Recommender Systems

Presented by Tanvir Deol
ts3deol@uwaterloo.ca

Notebook Link Here

Workshop Overview

Prereqs:

- Some basic ML knowledge
- Basic Linear Algebra

Goals:

- Start with basic architectures and build our way up in complexity
    - Basic Models
    - Models currently used in Industry
    - Emerging Architectures
- Implement our own recommendation systems

Workshop Outline

- What are Recommender Systems?
- Motivation
- History Timeline
- Level 1: Basic Models
  - Content Based Filtering (KNN)
  - Collaborative Filtering (Matrix Factorization)
- Level 2: Industry Models
  - Neural Collaborative Filtering
- Level 3: Emerging Architectures
  - Graph Neural Networks
- Notebook
- Useful Resources

What are Recommender Systems?

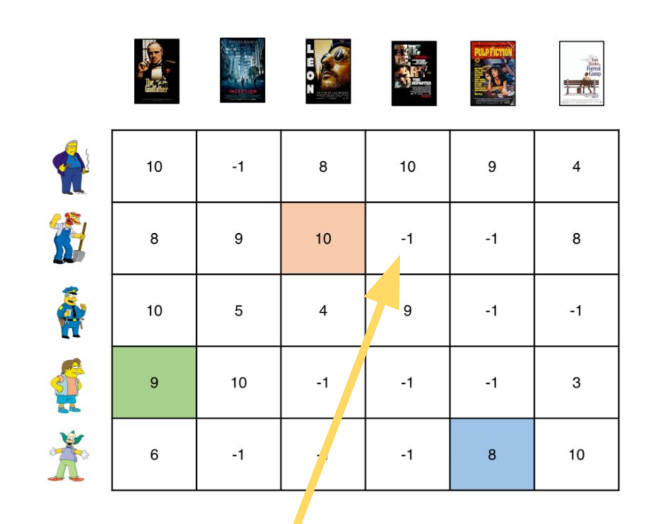
UW DSC.

Recommender Systems (Rec Sys) are a class of ML model made to provide personalized recommendations relevant to users

Terminology:
Items - Info about objects being recommended (e.g. movies)
Users - Info about users
Embeddings - Representations of a user/items in a N-dimensional vector

Items ***j***

Users ***i***



10	-1	8	10	9	4
8	9	10	-1	-1	8
10	5	4	9	-1	-1
9	10	-1	-1	-1	3
6	-1	[illegible]	-1	8	10

How much user ***i*** ‘likes’ item ***j***

Motivation

- Rec Sys are one of the most relevant ML topics to industry
- They are a core aspect of revenue-generation for many B2C tech companies
- Aim to maximize customer retention

Facts:

- Rec Sys are responsible for **35%** of Amazon.com's revenue
- **80%** of Netflix stream time comes from its recommendations

Examples in Industry:

- Amazon.com Products
- Netflix Movies
- Youtube Home page
- Instagram Explore Page
- Steam Explore Page

And so on ...

History Timeline

UW DSC.

Basic Models

Industry Models

Emerging Architectures

Content Based Filtering

Collaborative Filtering

Matrix Factorization

Neural Collaborative Filtering

Reinforcement Learning, Graph Neural Networks ...

1990

2000

2010

2020

Level 1 - Content Based Filtering

Defn: Recommending **similar items** to what the user likes

Ex - Video Game Recommendations

Features

	F1	F2	F3
User 1	0	4	9
User 2	6	2	1
User 3	7	3	5

Features

	F1	F2	F3
Item 1	5	5	2
Item 2	9	7	9
Item 3	0	3	8

Feature Labels:

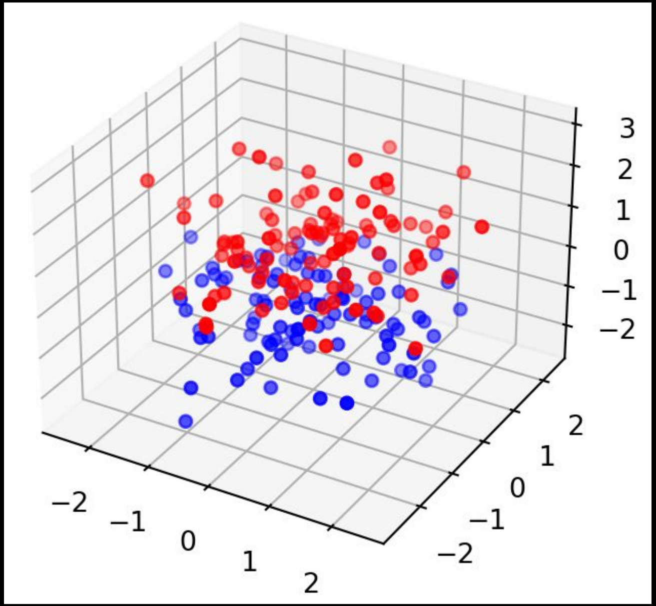
F1 - FPS

F2 - Adventure

F3 - Sports

Level 1 - Content Based Filtering

Given the user + item embeddings we can represent them in a N dimensional space:



Red Points - Users
Blue Points - Items

The Rec Sys has to find the most similar items for user ***i*** using a **similarity metric** that measures distance between embeddings/points

Level 1 - Content Based Filtering

UW DSC.

Similarity Metric: Euclidean Distance

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

Magnitude of a straight line between two points in a N-dimensional space

Find Similar Points: K-Nearest Neighbors (KNN)

- Initialize value of K
- For every item embedding:
  - Calculate distance between item and user i
- Find the K smallest calculated distances
- The corresponding items are your “similar items”

Precision @ K:

$$\frac{\text{(\# of relevant items at K)}}{K}$$

Level 1 - Content Based Filtering

Pros	Cons
Easy to understand + High interpretability	Needs hand-engineered features beforehand
Scalable for lots of users (since users treated independently)	Needs information for each user and item beforehand (expensive & infeasible)

Level 1 - Collaborative Filtering

UW DSC.

Defn: Recommending items based on what **similar users** like

Feedback Matrix A_{MxN}

	Harry Potter	The Triplets of Belleville	Shrek	The Dark Knight Rises	Memento
	✓		✓	✓	
		✓			✓
	✓	✓	✓		
				✓	✓

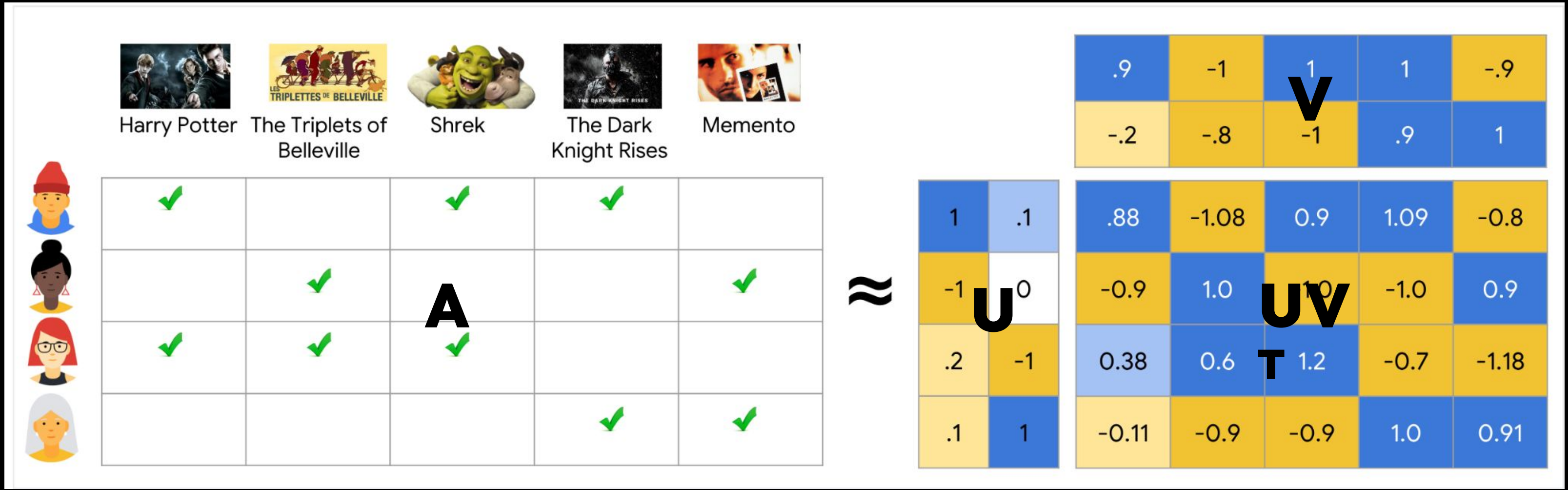
$A_{i,j} = 1 \rightarrow$ User i interested in item j

$A_{i,j} = 0 \rightarrow$ User i not interested in item j (or unobserved)

Level 1 - Collaborative Filtering

UW DSC.

Item Embedding Matrix



Feedback Matrix

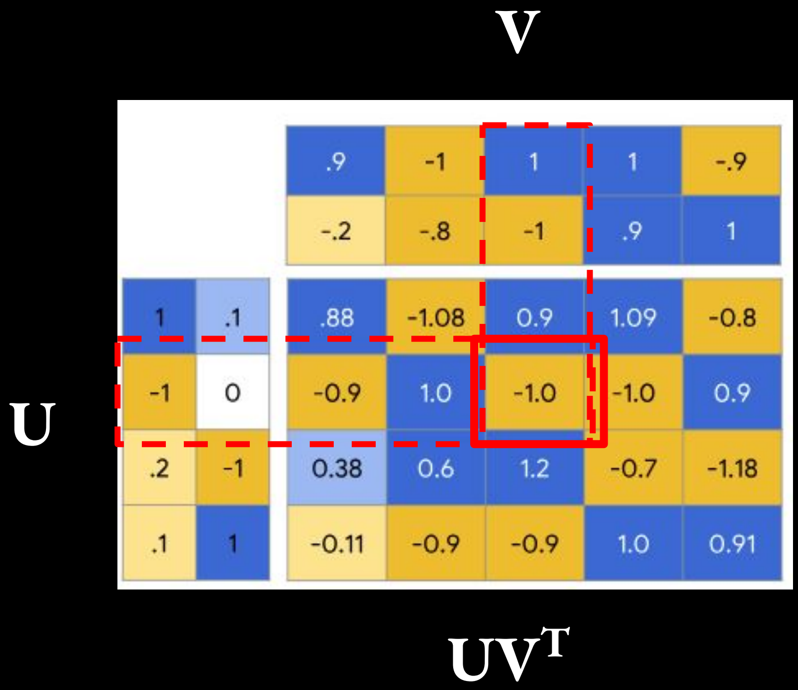
User Embedding Matrix

Level 1 - Collaborative Filtering

Matrix Factorization:
Finds user matrix *U* and item matrix *V* that minimizes the function

$$\min_{U \in \mathbb{R}^{m \times d},\, V \in \mathbb{R}^{n \times d}} \|A - UV^T\|_F^2.$$

Ways to solve
- Singular Value Decomposition (too slow)
- Stochastic Gradient Descent



Level 1 - Collaborative Filtering

Pros	Cons
Don't need to specify features beforehand	Unable to capture **non-linear relationships** between users and items
No domain knowledge required	(Since UV^T is made up of weighted sums)
Makes use of all data available for predictions	

Level 2 - Neural Collaborative Filtering

Was the next breakthrough in RecSys in 2017

Kickstarted the use of Deep Learning (Neural Networks) in Rec Sys.

Made the upgrade from the previous industry standard: Matrix Factorization

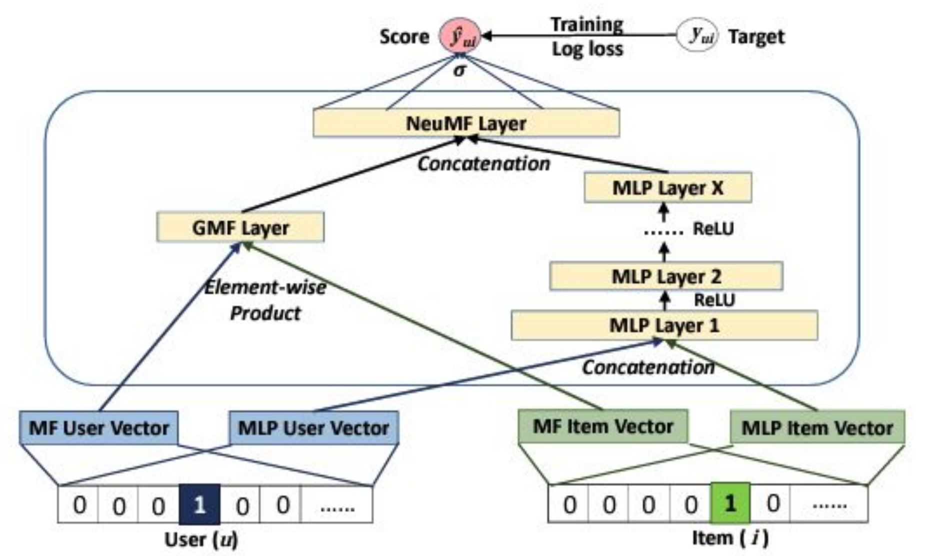


Figure 3: Neural matrix factorization model

Level 2 - Neural Collaborative Filtering

Purpose:
Model predicts interaction score between ***user u*** and **item *j***

Predicted score is denoted $\hat{y}_{ui}$

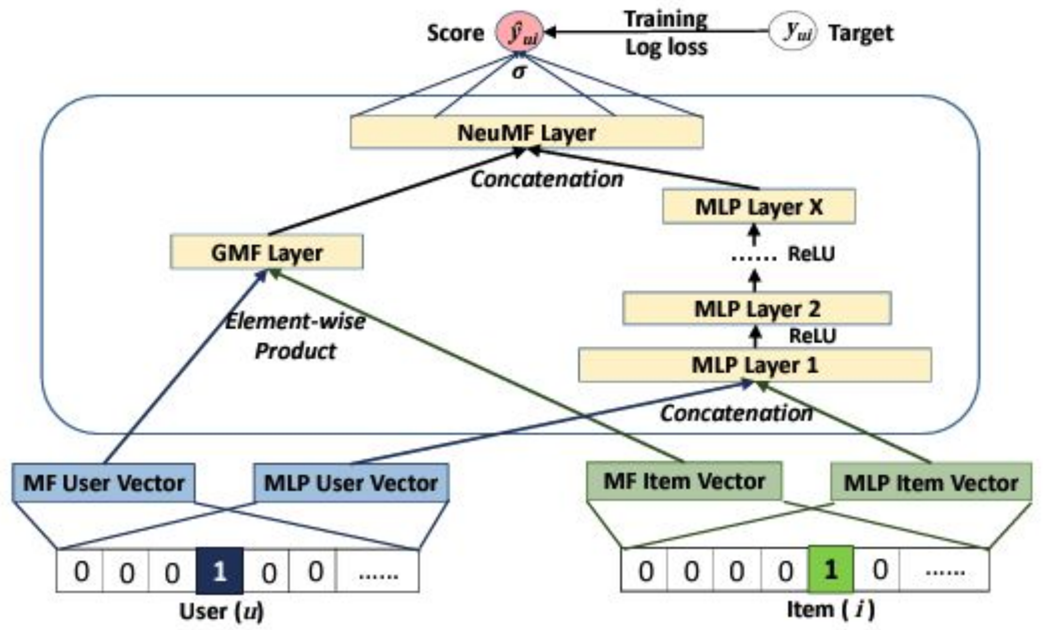


Figure 3: Neural matrix factorization model

Level 2 - Neural Collaborative Filtering

Input Layer:
IDs of (user, item) pair in sparse one hot encoded form

Ex:
4 → [0,0,0,1,0,0, ...]
5 → [0,0,0,0,1,0, ...]

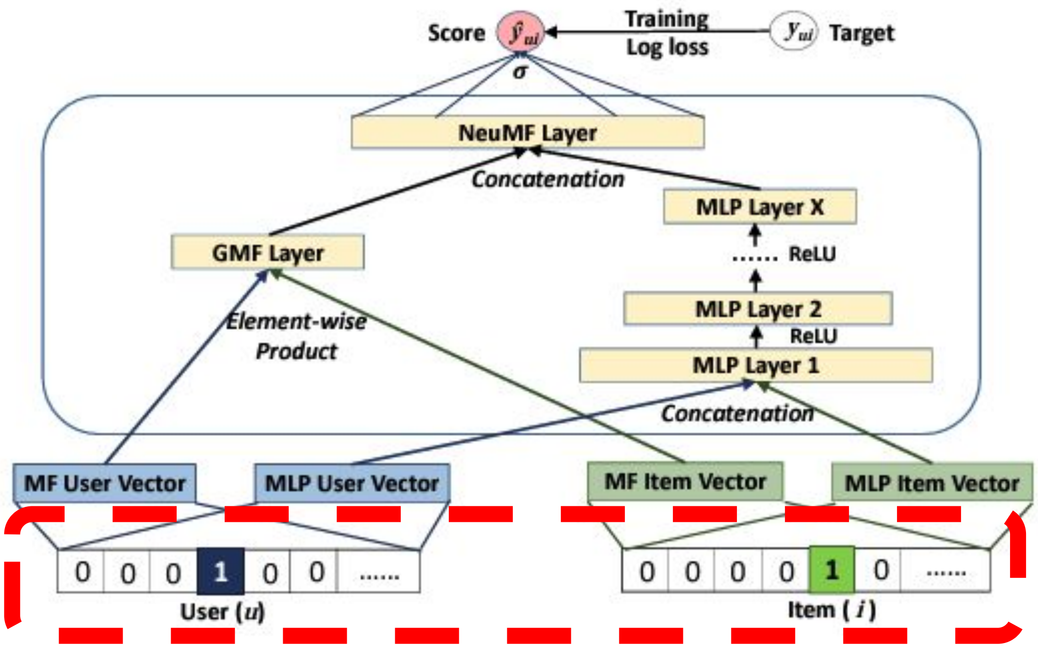
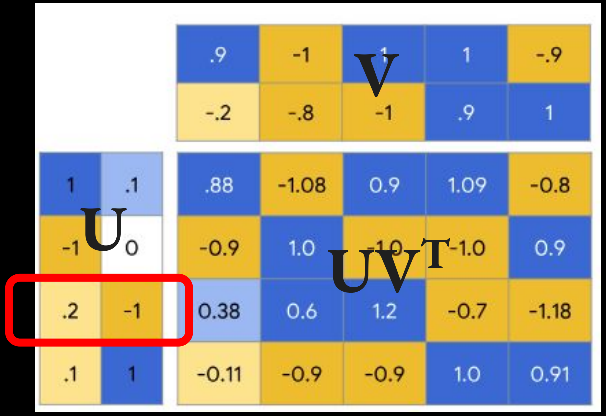


Figure 3: Neural matrix factorization model

Level 2 - Neural Collaborative Filtering

Embedding Layer:
The inputs converted to their corresponding embeddings from matrix factorization model

Embedding of user ID 3 below



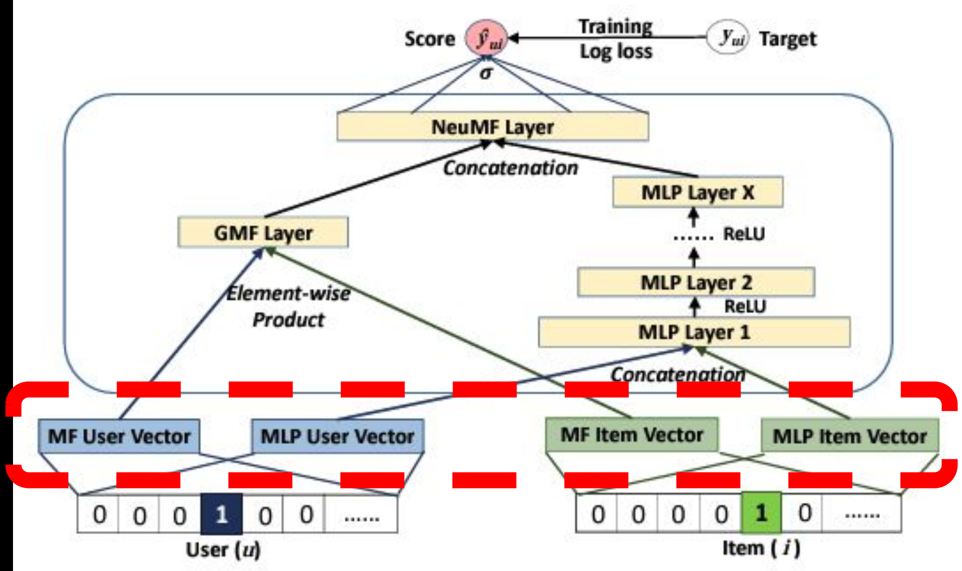


Figure 3: Neural matrix factorization model

Level 2 - Neural Collaborative Filtering

Generalized Matrix Factorization Layer:

$$\mathbf{x} = \mathbf{p}_u \odot \mathbf{q}_i$$
$$\hat{y}_{ui} = \alpha(\mathbf{h}^\top \mathbf{x}),$$

p_u = embedding for user u
q_i = embedding for item i
x = resulting input vector
a = activation function of NN
h = weights of NN
$\hat{y}_{ui}$ = prediction for user u, item i

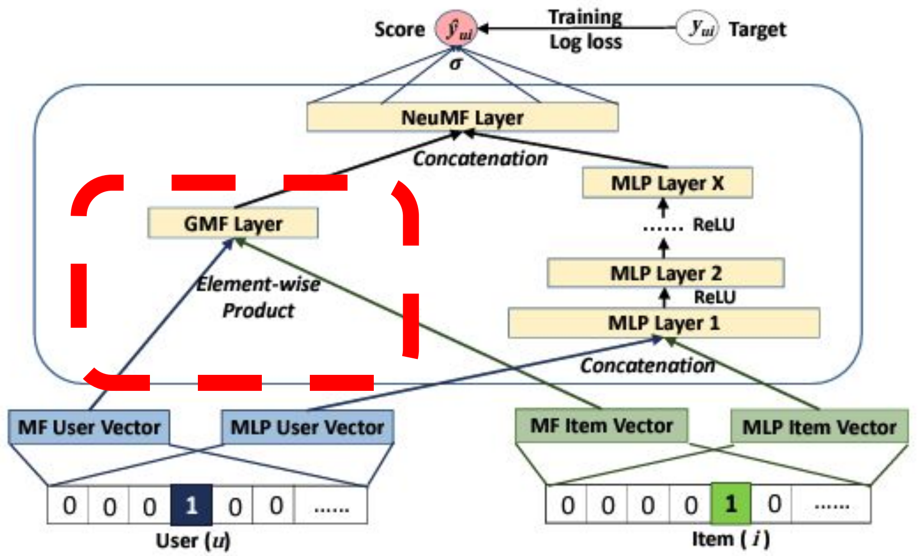


Figure 3: Neural matrix factorization model

Level 2 - Neural Collaborative Filtering

UW DSC.

Multi-Layer Perceptron Layer:

Responsible for capturing the non-linear patterns

$$z^{(1)} = \phi_1(\mathbf{U}_u, \mathbf{V}_i) = [\mathbf{U}_u, \mathbf{V}_i]$$
$$\phi^{(2)}(z^{(1)}) = \alpha^1(\mathbf{W}^{(2)} z^{(1)} + b^{(2)})$$
$$\cdots$$
$$\phi^{(L)}(z^{(L-1)}) = \alpha^L(\mathbf{W}^{(L)} z^{(L-1)} + b^{(L)}))$$
$$\hat{y}_{ui} = \alpha(\mathbf{h}^\top \phi^L(z^{(L-1)}))$$

W - weight matrix
b - bias vector
a - activation function
Φ - function of labelled layer
z - output of labelled layer
$\hat{y}_{ui}$ = prediction for user u, item i

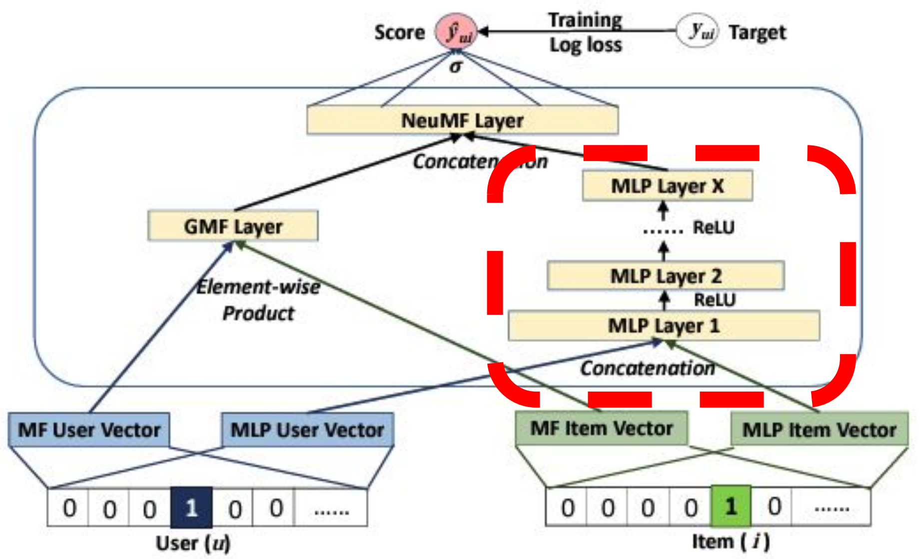


Figure 3: Neural matrix factorization model

Level 2 - Neural Collaborative Filtering

NeuMF Layer:
Combines results both networks

Concatenates the second last layers of the GMF and MLP networks to get:

$$\hat{y}_{ui} = \sigma(\mathbf{h}^{\top}[\mathbf{x}, \phi^{L}(z^{(L-1)})]).$$

σ - sigmoid activation function
h - projection matrix
$\hat{y}_{ui}$ = prediction for user u, item i

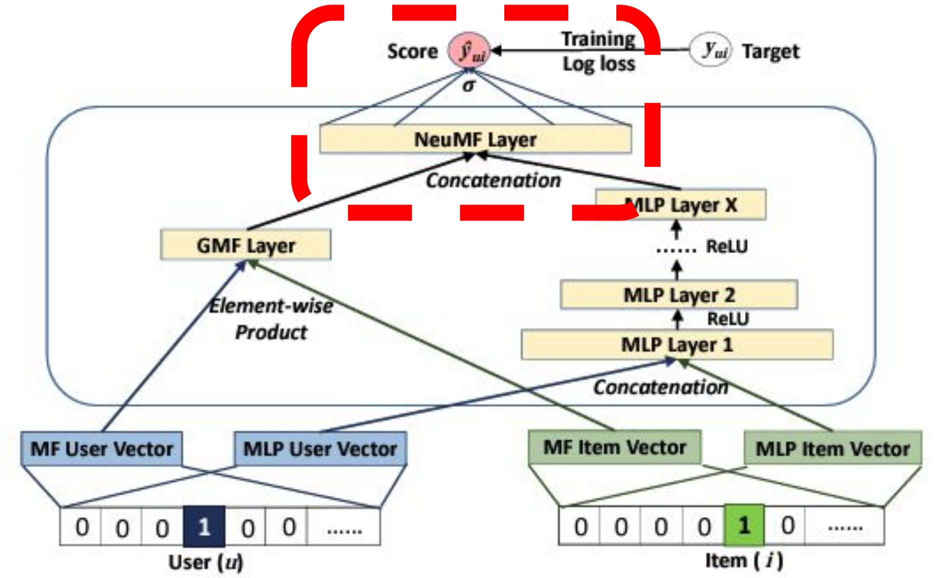


Figure 3: Neural matrix factorization model

Level 2 - Neural Collaborative Filtering

Loss Score:

Neural Collaborative Filtering uses pointwise loss:

$$L_{sqr} = \sum_{(u,i)\in\mathcal{Y}\cup\mathcal{Y}^-} w_{ui}(y_{ui} - \hat{y}_{ui})^2,$$

y - set of observed interactions

y^- - set of unobserved interactions

w_{ui} - weight hyperparameter

y_{ui} - target

$\hat{y}_{ui}$ - prediction

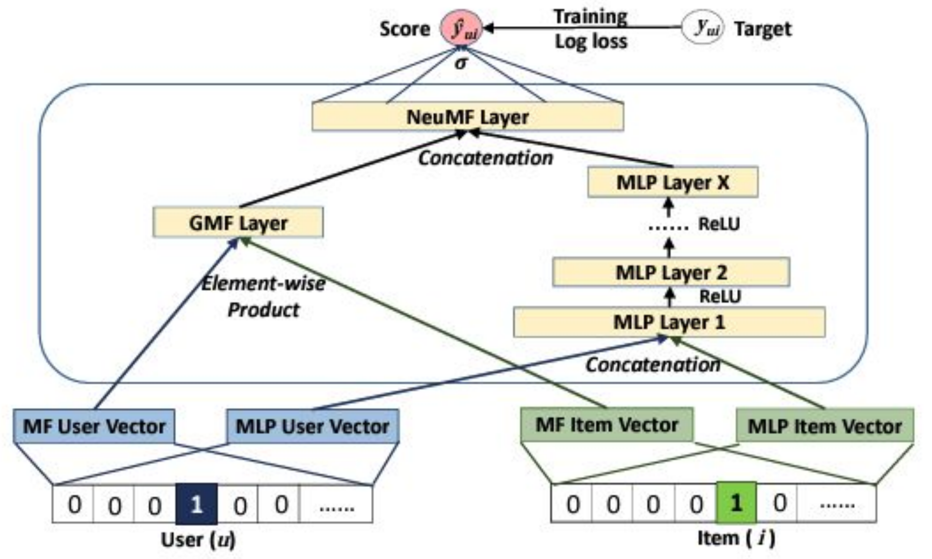


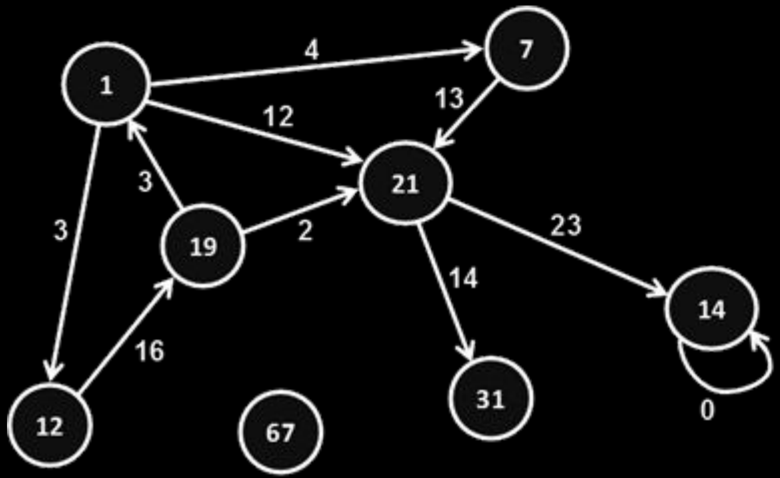
Figure 3: Neural matrix factorization model

Level 3 - Graph Neural Networks

Graph Based methods help capture **more complex relational dependencies** in the data, and have been gaining popularity since 2020

Work well for data with relational dependencies

- Social Media networks (mutual friends)
- Large Knowledge Graphs
- Citation Networks



Level 3 - Graph Neural Networks

Amazon - "Using graph neural networks to recommend related products"

- Using GNNs to model asymmetric product relationships leading to 30-160% performance improvements

Snap Inc - "Graph Neural Networks for Friend Ranking in Large-scale Social Platforms"

- Propose GNN model that uses messages, likes, & other interactions to generate higher quality user representations

Google Maps - "ETA Prediction with Graph Neural Networks in Google Maps"

- Use GNN to model connectivity of road networks to reduce inaccuracies of ETA calculations by more than 50%

Notebook

We will show how to implement basic recommender systems based on the ideas covered today

1. Content-Based Filtering (KNN method)
2. Collaborative Filtering (Matrix Factorization method)
3. Neural Collaborative Filtering

Useful Resources

UW DSC.

Datasets:

- [Recommender System Datasets]
- [Stanford Graph/Recommendation Datasets]
- [Netflix $1M Prize Competition Dataset]

Architectures:

- [Intro to Content-Based Filtering]
- [Intro to Collaborative Filtering]
- [Neural Collaborative Filtering Paper (2017)]

Future of Recommendation Systems:

- [Amazon Science: GNNs for related products]
- [Snap Inc: GNNs for Large Scale Friend Ranking]
- [Google Maps: ETA Prediction using GNNs]