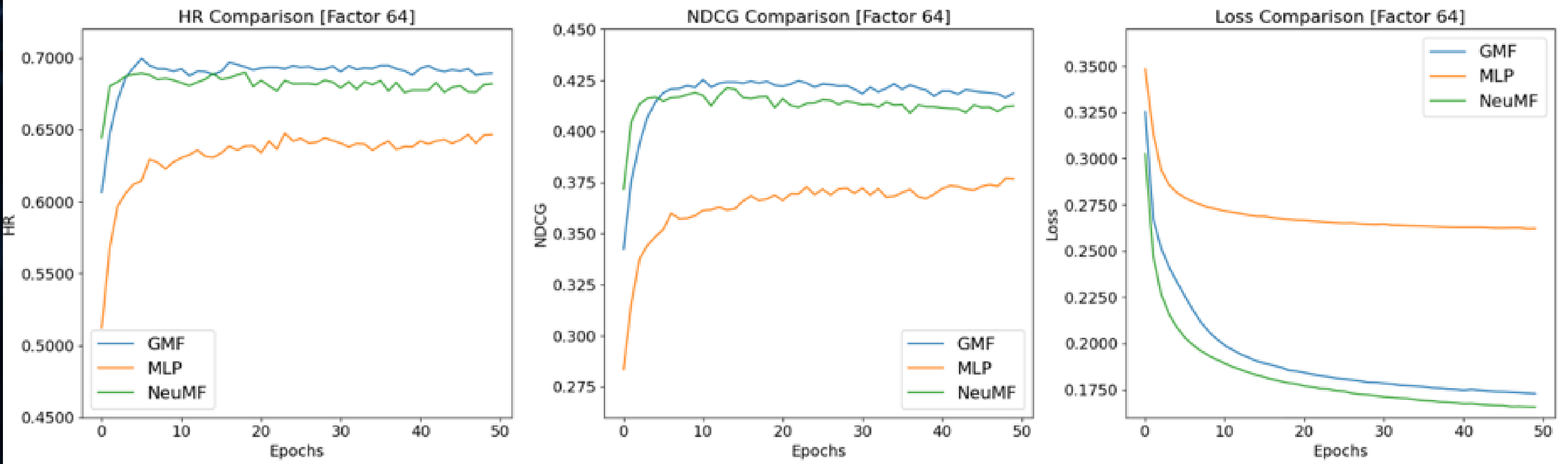
GMF: Best Iteration 46: HR = 0.6962, NDCG = 0.4125.
MLP: Best Iteration 38: HR = 0.6853, NDCG = 0.4031.
NeuMF: Best Iteration 27: HR = 0.6902, NDCG = 0.4103.

factor 16



GMF: Best Iteration 35: HR = 0.7129, NDCG = 0.4323.
MLP: Best Iteration 19: HR = 0.6874, NDCG = 0.4167.
NeuMF: Best Iteration 42: HR = 0.6894, NDCG = 0.4121.

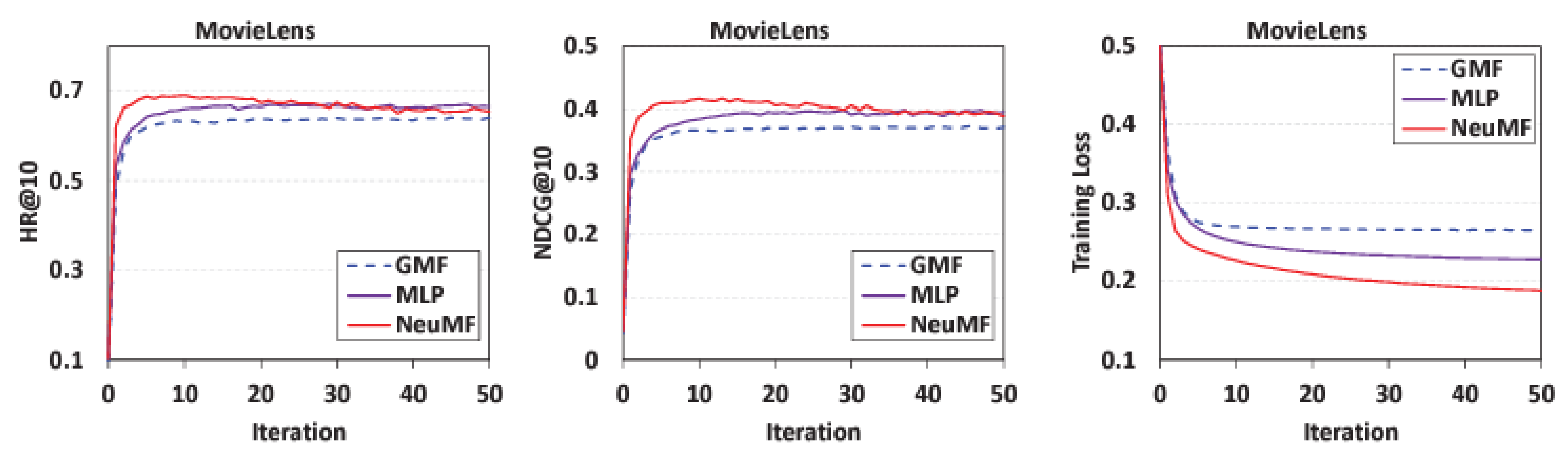
factor 32



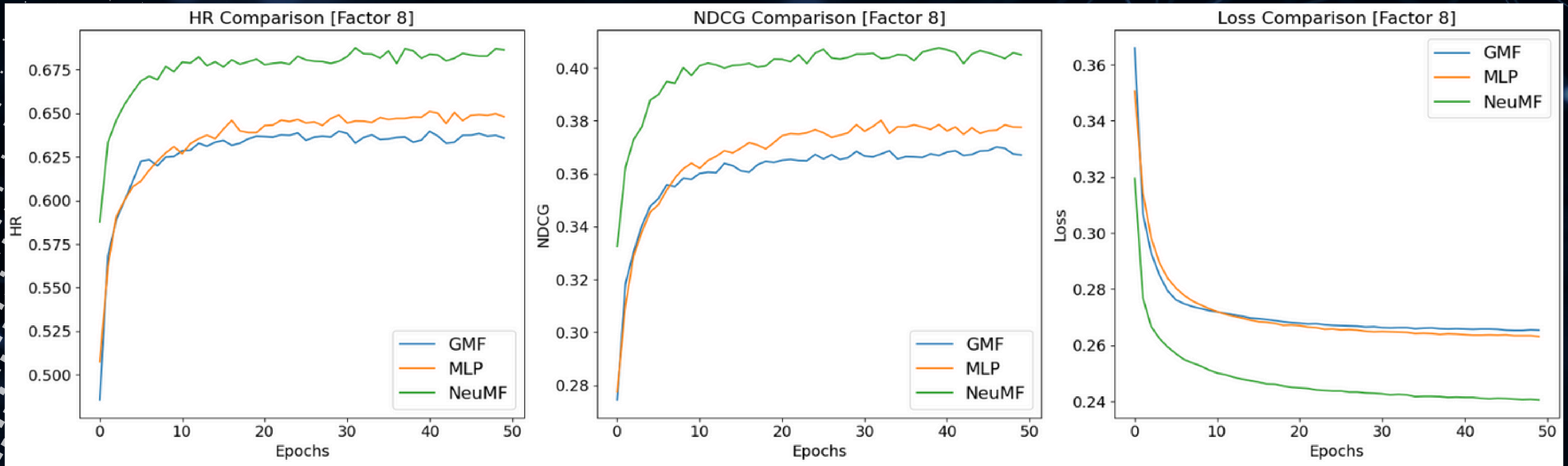
GMF: Best Iteration 5: HR = 0.6997, NDCG = 0.4189.
MLP: Best Iteration 23: HR = 0.6477, NDCG = 0.3729..
NeuMF: Best Iteration 18: HR = 0.6899, NDCG = 0.4171.

factor 64

COMPARE PEFORMANCE PER ITERATION

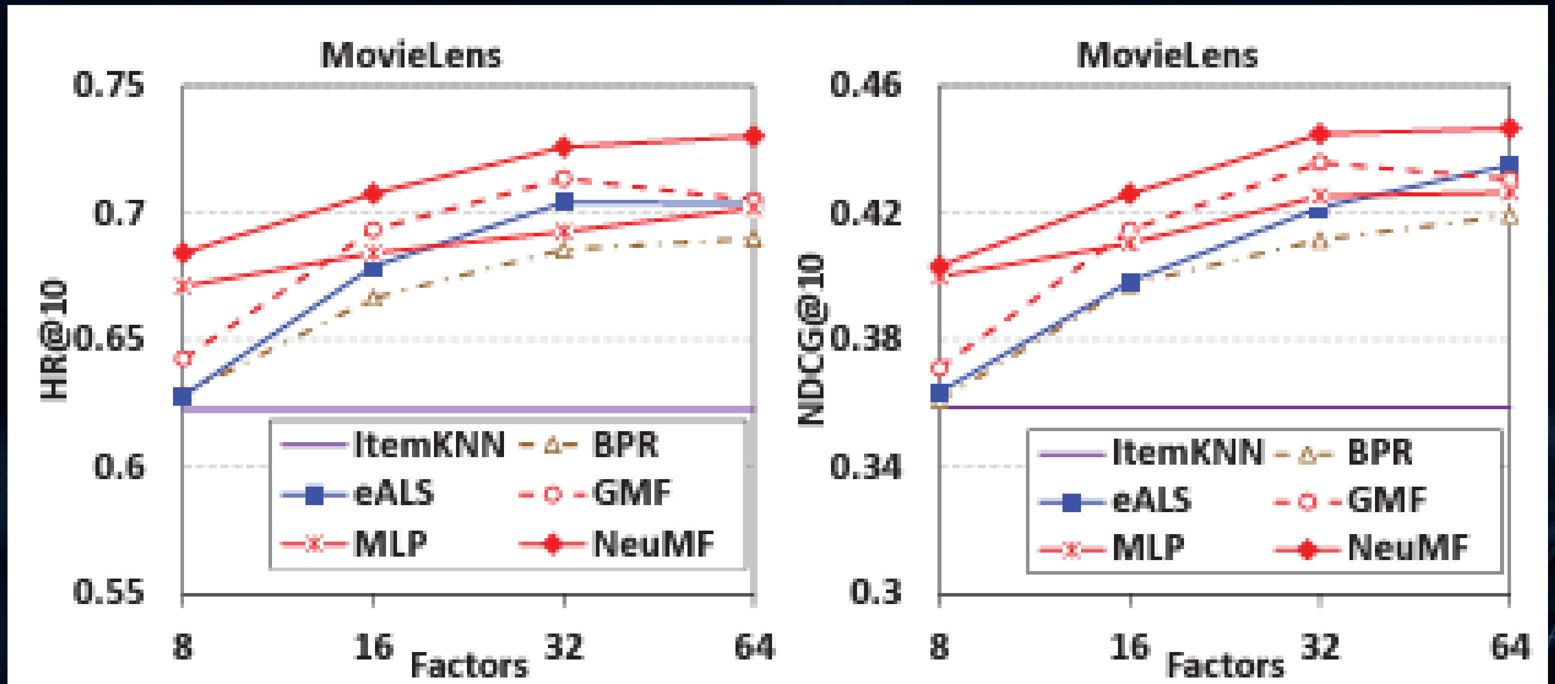


ORGINAL PERFORMANCE

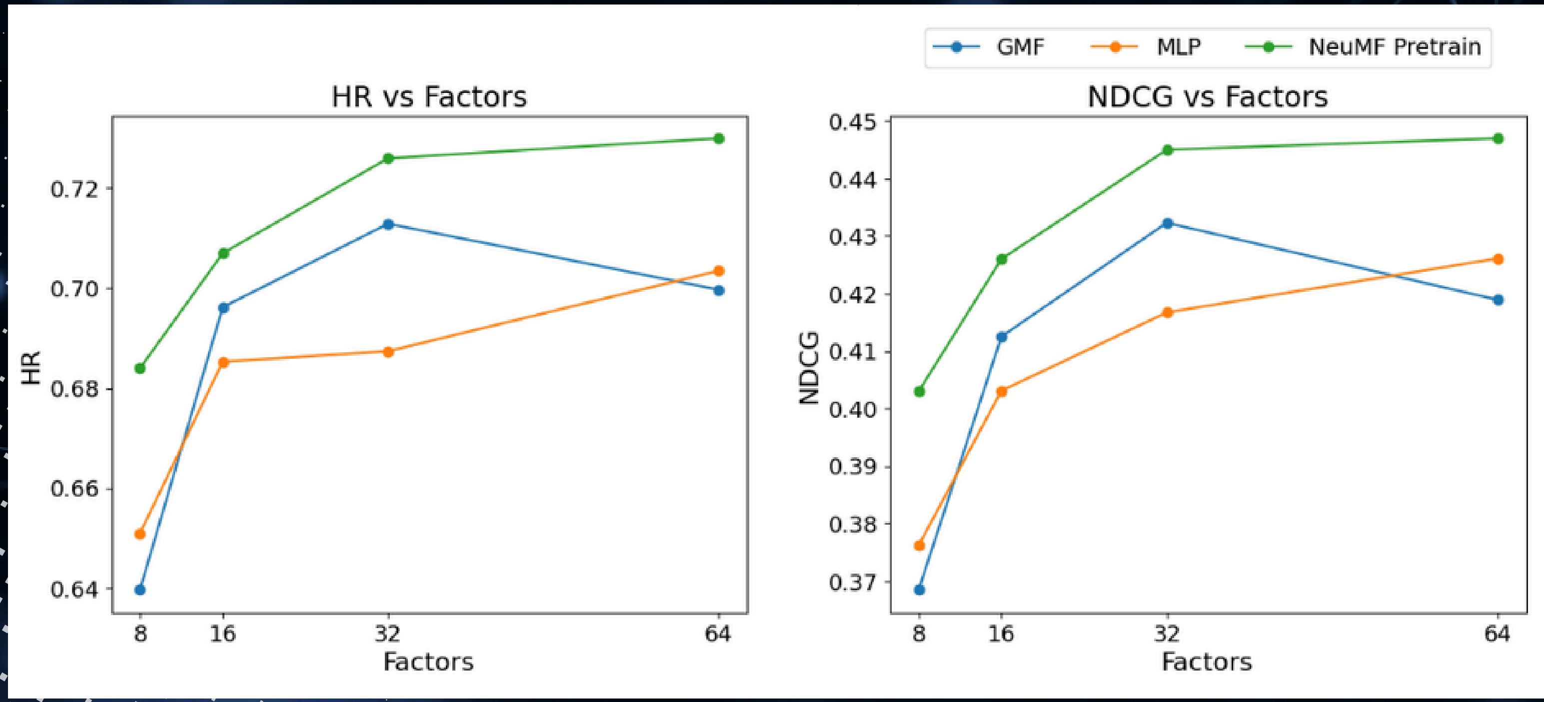


REPRODUCE PERFORMANCE

COMPARE PEFORMANCE PER FACTOR



ORGINAL PERFORMANCE



REPRODUCE PERFORMANCE

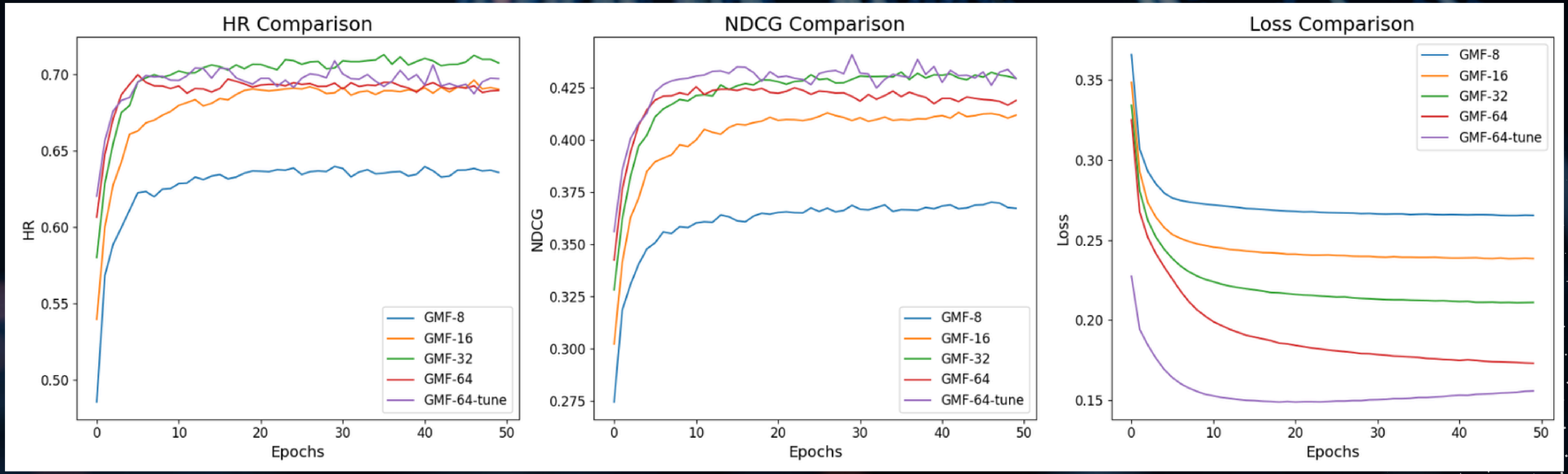
04 SUGGESTION

HYPERPARAMETER TUNING

- Learning Rate 0.002
- Number of negative negative samples per positive instance -> 8
- Number of factors (GMF) -> 64

OPTIZIMER SELECTION

- Adam



04 SUGGESTION

Metric	Before	After	Modified
HR	0.6997	0.7088	Increase 0.0091 (Approximately 1.3%)
NDCG	0.4189	0.4407	Increase 0.0218 (Approximately 5.2%)
Best Iteration	5	29	More Iteration

BENEFIT

Users Behaviour Data

Utilize the interaction data between users and items to generate value.

Collaborative Filtering

Instead of regular collaborative filterin, using neural network, NeuMF provides better performance and more trustworthy.

Improved Recommendation

User recieve better item recommendation from system results in good customer sastifaction.

THANK YOU

Members
6620422012 Tanutpong Kunakornkasem
6620422026 Winna Bangsoi
6620422027 Napatee Kanchanawilas