CSCE 670 - Information Storage and Retrieval

Lecture 19: Neural Collaborative Filtering and Quiz 3

Yu Zhang

yuzhang@tamu.edu

October 30, 2025

Course Website: https://yuzhang-teaching.github.io/CSCE670-F25.html

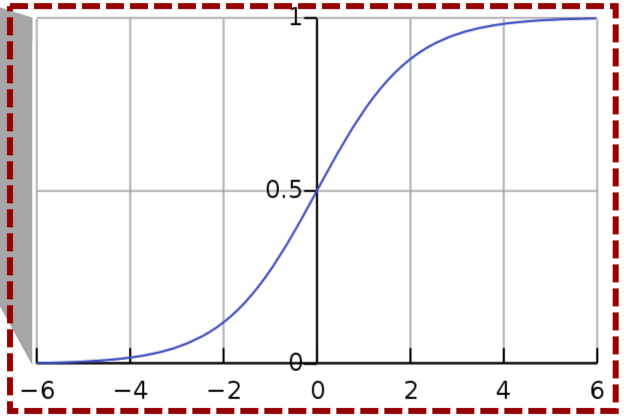
Neural Word Embeddings (word2vec)

[0.3, -0.7]	[-0.9, 0.4]	[0.8, 0.2]		[-0.2, -0.6]	[0.5, 0.9]	[-0.7, -0.1]
openai	*proposes*	*training*	*[MASK]*	*using*	*reinforcement*	*learning*

↓

chatgpt

Nonlinear



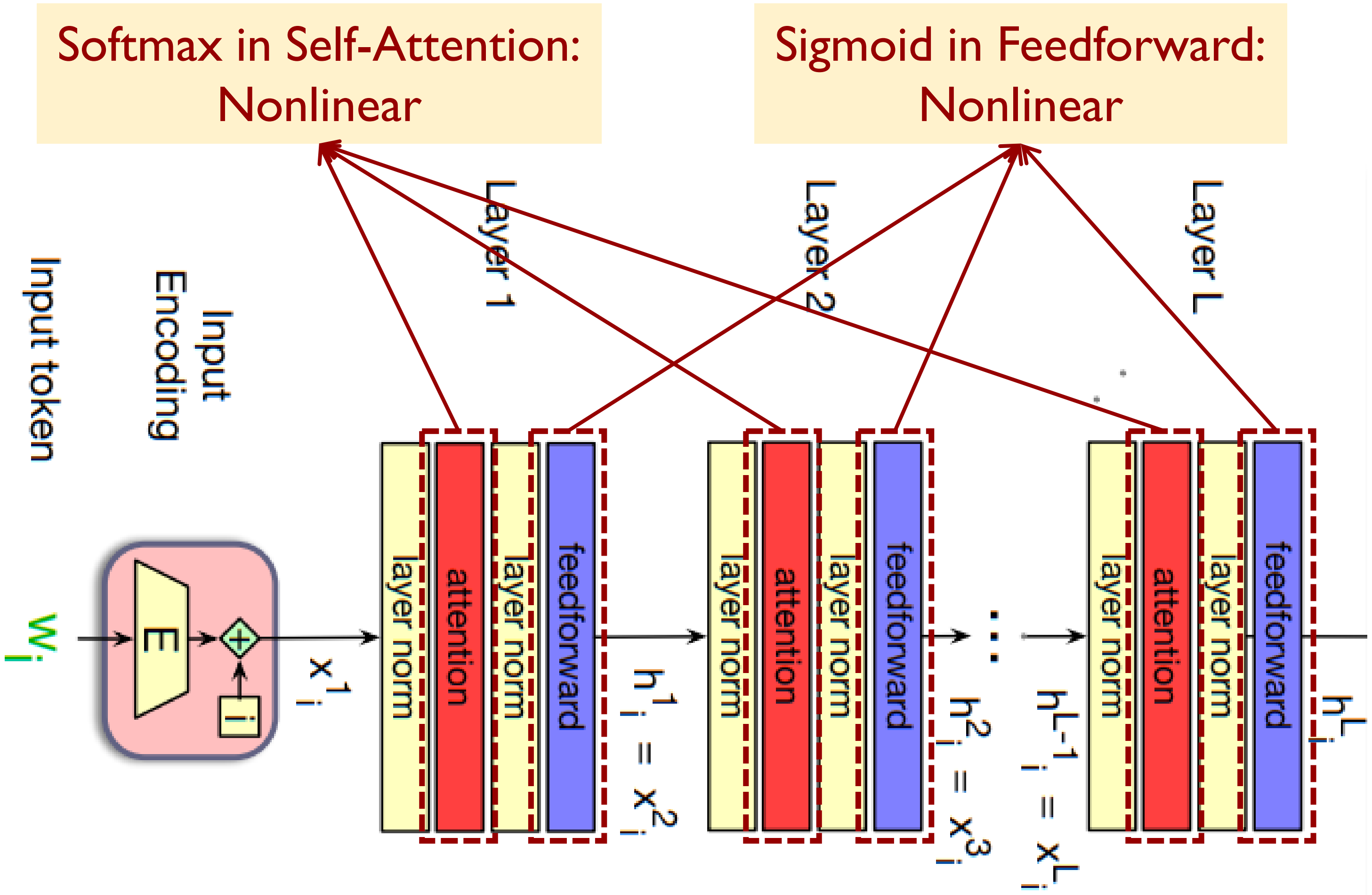
- [Levy and Golberg, NIPS 2014] word2vec is mathematically equivalent to factorizing a word-word matrix U, where

Nonlinear

$$U_{xy} = \boxed{\log} \frac{\#(x, y) \cdot |\mathcal{D}|}{\#(x) \cdot \#(y) \cdot b}$$

Neural Language Models (BERT)

- Transformer is a neural network



Nonlinearity in Recommender Systems

- Do the recommender system techniques we have learned so far contain enough nonlinearity?
  - No, matrix factorization is essentially modeling the inner product of vectors, which is a highly linear operation.

users

items

1		3			5			5		4	
		5	4			4			2	1	3
2	4		1	2		3		4	3	5	
	2	4		5			4			2	
		4	3	4	2					2	5
1		3		3			2			4	

$\boldsymbol{U}$

$\approx$

factors

items

.1	-.4	.2
-.5	.6	.5
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

$\boldsymbol{Q}$

$\times$

users

factors

1.1	-.2	.3	.5	-2	-.5	.8	-.4	.3	1.4	2.4	-.9
-.8	.7	.5	1.4	.3	-1	1.4	2.9	-.7	1.2	-.1	1.3
2.1	-.4	.6	1.7	2.4	.9	-.3	.4	.8	.7	-.6	.1

$\boldsymbol{P}^T$

$$U_{xi} = \boldsymbol{q}_i \boldsymbol{p}_x^T$$

How to Introduce Nonlinearity into Recommender Systems?

Neural Collaborative Filtering [He et al., WWW 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.

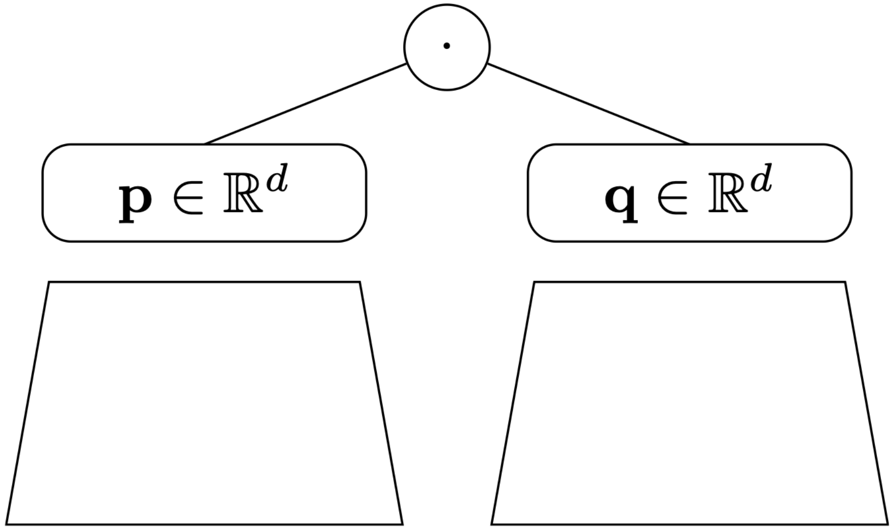
1. INTRODUCTION

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

Learning the Similarity Function

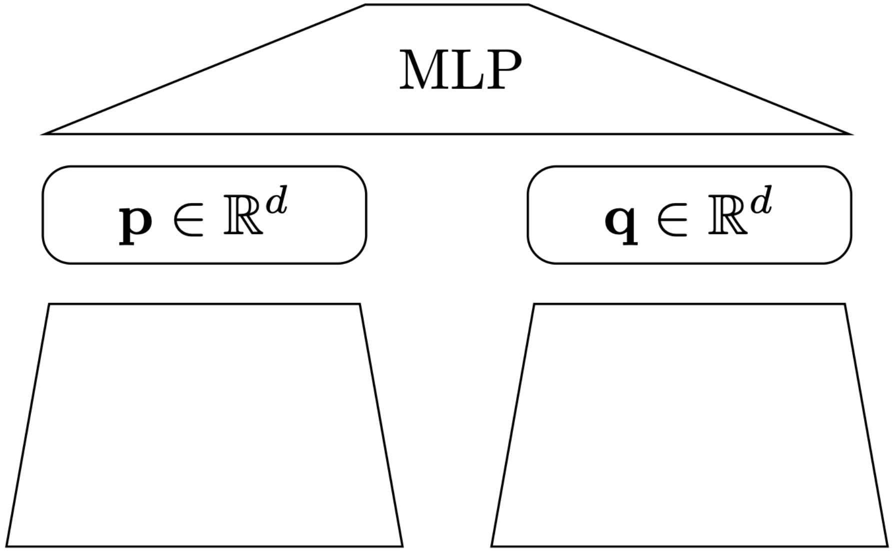
- Predefined similarity: inner product

$$\phi^{\text{dot}}(\mathbf{p}, \mathbf{q}) = \langle \mathbf{p}, \mathbf{q} \rangle$$



- Learned similarity: multi-layer perceptron (MLP)

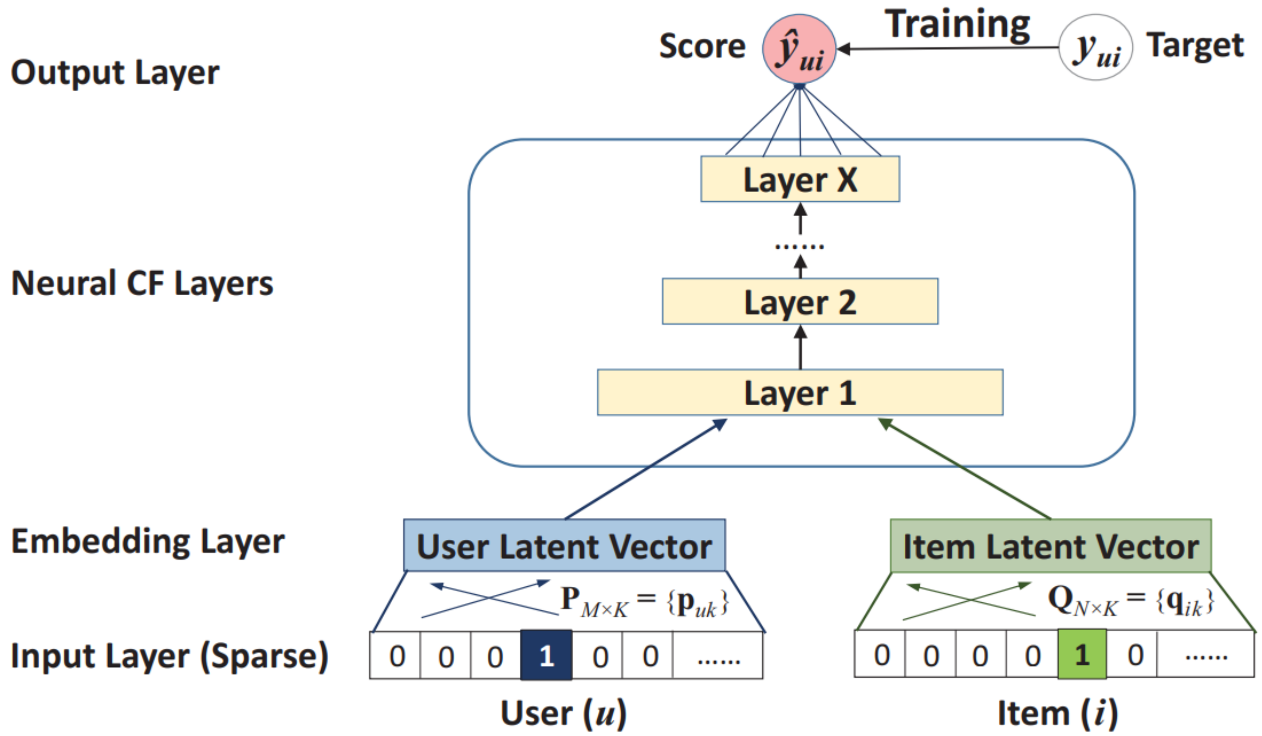
$$\phi^{\text{MLP}}(\mathbf{p}, \mathbf{q}) = \mathbf{f}_{W_l, \mathbf{b}_l}(\ldots \mathbf{f}_{W_1, \mathbf{b}_1}([\mathbf{p}, \mathbf{q}]) \ldots)$$



- We could use any sort of embeddings down here (e.g., from SVD or Matrix Factorization)

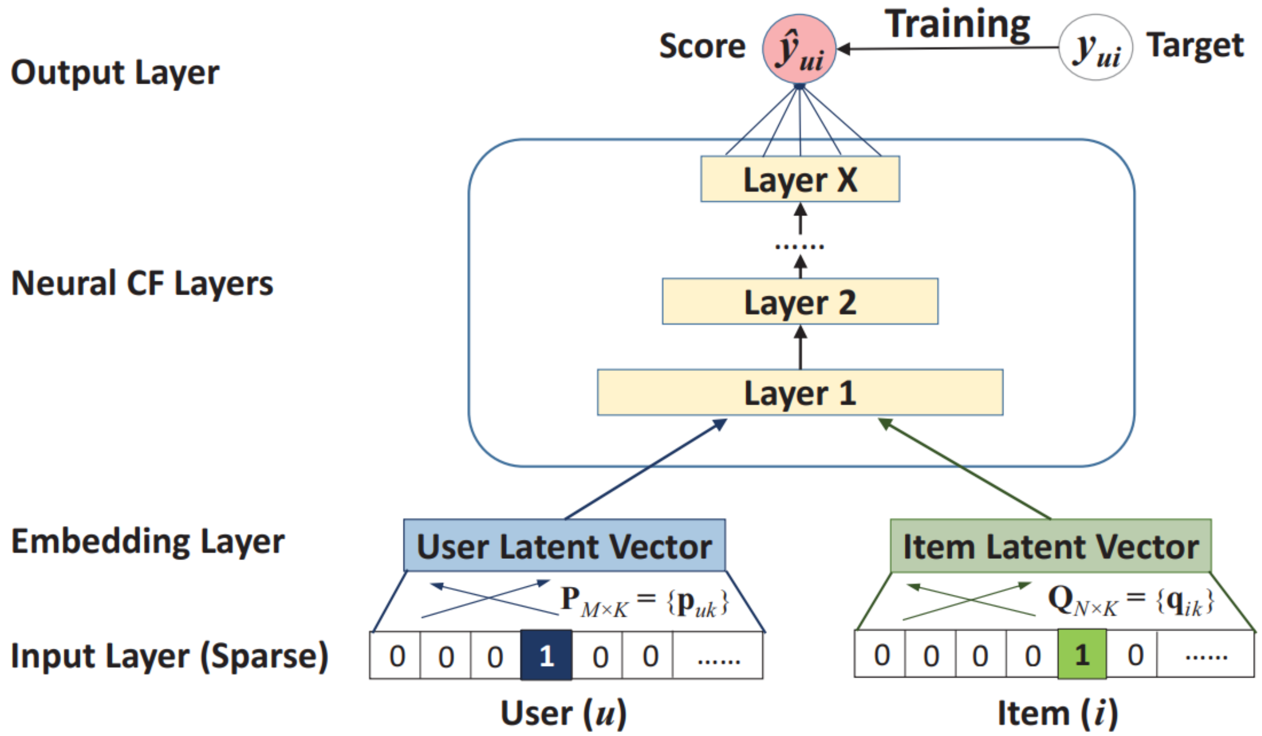
Example

- Suppose after matrix factorization,
  - User u's vector: $\boldsymbol{p} = [2, 2]$
  - Item i's vector: $\boldsymbol{q} = [2, 3]$
- Inner product: $\text{score}(u, i) = \boldsymbol{p}\boldsymbol{q}^T = 10$
- MLP (assuming 3 layers):
  - Input $\boldsymbol{x}_0 = [\boldsymbol{p}, \boldsymbol{q}] = [2, 2, 2, 3]$
  - $\boldsymbol{x}_1 = \text{Sigmoid}(\boldsymbol{W}_1\boldsymbol{x}_0 + \boldsymbol{b}_1)$
  - $\boldsymbol{x}_2 = \text{Sigmoid}(\boldsymbol{W}_2\boldsymbol{x}_1 + \boldsymbol{b}_2)$
  - $\hat{y} = \text{score}(u, i) = \text{Sigmoid}(\boldsymbol{W}_3\boldsymbol{x}_2)$



Example

- MLP (assuming 3 layers):
  - Input $\boldsymbol{x}_0 = [\boldsymbol{p}, \boldsymbol{q}] = [2, 2, 2, 3]$
  - $\boldsymbol{x}_1 = \text{Sigmoid}(\boldsymbol{W}_1 \boldsymbol{x}_0 + \boldsymbol{b}_1)$
  - $\boldsymbol{x}_2 = \text{Sigmoid}(\boldsymbol{W}_2 \boldsymbol{x}_1 + \boldsymbol{b}_2)$
  - $\hat{y} = \text{Sigmoid}(\boldsymbol{W}_3 \boldsymbol{x}_2)$
- Parameters?
  - $\boldsymbol{W}_1, \boldsymbol{b}_1, \boldsymbol{W}_2, \boldsymbol{b}_2, \boldsymbol{W}_3$
  - $\boldsymbol{P}, \boldsymbol{Q}$
- Learning objective?
  - $\text{RMSE} = (y - \hat{y})^2$
- Training data?
  - Known ratings in the user-item matrix



Combining Matrix Factorization and MLP

- Inner product: $\text{score}(u, i) = \boldsymbol{p}\boldsymbol{q}^T = 10$

- MLP (assuming 3 layers):
  - Input $\boldsymbol{x}_0 = [\boldsymbol{p}, \boldsymbol{q}] = [2, 2, 2, 3]$
  - $\boldsymbol{x}_1 = \text{Sigmoid}(\boldsymbol{W}_1\boldsymbol{x}_0 + \boldsymbol{b}_1)$
  - $\boldsymbol{x}_2 = \text{Sigmoid}(\boldsymbol{W}_2\boldsymbol{x}_1 + \boldsymbol{b}_2)$
  - $\text{score}(u, i) = \text{Sigmoid}(\boldsymbol{W}_3\boldsymbol{x}_2)$

- MLP does not explicitly model the interaction between $\boldsymbol{p}$ and $\boldsymbol{q}$, but inner product does!
- Can we fuse inner product and MLP?

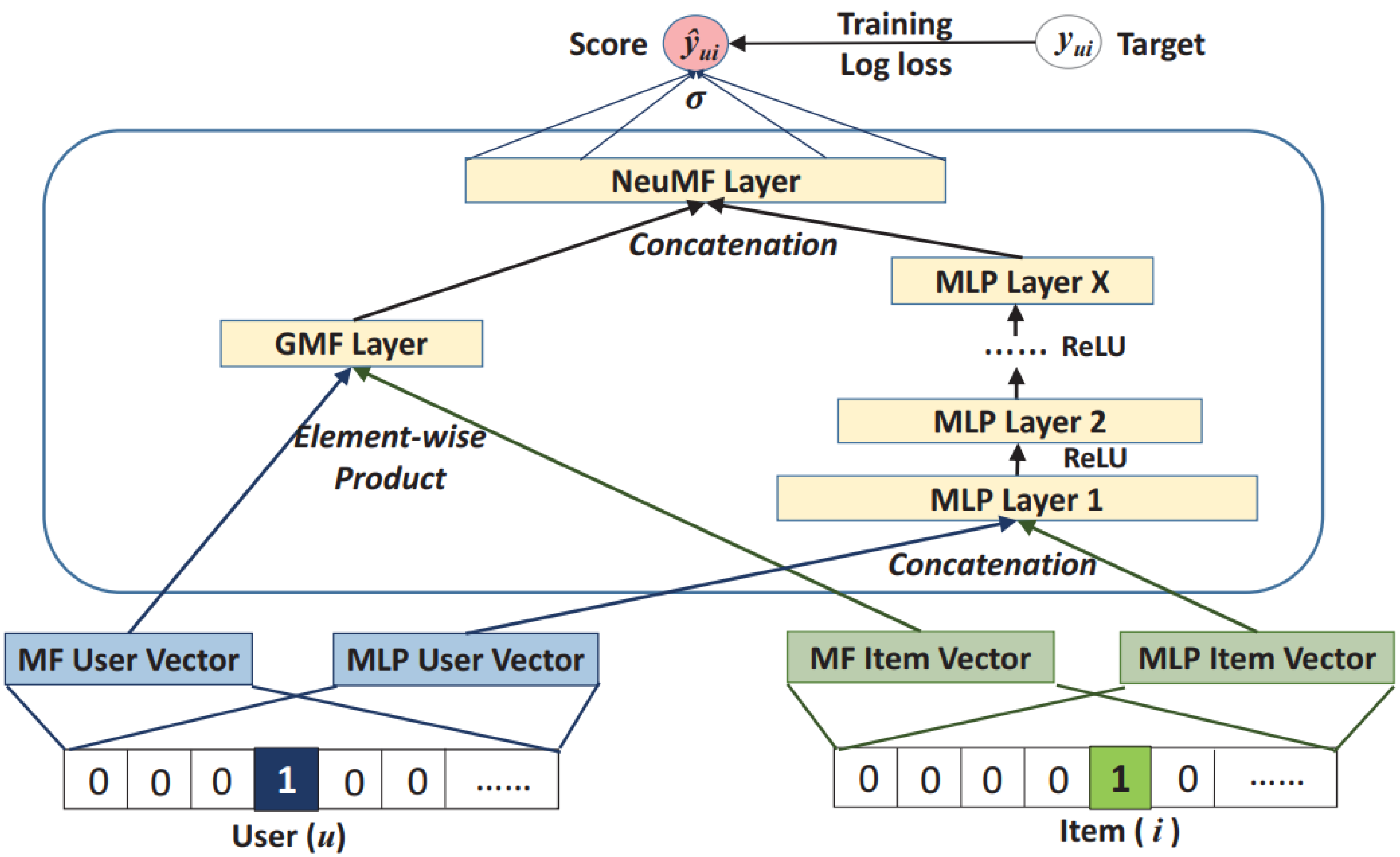
$$\text{score}(u, i) = \text{score}_{\text{MF}}(u, i) + \text{score}_{\text{MLP}}(u, i)$$

Combining Matrix Factorization and MLP

- User u's vector: $\boldsymbol{p} = [2, 2]$
- Item i's vector: $\boldsymbol{q} = [2, 3]$

- MLP: same

- Generalized Matrix Factorization (GMF):
  - Step 1: Element-wise product $\boldsymbol{p} \odot \boldsymbol{q} = [4, 6]$
  - Step 2: Nonlinear layer $\text{score}_{\text{MF}}(u, i) = \text{Sigmoid}(\boldsymbol{w} \begin{bmatrix} 4 \\ 6 \end{bmatrix})$
  - $\boldsymbol{w}$ are the parameters to be learned



Generalized Matrix Factorization vs. Inner Product

- User u's vector: $\boldsymbol{p} = [2, 2]$
- Item i's vector: $\boldsymbol{q} = [2, 3]$

- Generalized Matrix Factorization (GMF):
  - Step 1: Element-wise product $\boldsymbol{p} \odot \boldsymbol{q} = [4, 6]$
  - Step 2: Nonlinear layer $\text{score}_{\text{MF}}(u, i) = \text{Sigmoid}(\boldsymbol{w} \begin{bmatrix} 4 \\ 6 \end{bmatrix})$

- Suppose $\boldsymbol{w} = [1, -1]$, then

$$\boldsymbol{w}(\boldsymbol{p} \odot \boldsymbol{q}) = [1, -1] \begin{bmatrix} 2 \times 2 \\ 2 \times 3 \end{bmatrix} = 1 \times 2 \times 2 + (-1) \times 2 \times 3$$

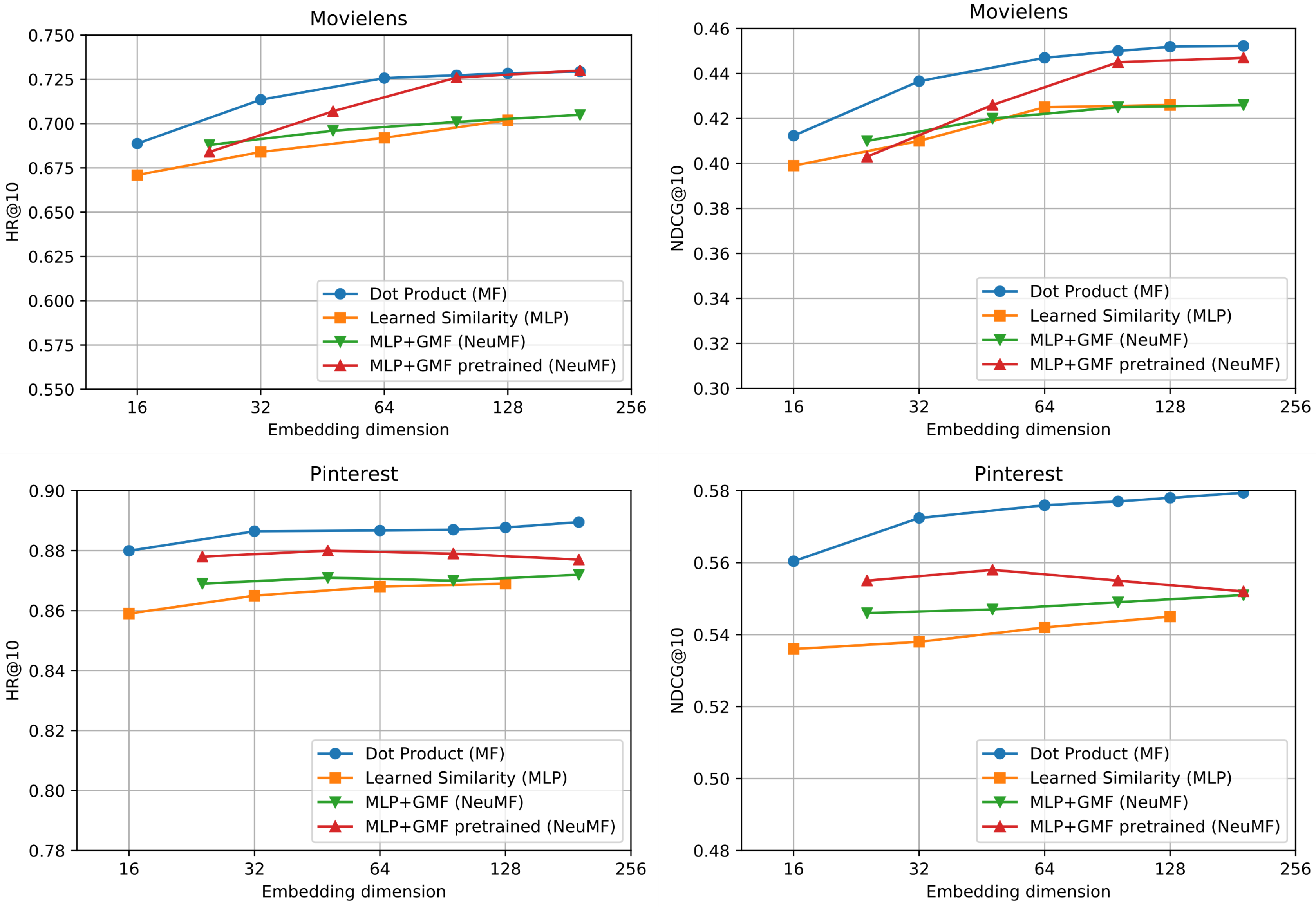
$$= [2, 2] \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \boldsymbol{p}\ \text{diag}\{\boldsymbol{w}\}\ \boldsymbol{q}^T$$

$$\boxed{\text{score}_{\text{MF}}(u, i) = \text{Sigmoid}(\boldsymbol{p}\ \text{diag}\{\boldsymbol{w}\}\ \boldsymbol{q}^T)}$$

Nonlinear

Learnable Parameters

However, …



- Is Neural Collaborative Filtering really better than Matrix Factorization?
- Next Lecture

Quiz 3

Thank You!

Course Website: https://yuzhang-teaching.github.io/CSCE670-F25.html