

Neural Graph Collaborative Filtering

2020.12.08. Presenter: JJY

Wang, Xiang, et al. "Neural graph collaborative filtering." *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 2019.

1. Review

- Recommender System

사용자의 선호 및 과거의 행동을 바탕으로 개인에게 알맞는 아이템(컨텐츠)를 제공하는 시스템

- └ ...



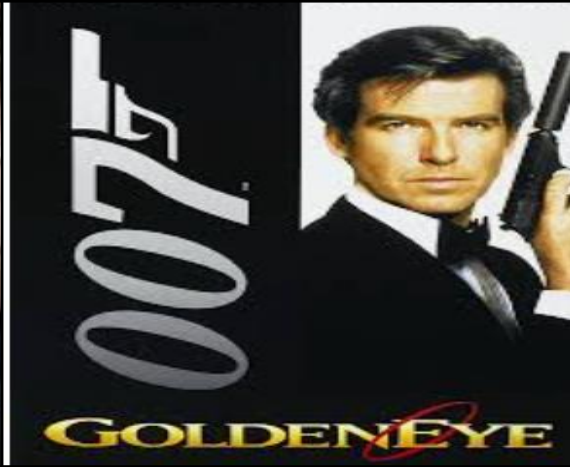





- └ **Collaborative Filtering**

 - └ ...

 - └ **Neural Collaborative Filtering**

1. Review

Collaborative Filtering

<div>Movies</div> <div>Users</div>				
	1	0	1	1
	1	1	0	0
	0	0	1	1
	0	1	0	1



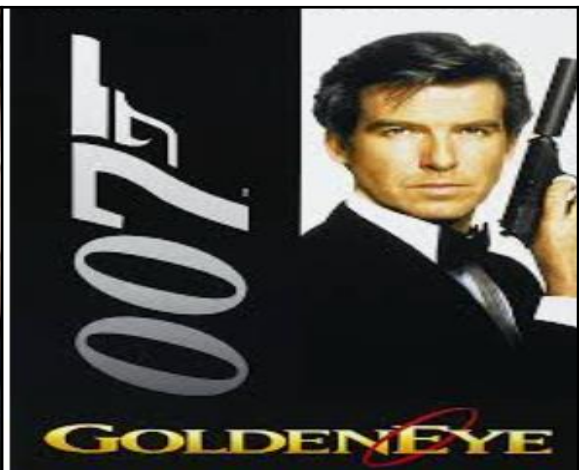

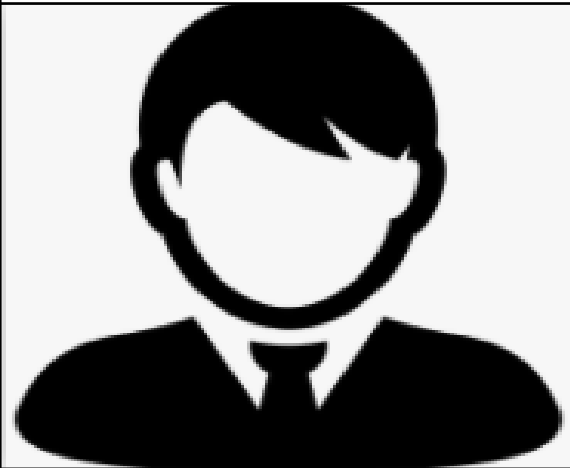



1. Review

Collaborative Filtering

User3 에게
어떤 영화를 추천해줄까?

유저 간 Cosine 유사도 구하기

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$

<div>Movies</div> <div>Users</div>					
		1	0	1	1
		1	1	0	0
		0	0	1	1
		0	1	0	1

Cosine 유사도



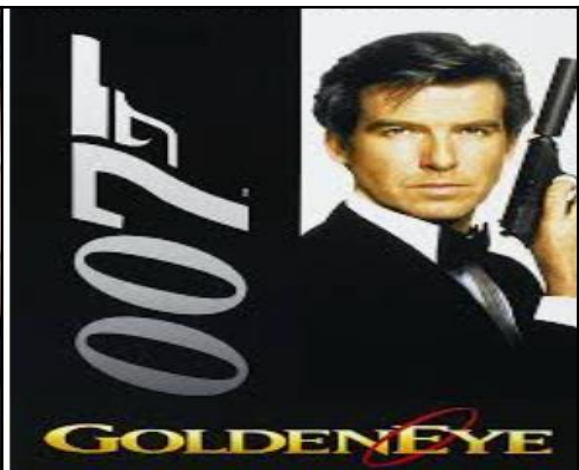

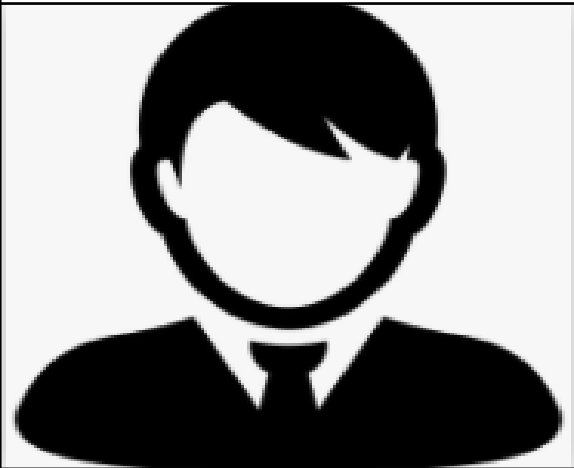



0.81

0.00

0.50

1. Review

Collaborative Filtering

<div>Movies</div> <div>Users</div>				
	1	0	1	1
				
	0	0	1	1
				

Cosine 유사도

0.81

0.00

0.50

1. Review

Neural Collaborative Filtering (2017)

배경: 2009 년에 기존의 Netflix의 추천 시스템의 성능을 10 % 이상 개선해주는 알고리즘이 Netflix 공모전에서 수상작으로 선정 되었다. 이것은 우리가 뒤에서 알아볼 행렬 인수분해 **Matrix Factorization**를 이용한 방법이다. 이것이 소개된 이후, 이 MF는 최근까지도 협업 필터링 방법 중 가장 인기있는 방법으로 여겨진다.



[1] 저자들의 MF에 대한 지적

사용자와 아이템 간의 상호성을 표현하기에 inner product는 너무 단순한 선택이다. 이것은 복잡한 그들 간의 상호성을 포착하기에 충분치 않다.

$$\hat{y}_{ui} = f(u, i | \mathbf{p}_u, \mathbf{q}_i) = \mathbf{p}_u^T \mathbf{q}_i = \sum_{k=1}^K p_{uk} q_{ik},$$

[2] 저자들의 제안

MF 뿐만 아니라, 최근 여러 분야에서 사용되는 DNN을 도입해보자.

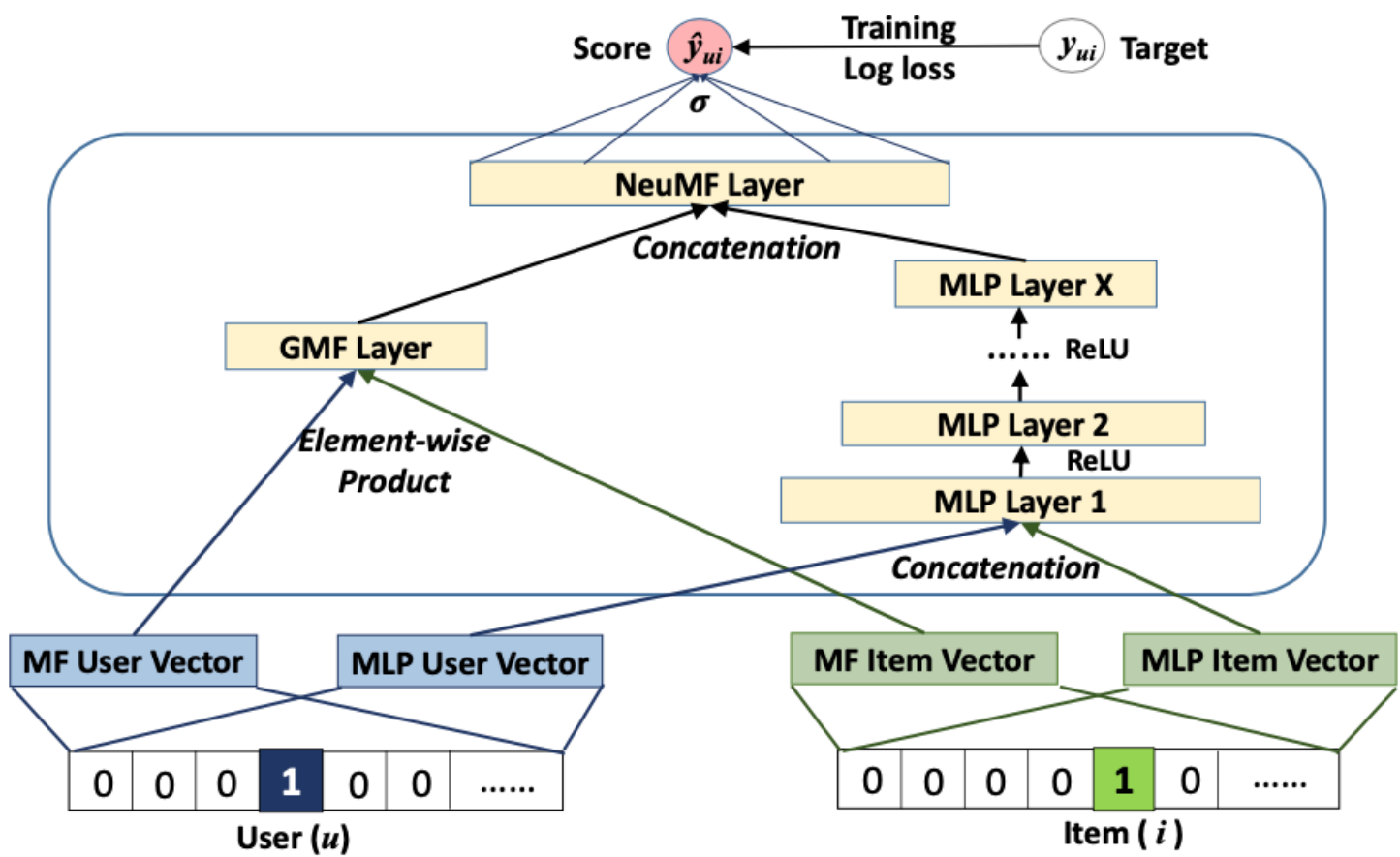


Figure 3: Neural matrix factorization model

진행중인 신경망 협업 필터링

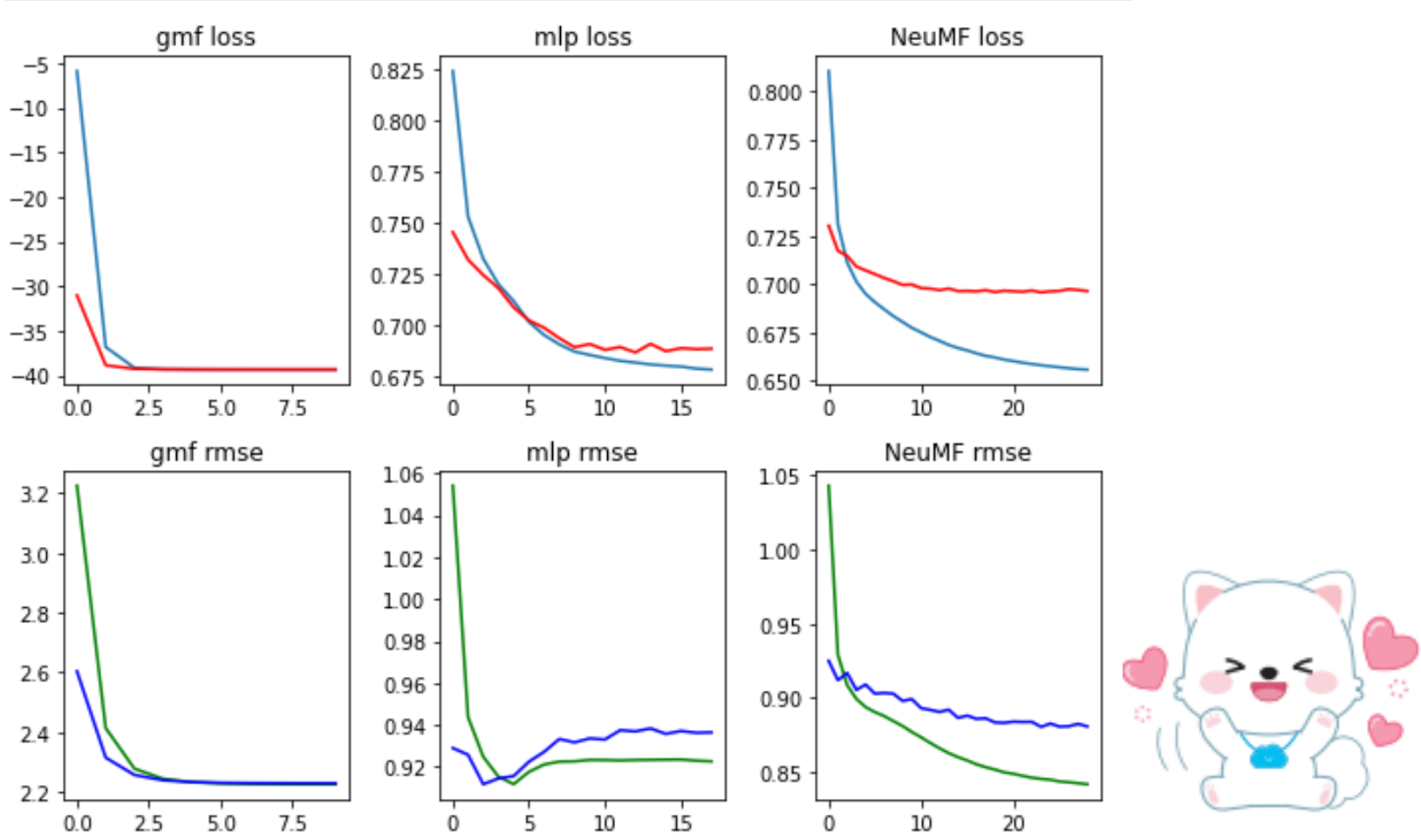
사용 모델: 신경망 협업 필터링의 Movielens 1M 적용 (성능 평가 기준: RMSE 오차율)

RANK	MODEL	RMSE ↓	NDCG@10	HR@10	NDCG	PAPER	CODE	RESULT	YEAR
1	Bayesian timeSVD++ flipped	0.818				On the Difficulty of Evaluating Baselines: A Study on Recommender Systems			2019
2	Sparse FC	0.824				Kernelized Synaptic Weight Matrices			2018
3	CF-NADE	0.829				A Neural Autoregressive Approach to Collaborative Filtering			2016
4	I-AutoRec	0.831				AutoRec: Autoencoders Meet Collaborative Filtering			2015
5	GC-MC	0.832				Graph Convolutional Matrix Completion			2017
6	I-CFN	0.8321				Hybrid Recommender System based on Autoencoders			2016
7	NNMF	0.843				Neural Network Matrix Factorization			2015
8	IGMC	0.857				Inductive Matrix Completion Based on Graph Neural Networks			2019
9	U-CFN	0.8574				Hybrid Recommender System based on Autoencoders			2016
10	Factorized EAE	0.860				Deep Models of Interactions Across Sets			2018
11	RECONST- NET	0.860				Collaborative Filtering with Smooth Reconstruction of the Preference Function			2021
12	Factorization with dictionary learning	0.866				Dictionary Learning for Massive Matrix Factorization			2016

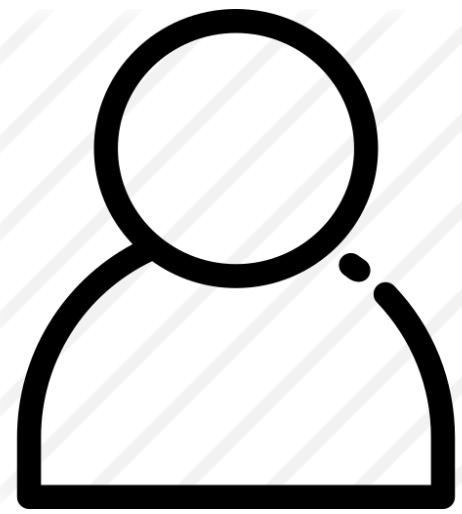
MovieLens 1M 기준 추천 시스템에서 사용되는 모델들 세계 랭킹

RMSE는 오차율이기 때문에 낮을수록 좋다.

실제 실험 결과



Test RMSE:
0.8807



사용자 3

유저가 선호도를 평가한 영화 목록

Hustler, The (1961), 점수:5
Star Wars: Episode VI - Return of the Jedi (1983), 점수:3
Fistful of Dollars, A (1964), 점수:4
Alien (1979), 점수:4
Die Hard (1988), 점수:4
Star Wars: Episode IV - A New Hope (1977), 점수:5
Saving Private Ryan (1998), 점수:5
Jurassic Park (1993), 점수:4
Star Wars: Episode V - The Empire Strikes Back (1980), 점수:2
Raiders of the Lost Ark (1981), 점수:5
Rocky (1976), 점수:5
E.T. the Extra-Terrestrial (1982), 점수:4
Thelma & Louise (1991), 점수:4
Mad Max (1979), 점수:4
King Kong (1933), 점수:4
Jaws (1975), 점수:5
Predator (1987), 점수:1
Good, The Bad and The Ugly, The (1966), 점수:5
Run Lola Run (Lola rennt) (1998), 점수:5
Goldfinger (1964), 점수:5
Terminator, The (1984), 점수:5

추천 시스템에 의해 추천된 영화 목록

American Beauty (1999)
Shawshank Redemption, The (1994)
Pulp Fiction (1994)
Usual Suspects, The (1995)
Reservoir Dogs (1992)

2. NGCF Neural Graph Collaborative Filtering (2019)

CF: Collaborative Filtering

CF는 행동적으로 유사한 사용자들이 특정 아이템에 대해 비슷한 선호를 보이는 경향이 있을 것임을 가정한다. 이 Learnable CF에는 2 가지 주요 요소가 있다.

- ① Embedding: 유저와 아이템을 벡터로 표현
- ② Interaction Modeling: 위 임베딩을 기반으로 historical interaction을 재구축

(ex1) MF는 u/i ID를 vector로 임베딩, 그 뒤 inner product로 그것들의 interaction을 표현
(ex2) NCF는 MF의 inner product를 NonLinear Neural Net으로 대체

저자의 주장:

1. [기존 방법이 가진 불명료성]

위 사례에서 소개된 방법들은 만족스럽지 않다. 왜냐하면 Embedding 함수는 중요한 Collaborative Signal의 명시적인 인코딩을 결여하기 때문이다. 가령, 우리는 임베딩 함수에서 user1, user2와 item1 간의 연결 여부 및 연결 방향에 대해 알 수 없다. 위 사례에서 소개된 방법들에 따르면, 이 Collaborative Signal은 users(or items) 간 행동적 유사성 behaviour similarity를 드러내는 u-i (between users and items) interaction에서 잠재적latent일 뿐이다.

2. [기존 방법이 가진 곤경]

u-i(between users and items) Interaction을 Embedding으로 통합시키는 것은 유용해 보이는 반면에, 이것은 항상 성공적이지 않다. 왜냐하면 실제 사용에서 Interaction Matrix가 매우 큰 경우에, 우리가 원하는 Collaborative Signal을 추출하기 어렵게 되기 때문이다.

⇒ 우리는 고차 연결성 High order Connectivity을 이용함으로써, 이 도전을 다룰 것이다. 이 고차 연결성은 Collaborative Signal 을 Interaction Graph Structure로 인코딩하는 방법이다.

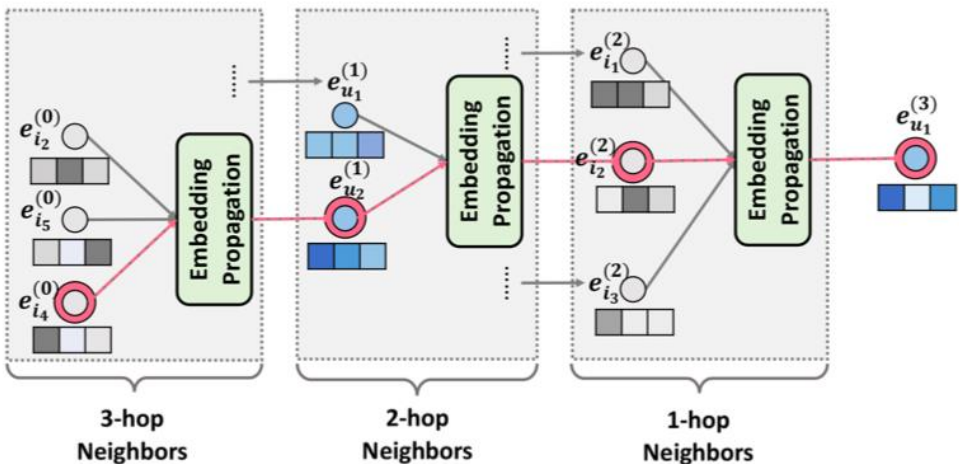
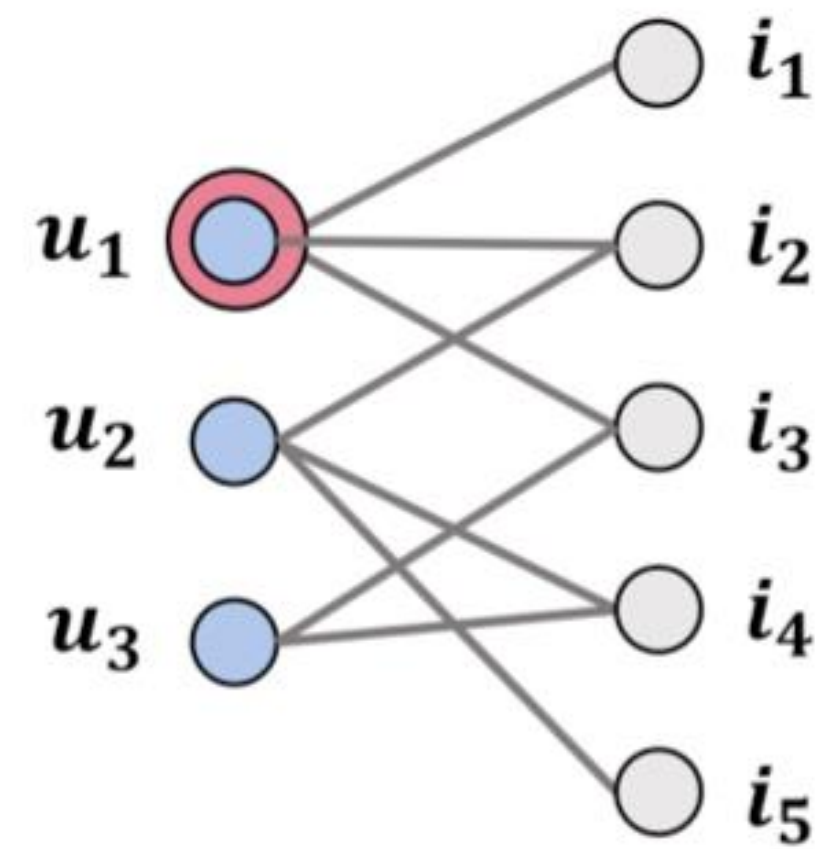
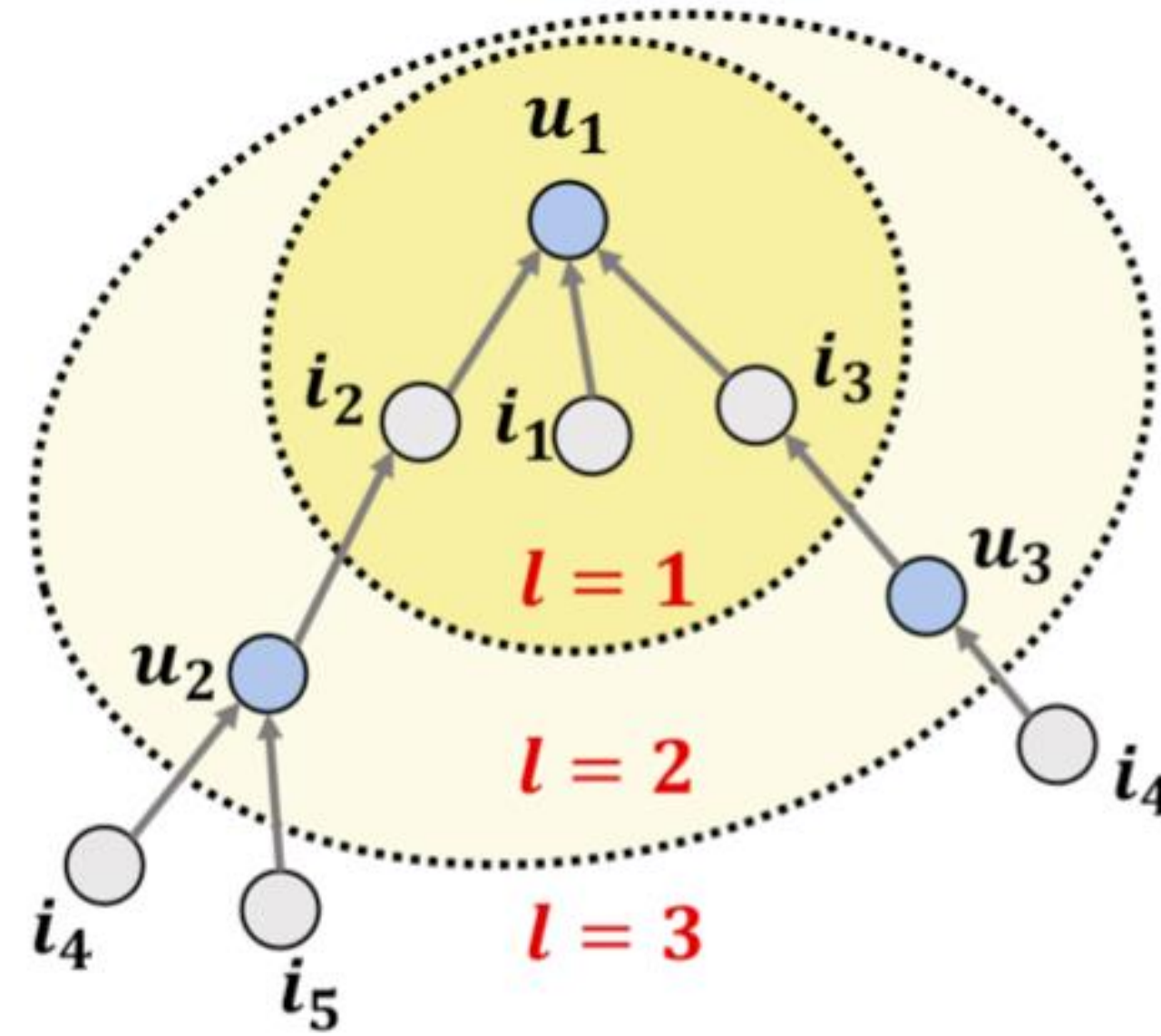


Figure 3: Illustration of third-order embedding propagation for user u_1 . Best view in color.

2. NGCF Neural Graph Collaborative Filtering (2019)



User-Item Interaction Graph

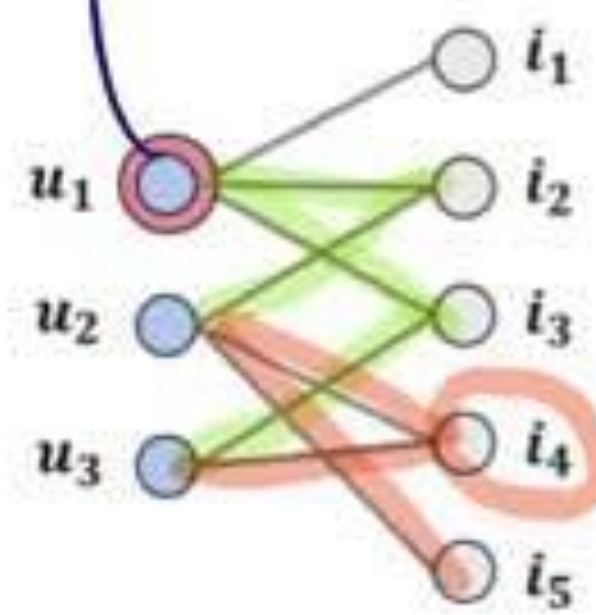


High-order Connectivity for u_1

Figure 1: An illustration of the user-item interaction graph and the high-order connectivity. The node u_1 is the target user to provide recommendations for.

2. NGCF Neural Graph Collaborative Filtering (2019)

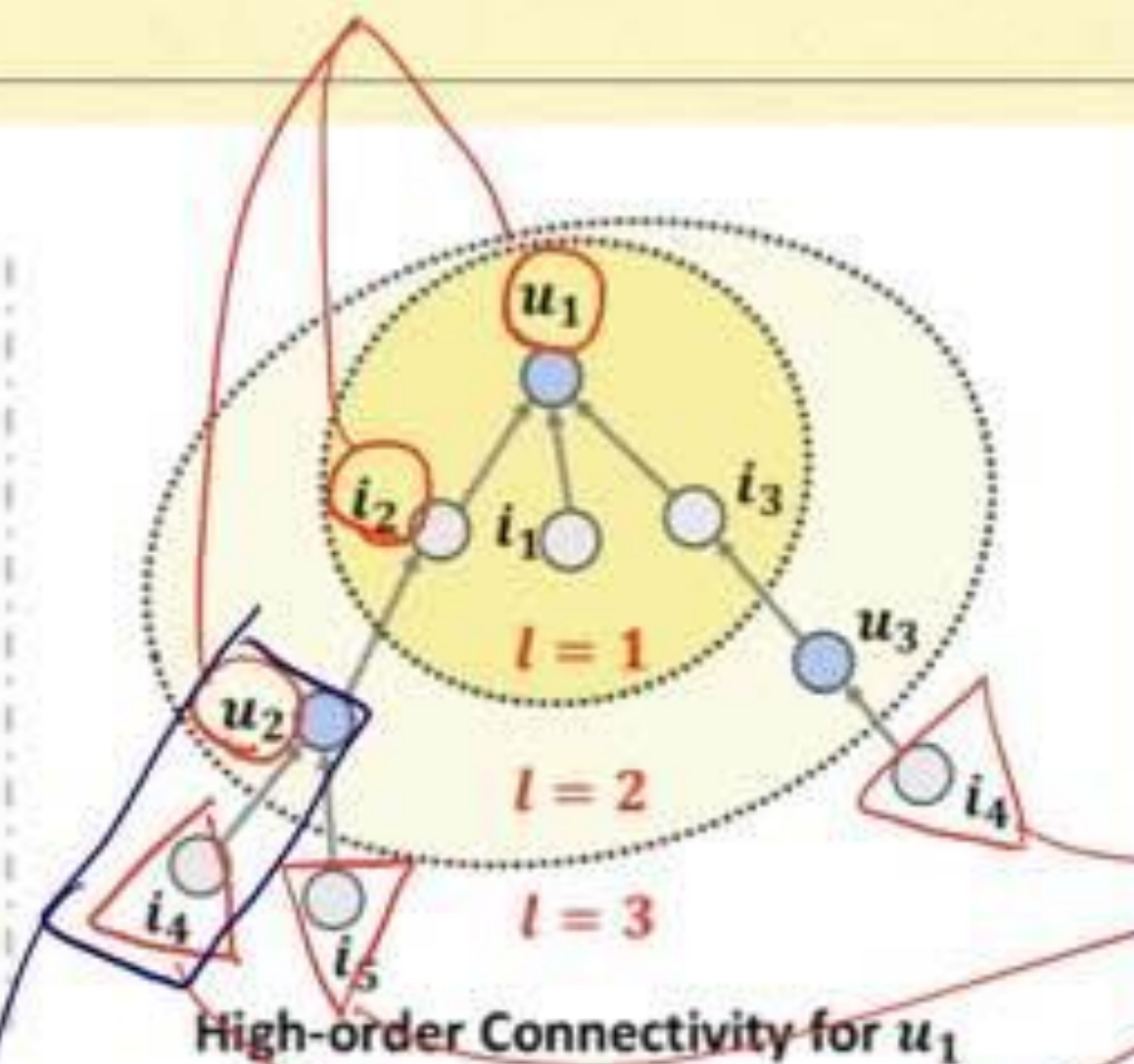
item 추천을 받고자 하는 u_1 .



User-Item Interaction Graph

Figure 1: An illustration of the user-item interaction graph and the high-order connectivity. The node u_1 is the target user to provide recommendations for.

* u_1, u_2 는 i_2 와 interact.



u_2 가 i_4 를 이전에 구매했기 있어.

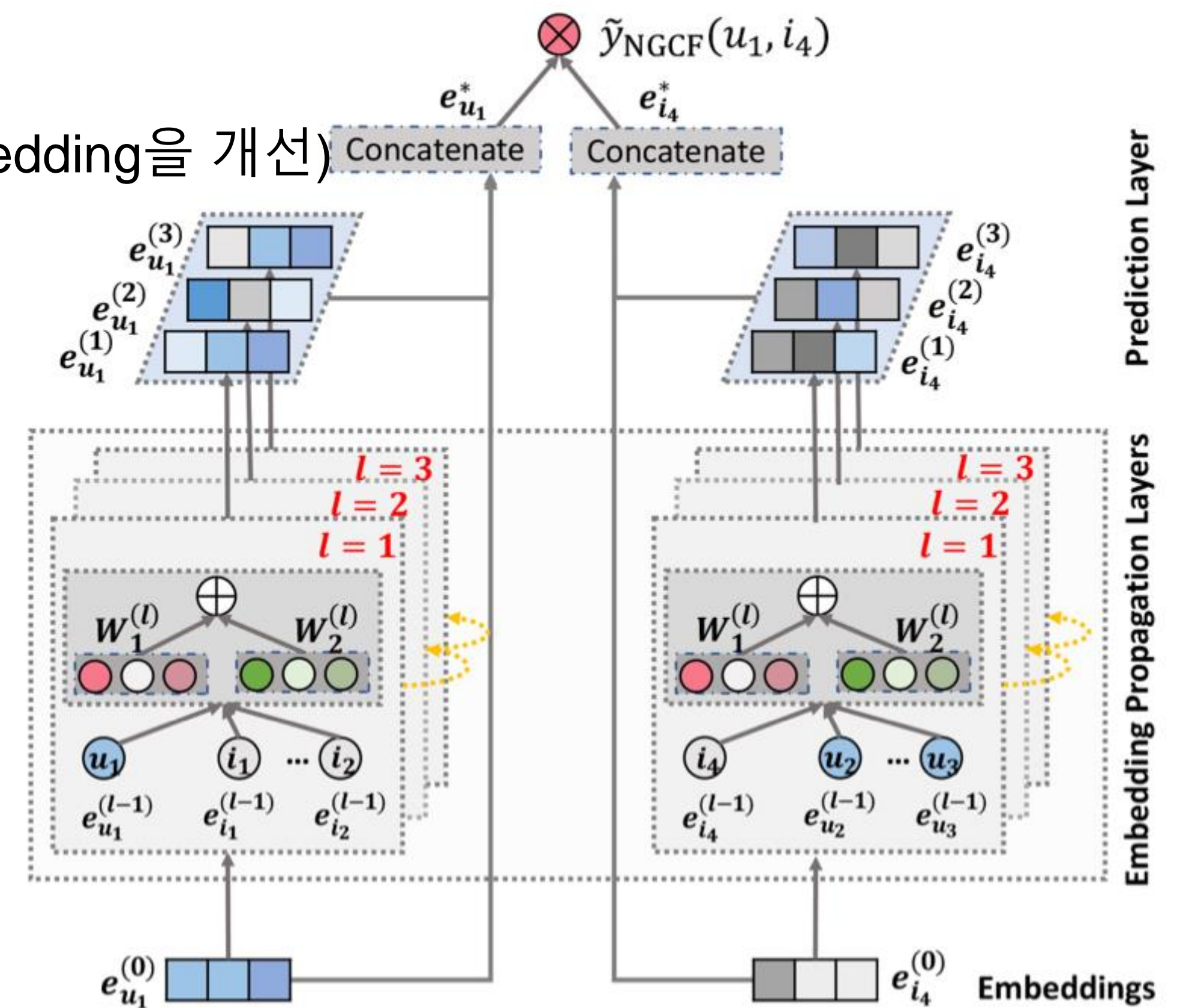
* 1 이상의 길이를 가진 모든 node에서 u_1 에 도달.

i_4 는 2번, i_5 는 1번 u_1 에 연결됨.
 $\Rightarrow i_4$ 가 u_1 에 i_5 보다 더 가깝다.

2. NGCF Neural Graph Collaborative Filtering (2019)

3 가지 주요 요소

1. Embedding Layer (User Embedding, Item Embedding)
2. 여러 개의 Embedding Propagation Layers (고차 연결성을 부여하여 Embedding을 개선)
3. Prediction Layers

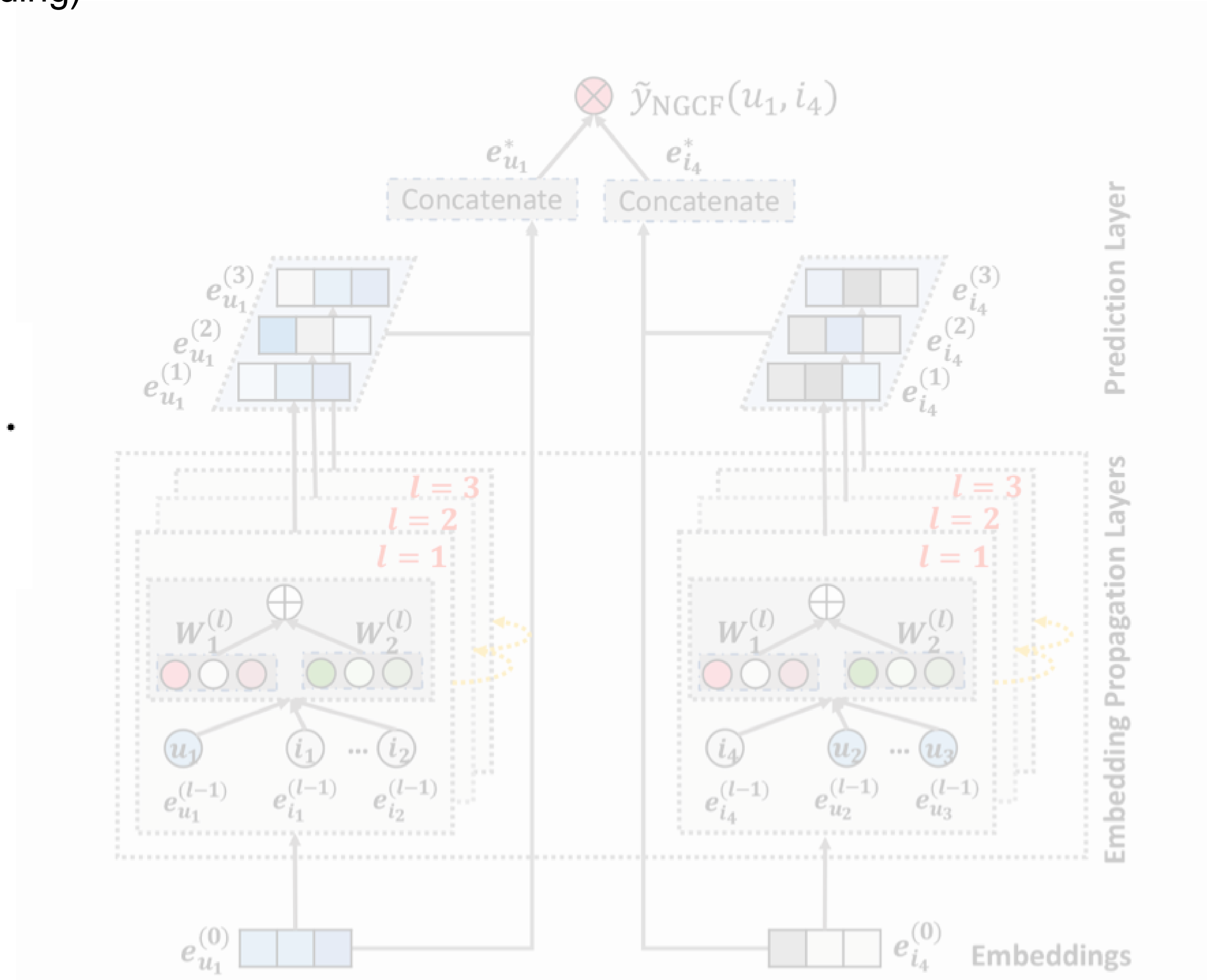


2. NGCF Neural Graph Collaborative Filtering (2019)

1. Embedding Layer (User Embedding, Item Embedding)

embedding vector $\mathbf{e}_u \in \mathbb{R}^d$ ($\mathbf{e}_i \in \mathbb{R}^d$),

$$\mathbf{E} = \left[\underbrace{\mathbf{e}_{u_1}, \dots, \mathbf{e}_{u_N}}_{\text{users embeddings}}, \underbrace{\mathbf{e}_{i_1}, \dots, \mathbf{e}_{i_M}}_{\text{item embeddings}} \right].$$



2. NGCF Neural Graph Collaborative Filtering (2019)

1. Embedding Layer (User Embedding, Item Embedding)

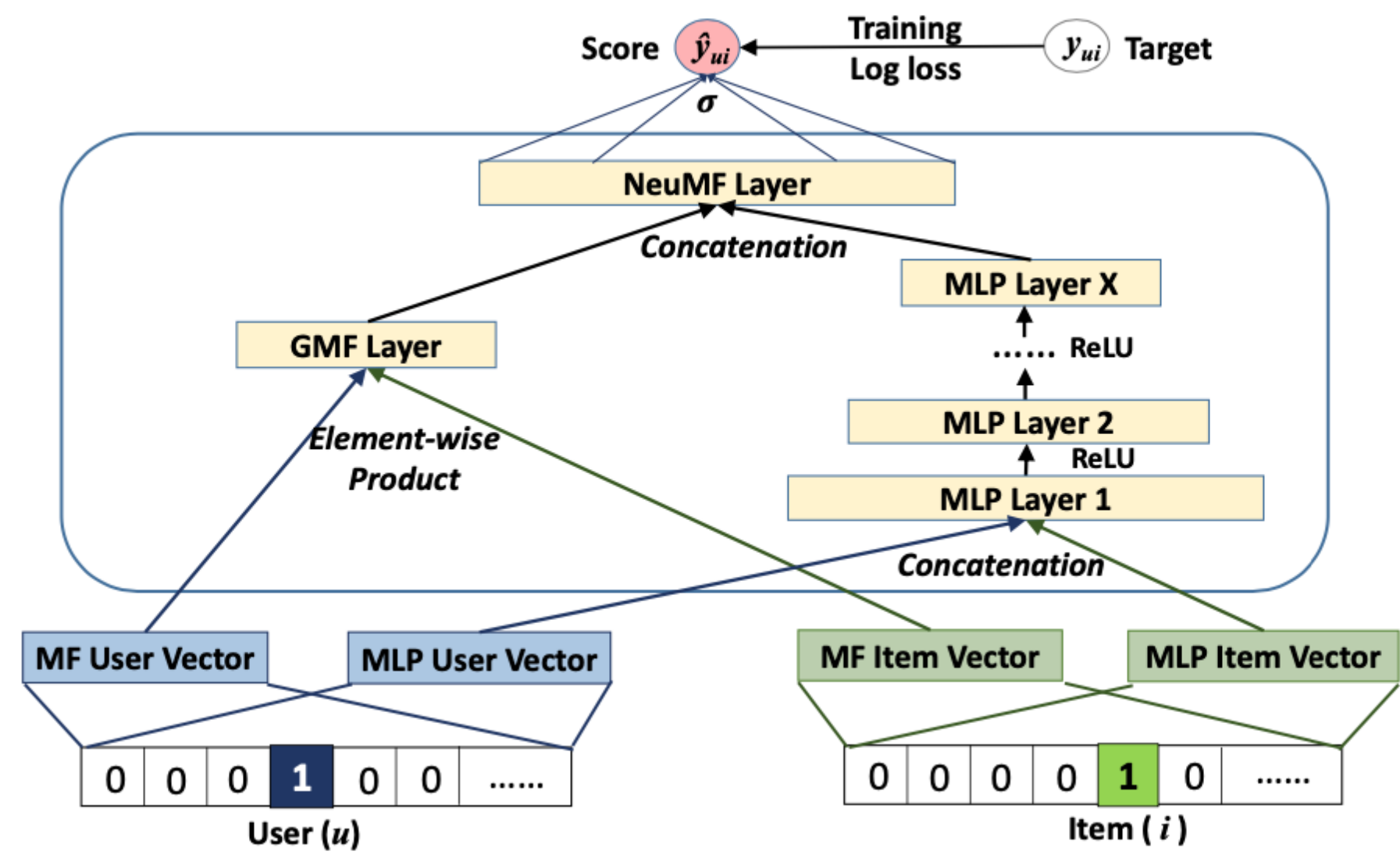
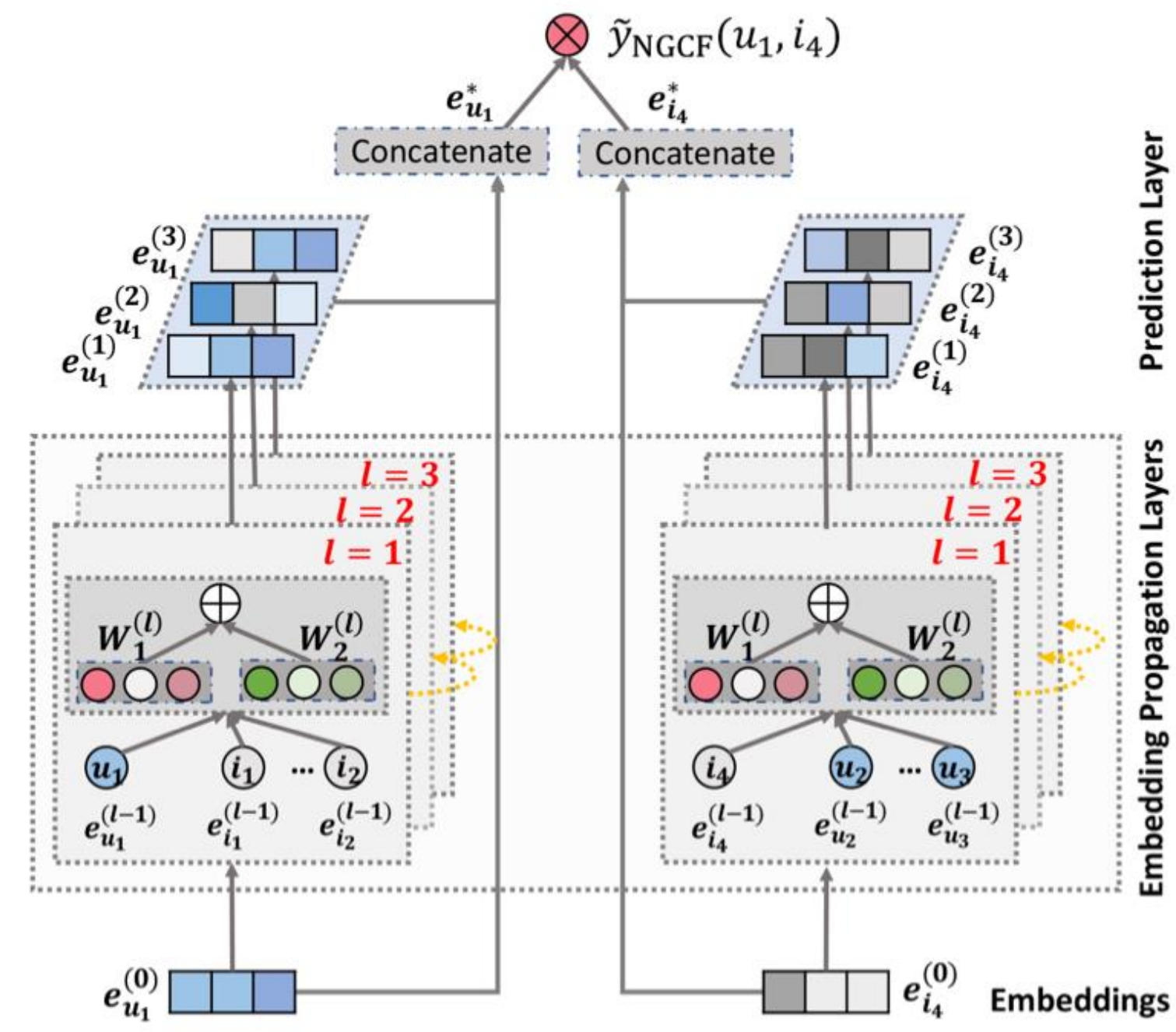


Figure 3: Neural matrix factorization model

NCF 2017



NGCF 2019

NGCF에서는 Embedding 개선 step이 Collaborative Signal을 Embedding에 명시적으로 주입하기 때문에 추천에 대해 더 효과적

2. NGCF Neural Graph Collaborative Filtering (2019)

2. Embedding Propagation Layers: User-item Embedding을 개선하기 위해, GNN 아키텍처 사용

(1) **First-Order Propagation** **장점:** Emb Propagation Layers는 user-item representation과 관련된 First-Order 연결 정보를 명시적으로 이용한다.

- Interacted items는 users의 선호에 대한 직접적인 증거가 될 수 있다.
- 동일하게, 이 items을 소비하는 users는 item의 특징으로 간주될 수 있고, 2 개의 items 간 Collaborative Similarity를 측정하는 데 사용될 수 있다.

* User-item Embedding Propagation을 수행하기 위한 기초들

** Message Construction

a connected user-item pair (u,i)

$$m_{u \leftarrow i} = \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}} \left(W_1 e_i + W_2 (e_i \odot e_u) \right), \tag{3}$$

$$m_{u \leftarrow i} = \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}} \left(W_1 e_i + W_2 (e_i \odot e_u) \right),$$

(Handwritten notes: item과 user의 연관성을 고려하여 전달되는 값, item에서 순수하게 전달되는 값)

Message: 어떤 정보가 이동하는 흐름
Ex) 어떤 사용자가 item을 소비했다면, 그 item의 정보가 user에게로 흐른다.

** Message Aggregation

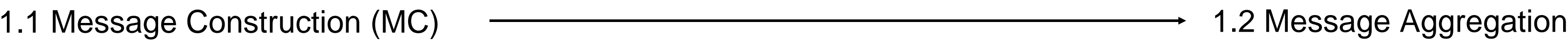
$$e_u^{(1)} = \text{LeakyReLU} \left(m_{u \leftarrow u} + \sum_{i \in \mathcal{N}_u} m_{u \leftarrow i} \right),$$

(Handwritten notes: m_{u \leftarrow u} = W_1 e_u, user's preference information; 1st emb propagation layer 이후에 얻은 user의 representation)

2. NGCF Neural Graph Collaborative Filtering (2019)

2. Embedding Propagation Layers: User-item Embedding을 개선하기 위해, GNN 아키텍처 사용

(1) First-Order Propagation



a connected user-item pair (u,i)

$$\mathbf{m}_{u \leftarrow i} = f(\mathbf{e}_i, \mathbf{e}_u, p_{ui}),$$

$\mathbf{m}_{u \leftarrow i}$: message Embedding
 $\mathbf{e}_i, \mathbf{e}_u$: Input
 p_{ui} : 각 propagation의 decay factor를 지니는 coefficient.

$$\mathbf{m}_{u \leftarrow i} = \frac{1}{\sqrt{|N_u||N_i|}} (\mathbf{W}_1 \mathbf{e}_i + \mathbf{W}_2 (\mathbf{e}_i \odot \mathbf{e}_u))$$

$\mathbf{W}_1, \mathbf{W}_2$: Trainable Weights
 $\mathbf{e}_i, \mathbf{e}_u$: interaction
 $\mathbf{e}_i \odot \mathbf{e}_u$: \mathbf{e}_i 와 \mathbf{e}_u 에 관련된 message로 인코딩 됨.
 $p_{ui} = \frac{1}{\sqrt{|N_u||N_i|}}$: historical item이 users 신호에 얼마나 많이 영향을 주는가를 반영.

$$\mathbf{e}_u^{(1)} = \text{LeakyReLU} \left(\mathbf{m}_{u \leftarrow u} + \sum_{i \in N_u} \mathbf{m}_{u \leftarrow i} \right),$$

$\mathbf{e}_u^{(1)}$: 1st emb propagation layer 이후에 얻은 user의 representation
 $\mathbf{m}_{u \leftarrow u} = \mathbf{W}_1 \mathbf{e}_u$: 원래 features의 정보 보충

- $\mathbf{e}_i, \mathbf{e}_u$ 의 interaction은 $\mathbf{e}_i \odot \mathbf{e}_u$ 연산에 의해 message로 인코딩된다.
 - $\mathbf{P}_{ui} = \frac{1}{\sqrt{|N_u||N_i|}}$ where N_u, N_i denote the first-hop neighbors of user u and item i
 - 이 MC 과정은 위 식 같이, (1) item 정보의 전달, (2) u-i의 연관성 정보 전달로 이루어진다.
- MC 과정 이후, 1hop의 이웃 정보들은 모두 합산된다.
 - 위 합산 결과는 MA 과정에서 위 식 같이 user 정보 전달과 더해진 뒤, activation을 지난다.

2. NGCF Neural Graph Collaborative Filtering (2019)

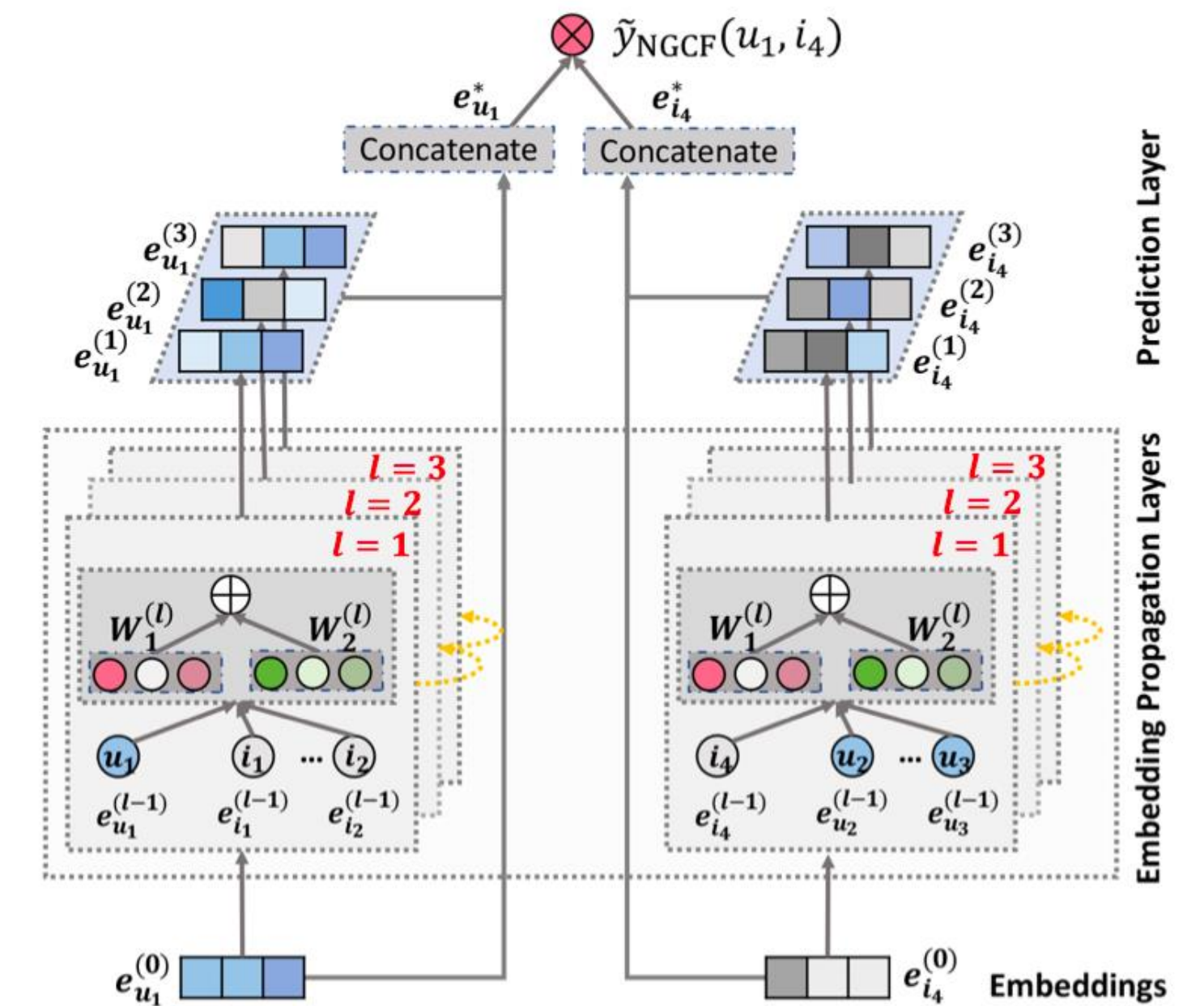
2. Embedding Propagation Layers: User-item Embedding을 개선하기 위해, GNN 아키텍처 사용

(2) High-Order Propagation

First-Order Propagation을 여러 개 쌓으면 High-Order Propagation을 만들 수 있다. 이런 **고차 연결**은 user-item 간 relevance score를 추정하는 협업 신호 Collaborative Singal를 인코딩하는 데 중요하다.

L 개의 Emb Prop Layers를 쌓음으로써, 어떤 한 유저는 L 번째 hop neighbors로부터 전파된 메시지를 받을 수 있다.

$$\mathbf{e}_u^{(l)} = \text{LeakyReLU}\left(\mathbf{m}_{u \leftarrow u}^{(l)} + \sum_{i \in \mathcal{N}_u} \mathbf{m}_{u \leftarrow i}^{(l)}\right),$$
$$\begin{cases} \mathbf{m}_{u \leftarrow i}^{(l)} = p_{ui} \left(\mathbf{W}_1^{(l)} \mathbf{e}_i^{(l-1)} + \mathbf{W}_2^{(l)} (\mathbf{e}_i^{(l-1)} \odot \mathbf{e}_u^{(l-1)}) \right), \\ \mathbf{m}_{u \leftarrow u}^{(l)} = \mathbf{W}_1^{(l)} \mathbf{e}_u^{(l-1)}, \end{cases}$$



2. NGCF Neural Graph Collaborative Filtering (2019)

2. Embedding Propagation Layers: User-item Embedding을 개선하기 위해, GNN 아키텍처 사용

(2) High-Order Propagation

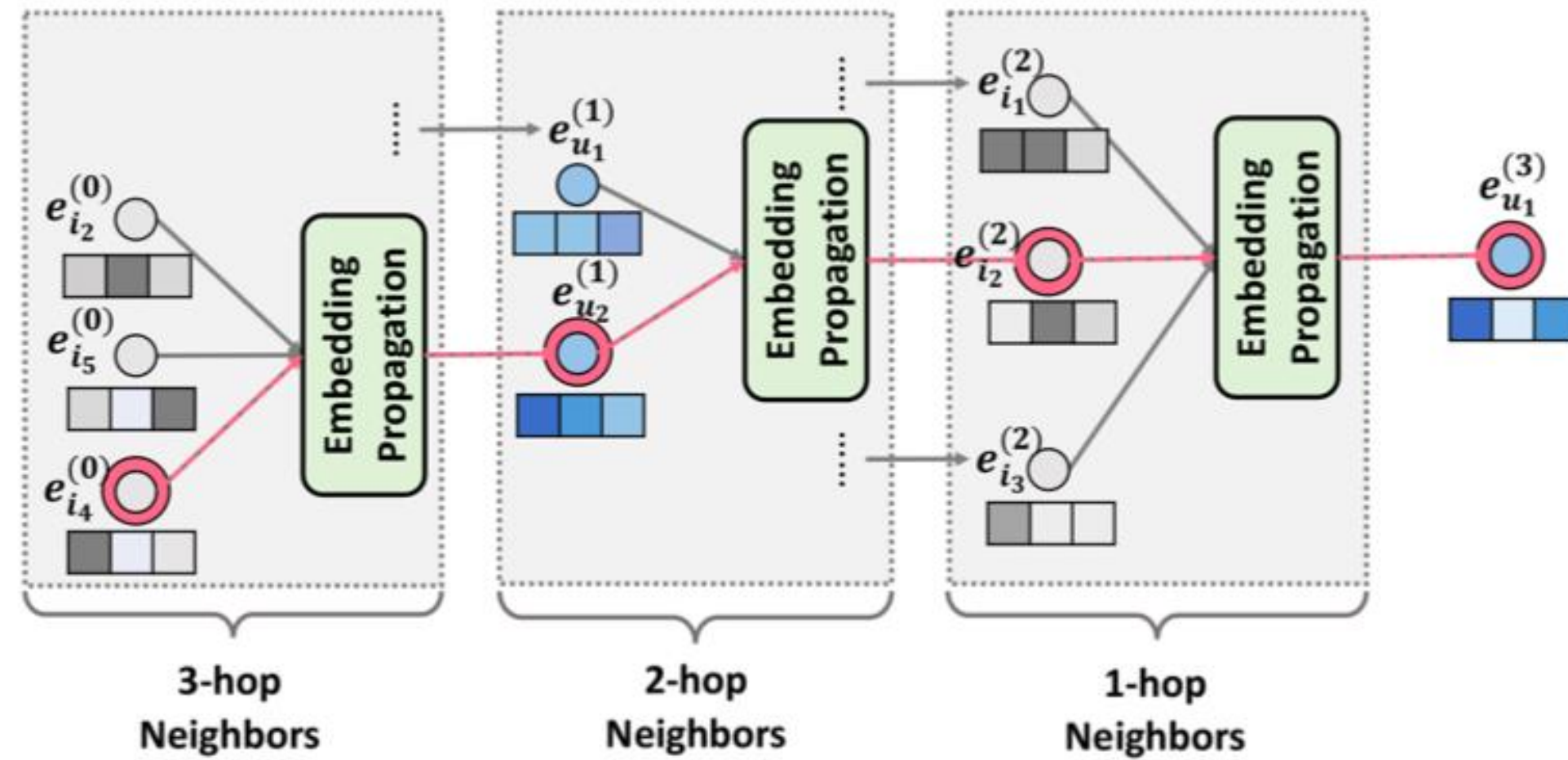


Figure 3: Illustration of third-order embedding propagation for user u_1 . Best view in color.

Collaborative Signal

$i_4 \rightarrow u_2 \rightarrow i_2 \rightarrow u_1$

여기서, i_4 는 명시적으로 u_1 에 인코딩된다.

2. NGCF Neural Graph Collaborative Filtering (2019)

2. Embedding Propagation Layers: User-item Embedding을 개선하기 위해, GNN 아키텍처 사용

Batch Implementation을 이용하기 위해, 앞에서 본 식들을 Matrix Form으로 표현해보자.

$$\mathbf{E}^{(l)} = \text{LeakyReLU}\left((\mathcal{L} + \mathbf{I})\mathbf{E}^{(l-1)}\mathbf{W}_1^{(l)} + \mathcal{L}\mathbf{E}^{(l-1)} \odot \mathbf{E}^{(l-1)}\mathbf{W}_2^{(l)}\right),$$

where $\mathbf{E}^{(l)} \in \mathbb{R}^{(N+M) \times d_l}$

I: identity Matrix

L: user-item graph에 대한 Laplacian Matrix

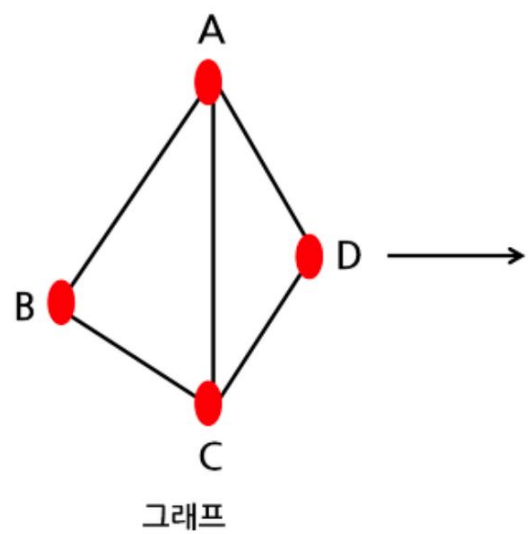
$$\mathcal{L} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \text{ and } \mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^\top & \mathbf{0} \end{bmatrix},$$

A = adjacent matrix

* R = user-item interaction matrix

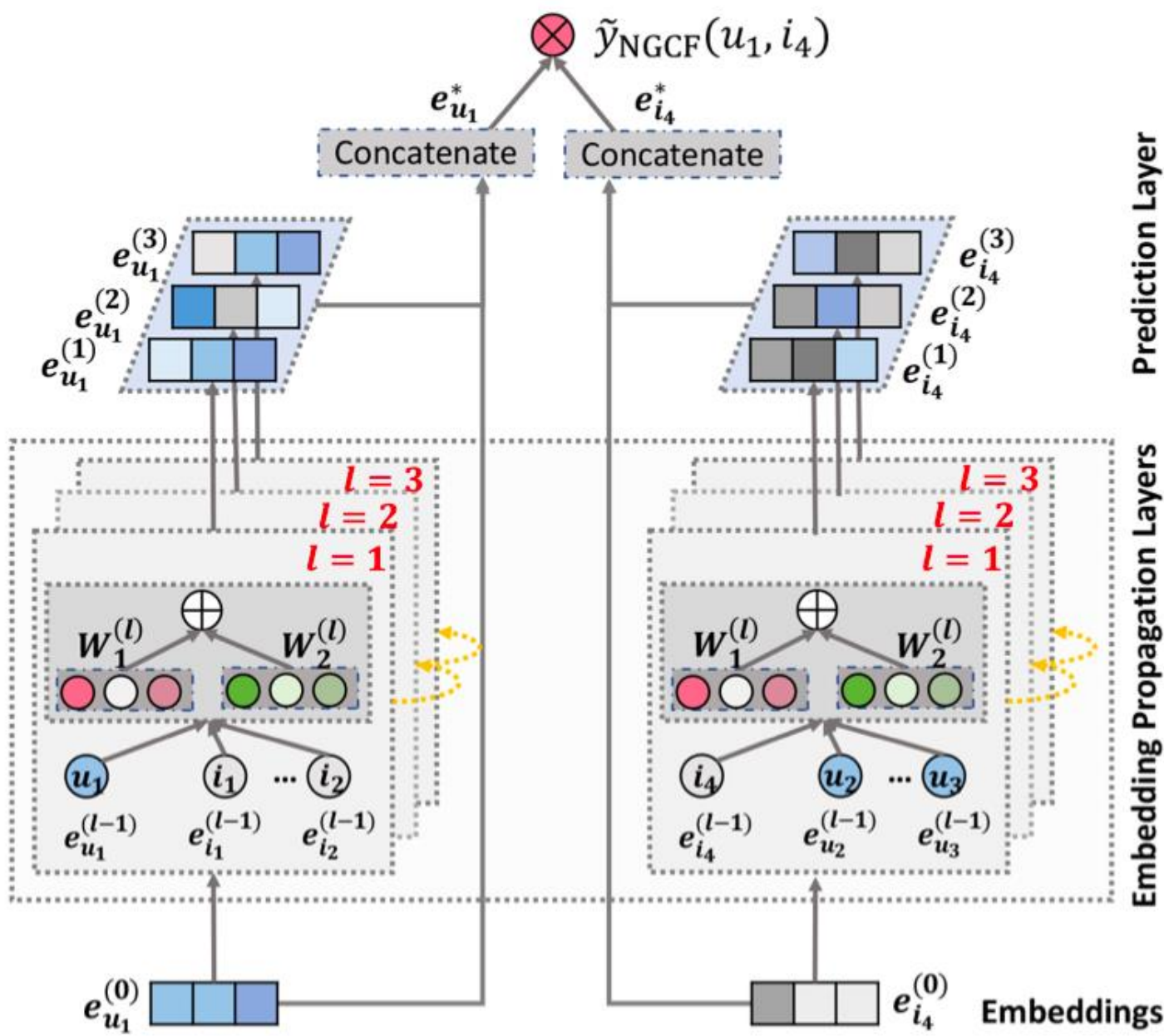
D = diagonal degree matrix

인접행렬:
그래프의 각 꼭지점의 연결관계를 나타낸 행렬



	A	B	C	D
A	0	1	1	1
B	1	0	1	0
C	1	1	0	1
D	1	0	1	0

인접행렬



2. NGCF Neural Graph Collaborative Filtering (2019)

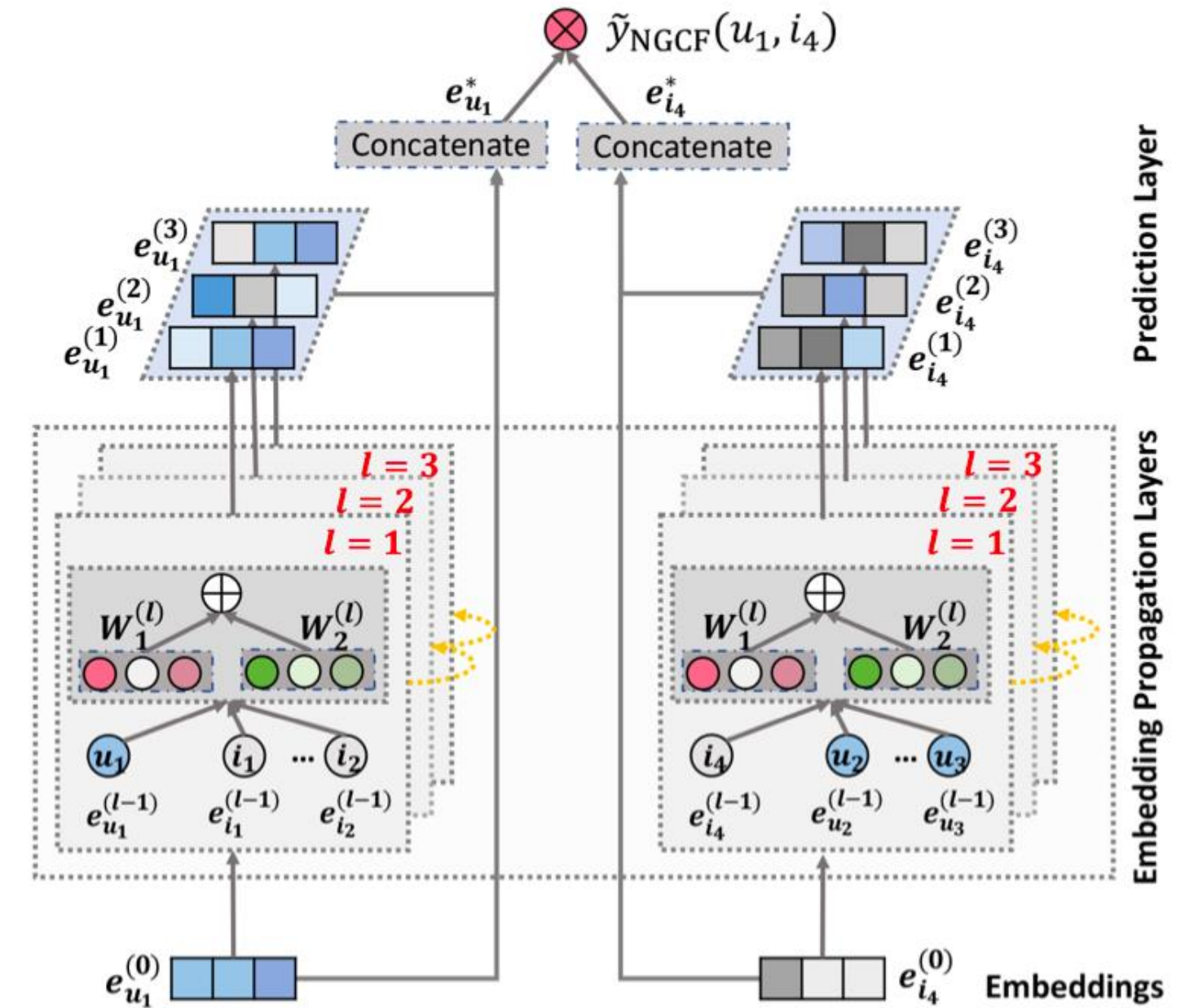
3. Model Prediction

Final Layers: Concatenated item embedding, user embedding

$$\mathbf{e}_u^* = \mathbf{e}_u^{(0)} \parallel \cdots \parallel \mathbf{e}_u^{(L)}, \quad \mathbf{e}_i^* = \mathbf{e}_i^{(0)} \parallel \cdots \parallel \mathbf{e}_i^{(L)},$$

Target item에 대한 유저의 선호를 추정하기 위해 inner product

$$\hat{y}_{\text{NGCF}}(u, i) = \mathbf{e}_u^{*T} \mathbf{e}_i^*.$$



2. NGCF Neural Graph Collaborative Filtering (2019)

4. Optimization

BPR loss: 추천 시스템에서 널리 사용되는 방법

$$Loss = \sum_{(u,i,j) \in O} -\ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda \|\Theta\|_2^2,$$

모델은 (u,i,j) triplet의 배치로 학습된다.
유저 u, 아이템 i는 interact 하지만, u와 다른 아이템 j는 interact 하지 않는다.

2. NGCF Neural Graph Collaborative Filtering (2019)

배경

Implicit Feedback: r_{ij} ex) 클릭 수, 재생 시간

For user - item, Observed set: $S = (u, i) \in U * I : r_{ui} > 0$

For only item, $I_u^+ (positive_item) = i \in I : (u, i) \in S$ # 유저에게 피드백을 받은 아이템

For only item, $I_u^- (negative_item) = i \in I : (u, i) \in S^c$ # 유저에게 피드백을 받지 못한 아이템.

단, except $r_{ui} = 0$

➡ 집합 $D_s = \{(u, i, j), i \in I_u^+, j \in I_u^-\}$

$P(i >_u j)$ user가 j 보다 i를 더 선호할 확률

$P(j >_u i)$ user가 i 보다 j를 더 선호할 확률

2. NGCF Neural Graph Collaborative Filtering (2019)

배경

➡ 집합 $D_s = \{(u, i, j), i \in I_u^+, j \in I_u^-\}$

어떻게 user가 선호하는 item에 대한 확률을 구할 수 있을까?

$P(i >_u j)$ user가 j 보다 i를 더 선호할 확률

$P(j >_u i)$ user가 i 보다 j를 더 선호할 확률

➡
$$P(i >_u j) = \sigma(\hat{y}_{ui} - \hat{y}_{uj})$$

시그모이드 함수의 역할: 베르누이 분포화

$$\sigma(x) + \sigma(-x) = 1$$

$$P(i >_u j) + P(j >_u i) = 1$$

2. NGCF Neural Graph Collaborative Filtering (2019)

5. Evaluation

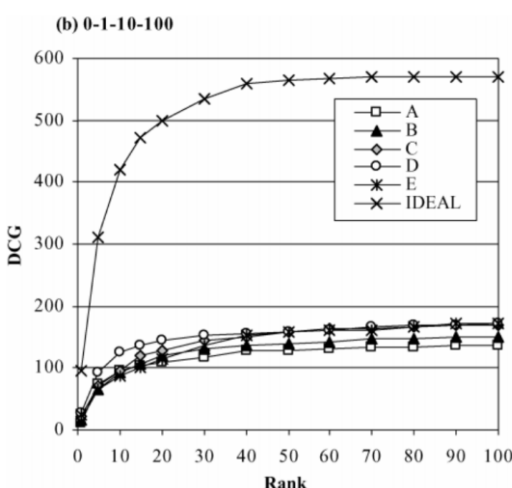
NDCG : Normalized Discounted Culmulative Gain추천을 받는 대상과 관련성이 높은 아이টে을 잘 예측했는지를 보여주는 평가 지표

$$CG_p = \sum_{i=1}^p rel_i$$

CG Culmulative Gain

$$DCG_p = \sum_{i=1}^p \frac{rel_i}{\log_2(i+1)}$$
$$DCG_p = \sum_{i=1}^p \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

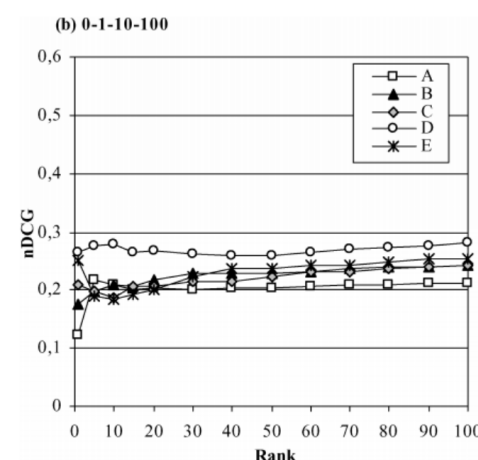
DCG Discounted Culmulative Gain



$$nDCG_p = \frac{DCG_p}{IDCG_p}$$

$$IDCG_p = \sum_{i=1}^p \frac{rel_i^{opt}}{\log_2(i+1)}$$

nDCG normalized Discounted Culmulative Gain



상위 p 개의 추천 결과들의 관련성 점수rel을 합한 누적값 기존의 CG에서 랭킹 순서에 따라 비중을 줄여discounted 관련성 점수를 계산
CG는 상위 p 개의 추천 결과를 모두 동일한 비중으로 계산 하위권으로 갈수록 동일한 rel 대비 작은 DCG를 갖게 된다.

기존의 DCG는 p의 길이에 따라 DCG가 많이 커질 수 있다. nDCG는 p 값에 상관없이 일정한 스케일링 값을 가질 수 있도록 normalization을 취한 것이다. 그래서 이 값은 0, 1 사이에서 얻어진다.
위 식에서 IDCG는 ideal DCG이다. 이 IDCG는 p 개의 관련성 점수 rel이 큰 순서대로 재배열된 경우에 DCG를 계산한 값

[EXAMPLE]

추천 시스템 S1, S2가 있다고 해보자. 각 추천 시스템은 5 개의 음악을 사용자에게 추천해준다. 여기서 관련성 점수 rel은 0 - 3 내의 정수 값을 갖는다.

- S1 이 선택한 음악: {A, E, C, D, F} => {3, 1, 2, 2, 1}
- S2 가 선택한 음악: {A, B, C, G, E} => {3, 3, 2, 0, 1}
- 각 음악들의 관련도: {A=3, B=3, C=2, D=2, E=1, F=1, G=0, 나머지 음악=0}

Ideal Gain Vector: $I = < 3, 3, 2, 2, 1, 1, 0, 0, 0, \dots >$

(1) DCG로 계산한 경우

$$S1) \ DCG_5 = \frac{3}{\log_2(2)} + \frac{1}{\log_2(3)} + \frac{2}{\log_2(4)} + \frac{2}{\log_2(5)} + \frac{1}{\log_2(6)}$$
$$= 3 + 0.631 + 1 + 0.861 + 0.387 = 5.879$$
$$S2) \ DCG_5 = \frac{3}{\log_2(2)} + \frac{3}{\log_2(3)} + \frac{2}{\log_2(4)} + \frac{0}{\log_2(5)} + \frac{1}{\log_2(6)}$$
$$= 3 + 1.893 + 1 + 0 + 0.387 = 6.280$$

(1.5) IDCG로 계산한 경우

$$IDCG_5 = \frac{3}{\log_2(2)} + \frac{3}{\log_2(3)} + \frac{2}{\log_2(4)} + \frac{2}{\log_2(5)} + \frac{1}{\log_2(6)}$$
$$= 3 + 1.893 + 1 + 0.861 + 0.387 = 7.141$$

(2) NDCG로 계산한 경우

$$S1) \ nDCG_5 = \frac{DCG_5}{IDCG_5} = \frac{5.879}{7.141} = 0.823$$
$$S2) \ nDCG_5 = \frac{DCG_5}{IDCG_5} = \frac{6.280}{7.141} = 0.879$$

2. NGCF Neural Graph Collaborative Filtering (2019)

6. Experiments

- **RQ1:** How does NGCF perform as compared with state-of-the-art CF methods?
- **RQ2:** How do different hyper-parameter settings (*e.g.*, depth of layer, embedding propagation layer, layer-aggregation mechanism, message dropout, and node dropout) affect NGCF?
- **RQ3:** How do the representations benefit from the high-order connectivity?

Datasets

- Gowalla
- Yelp2018
- Amazon-book

Table 1: Statistics of the datasets.				
Dataset	#Users	#Items	#Interactions	Density
Gowalla	29, 858	40, 981	1, 027, 370	0.00084
Yelp2018*	31, 668	38, 048	1, 561, 406	0.00130
Amazon-Book	52, 643	91, 599	2, 984, 108	0.00062

Train: val: test = 80: 10: 10

Evaluation Metrics

- K = 20 (추천 리스트 20 개)
 - Recall@K
 - NDCG@K

RQ 1 Answer

Table 2: Overall Performance Comparison.						
	Gowalla		Yelp2018*		Amazon-Book	
	recall	ndcg	recall	ndcg	recall	ndcg
MF	0.1291	0.1109	0.0433	0.0354	0.0250	0.0196
NeuMF	0.1399	0.1212	0.0451	0.0363	0.0258	0.0200
CMN	0.1405	0.1221	0.0457	0.0369	0.0267	0.0218
HOP-Rec	0.1399	0.1214	0.0517	0.0428	0.0309	0.0232
GC-MC	0.1395	0.1204	0.0462	0.0379	0.0288	0.0224
PinSage	0.1380	0.1196	0.0471	0.0393	0.0282	0.0219
NGCF-3	0.1569*	0.1327*	0.0579*	0.0477*	0.0337*	0.0261*
%Improv.	11.68%	8.64%	11.97%	11.29%	9.61%	12.50%
p-value	2.01e-7	3.03e-3	5.34e-3	4.62e-4	3.48e-5	1.26e-4

유의미한 성능 개선 

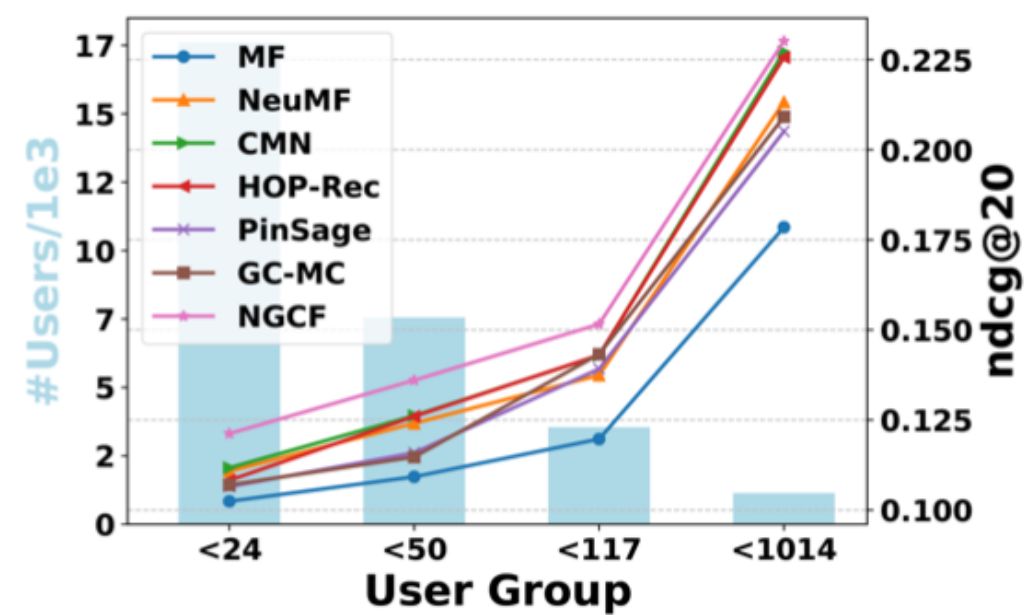
2. NGCF Neural Graph Collaborative Filtering (2019)

6. Experiments

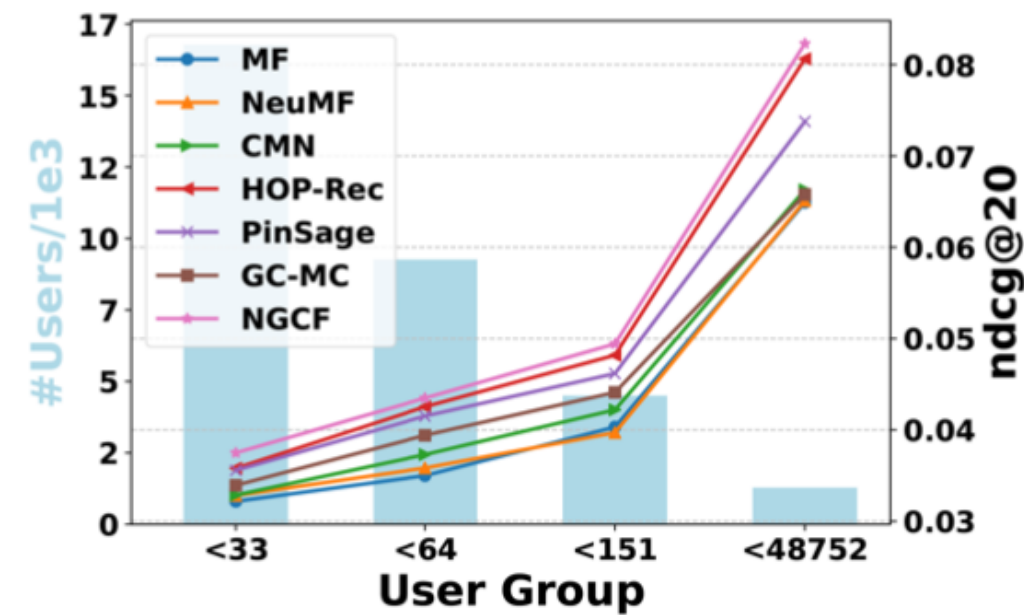
- **RQ1:** How does NGCF perform as compared with state-of-the-art CF methods?
- **RQ2:** How do different hyper-parameter settings (*e.g.*, depth of layer, embedding propagation layer, layer-aggregation mechanism, message dropout, and node dropout) affect NGCF?
- **RQ3:** How do the representations benefit from the high-order connectivity?

RQ 1 Answer

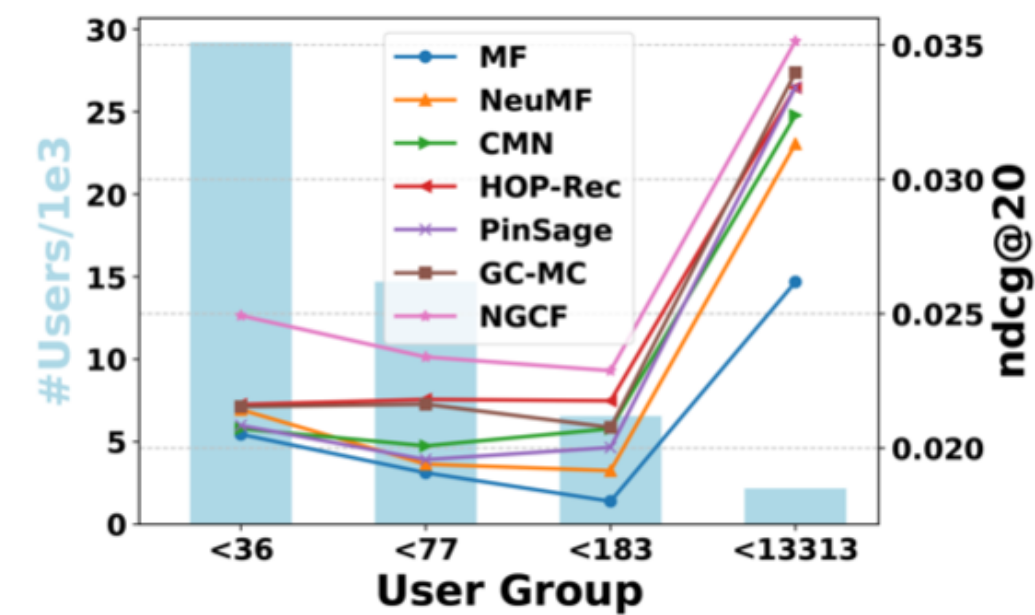
Sparsity 문제는 보통 추천 시스템의 표현성expressiveness를 제약한다. 비활동적인 사용자의 적은 상호작용은 고품질의 표현을 만들기에 불충분하다. 우리는 다양한 sparsity levels의 사용자 그룹을 나누어 실험을 수행한다. 한 명의 사용자 당 interaction 개수를 기초로 하여, 4 개의 그룹으로 테스트 셋을 나누었다. 이들 각각은 동일한 total interaction을 갖는다.



(a) ndcg on Gowalla



(b) ndcg on Yelp2018*



(c) ndcg on Amazon-book

NGCF는 모든 유저 그룹에서 다른 경쟁 모델들보다 좋은 성능을 보인다. 이것은 고차 연결성을 이용하는 것이 협업 신호를 효과적으로 포착함으로써 비활동적인 유저의 표현 학습을 매우 용이하게 해준다는 것을 보여준다.

2. NGCF Neural Graph Collaborative Filtering (2019)

6. Experiments

- **RQ1:** How does NGCF perform as compared with state-of-the-art CF methods?
- **RQ2:** How do different hyper-parameter settings (e.g., depth of layer, embedding propagation layer, layer-aggregation mechanism, message dropout, and node dropout) affect NGCF?
- **RQ3:** How do the representations benefit from the high-order connectivity?

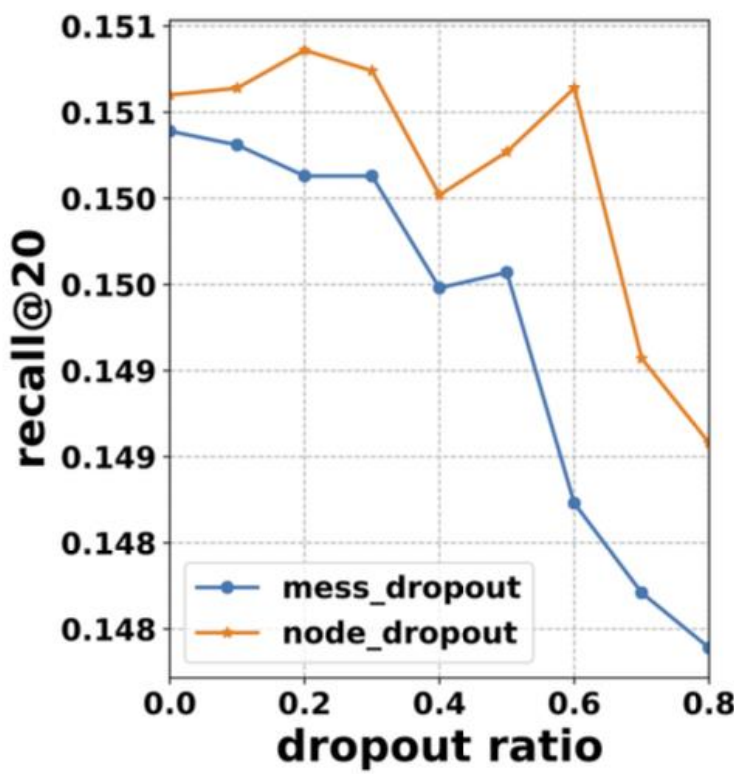
RQ 2 Answer

Table 3: Effect of embedding propagation layer numbers (L).

	Gowalla		Yelp2018*		Amazon-Book	
	recall	ndcg	recall	ndcg	recall	ndcg
NGCF-1	0.1556	0.1315	0.0543	0.0442	0.0313	0.0241
NGCF-2	0.1547	0.1307	0.0566	0.0465	0.0330	0.0254
NGCF-3	0.1569	0.1327	0.0579	0.0477	0.0337	0.0261
NGCF-4	0.1570	0.1327	0.0566	0.0461	0.0344	0.0263

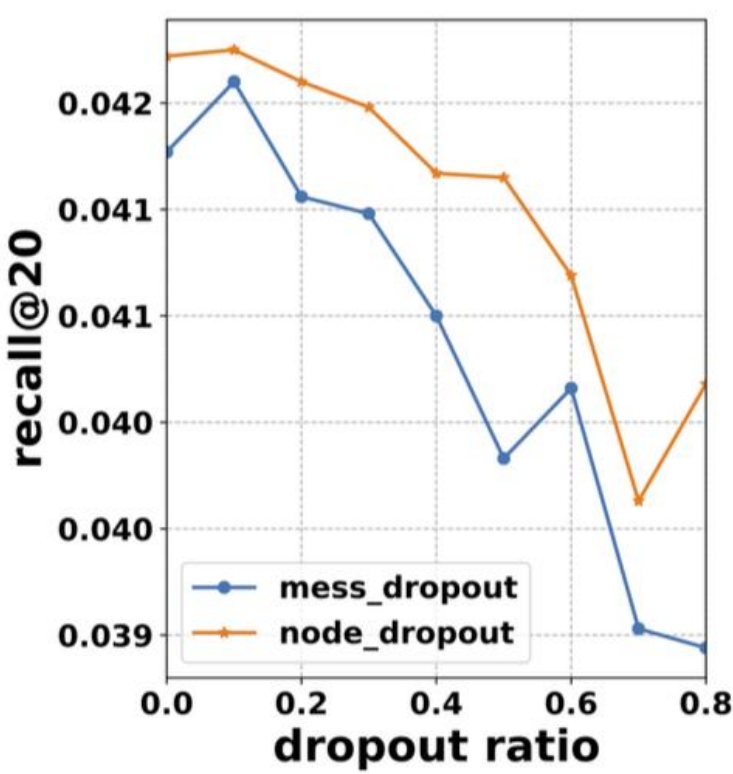
대체로, deeper한 경우가 성능이 더 좋은 경향을 보인다.

RQ 2 Answer

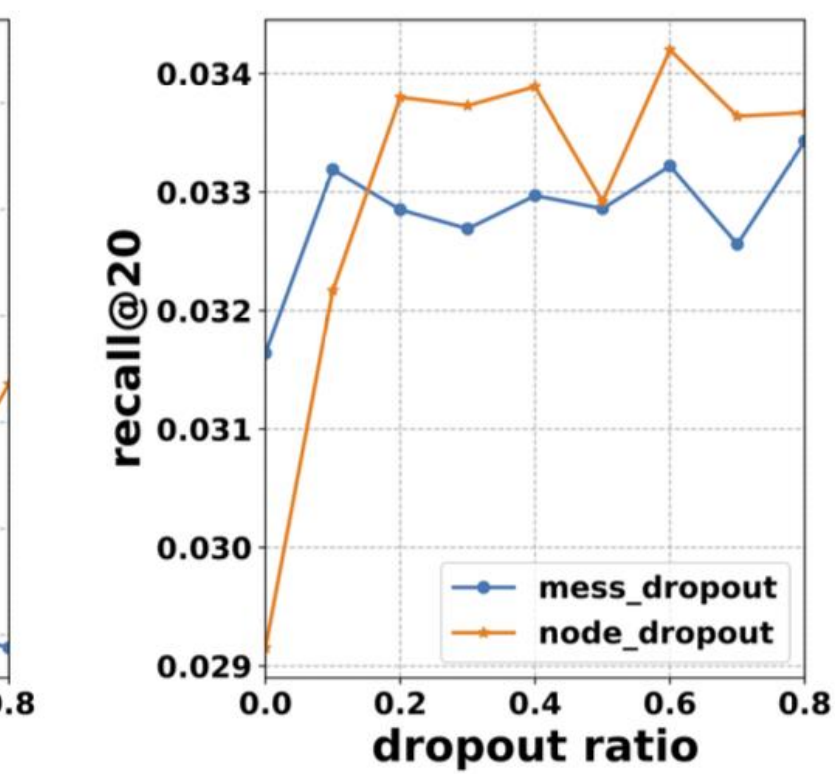


(a) Gowalla

Dropout rate가 낮을수록 좋은 경향이 있다.



(b) Yelp2018*



(c) Amazon-Book

rate가 0.1 이상부터는 대체로 비슷한 성능

>> 흥미로운 발견은 node dropout이 message dropout 보다 일관적으로 더 좋은 성능을 제공해준다는 것이다!

2. NGCF Neural Graph Collaborative Filtering (2019)

6. Experiments

- **RQ1:** How does NGCF perform as compared with state-of-the-art CF methods?
- **RQ2:** How do different hyper-parameter settings (*e.g.*, depth of layer, embedding propagation layer, layer-aggregation mechanism, message dropout, and node dropout) affect NGCF?
- **RQ3:** How do the representations benefit from the high-order connectivity?

RQ 3 Answer

어떻게 Embedding Propagation Layer가 Embedding Space의 표현 학습을 용이하게 해주는 방법에 대해 알아보자.

[EXAMPLE]

- Gowalla dataset에서 임의적으로 6 명의 사용자를 선택하고, 그들과 관련된 item도 선택해보자.
- 우리는 어떻게 그 표현이 NGCF의 깊이에 영향을 받는가에 대해 관찰한다.

주요 2 가지 관찰 포인트

(1) u, i 의 연결들은 embedding space에 잘 반영된다. NGCF-3의 표현은 식별할 수 있는 clustering을 보여준다. 이것은 동일한 색을 가진 점들이 clusters를 형성하는 경향이 있음을 의미한다.

(2) 두 그림 (a),(b)에서 동일한 유저 12201(yellow)를 동시에 분석해보면, 우리는 NGCF-3에서 그/그녀의 historical items의 embedding이 더 가까이에 있는 경향이 있음을 볼 수 있다. 이것은 NGCF-3가 <명시적인 협업 신호>를 <그것의 표현>에 주입할 수 있음을 보여준다.

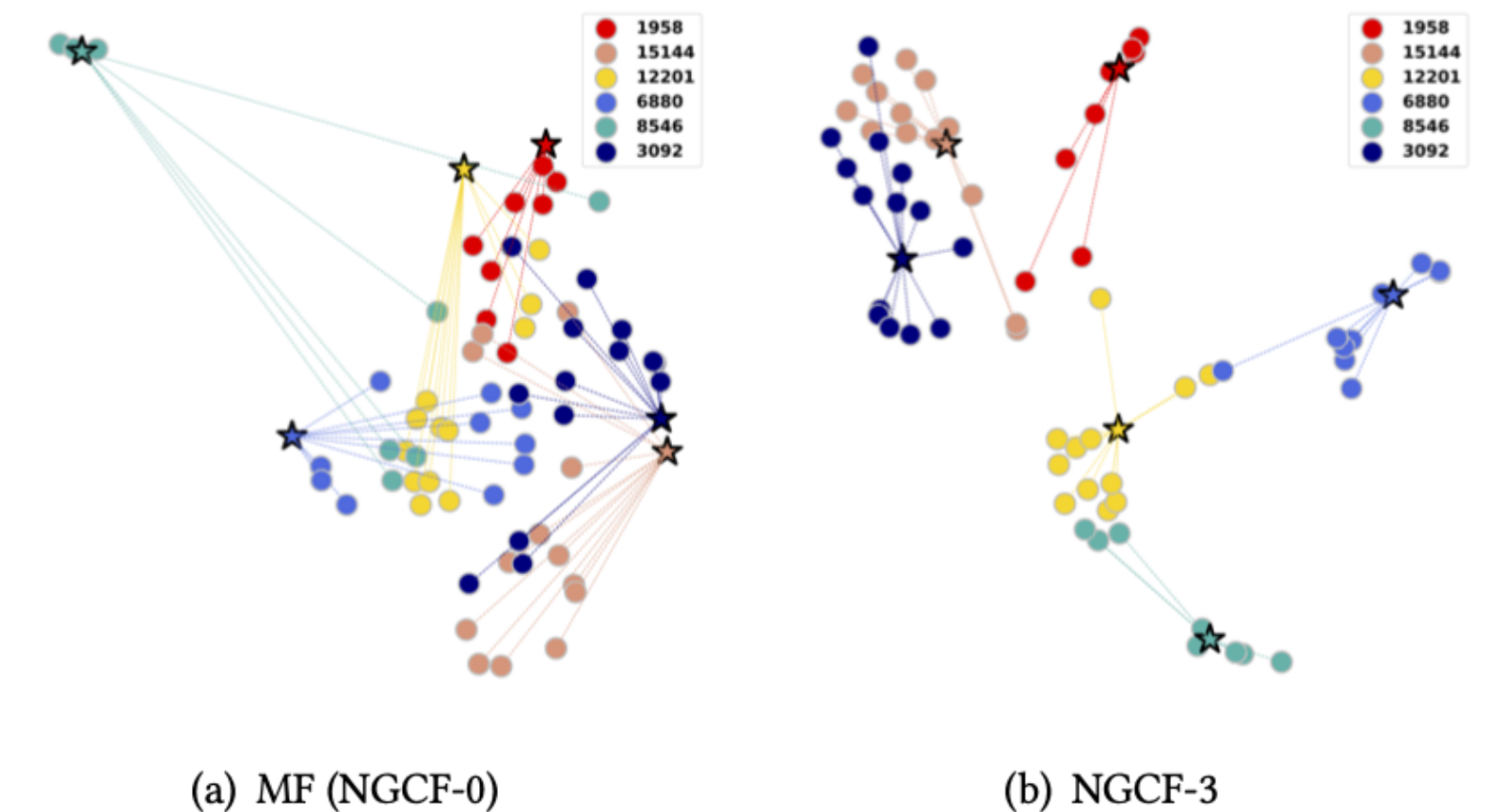


Figure 7: Visualization of the learned t-SNE transformed representations derived from MF and NGCF-3. Each star represents a user from Gowalla dataset, while the points with the same color denote the relevant items. Best view in color.

References

[1] Wang, Xiang, et al. "Neural graph collaborative filtering." Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval. 2019.

[2] <https://medium.com/@meuleman.mathias/reproducing-neural-graph-collaborative-filtering-a8982c7d3df6>

[3] https://github.com/SytzeAndr/NGCF_RP32/blob/hand-in/NGCF.ipynb

[4] <https://ride-or-die.info/normalized-discounted-cumulative-gain/>

[5] <https://j1w2k3.tistory.com/585>

[6] <https://pong dangstory.tistory.com/511>