

15기 추천시스템 세미나

Tobig's 14기 & 15기 강의자

박지은 & 이성범

딥러닝 기반 추천시스템

Unit 01 | 추천시스템 복습

Unit 02 | MLP-based Matrix Factorization

Unit 03 | RNN-based Model

Unit 04 | Autoencoder-based Model

Unit 01 | 추천시스템 복습

추천시스템이란?

- 사용자(user)에게 상품(item)을 제안하는 소프트웨어 도구이자 기술
- 어떤 사용자에게 어떤 상품을 어떻게 추천할까?

items



제안된 상품

query



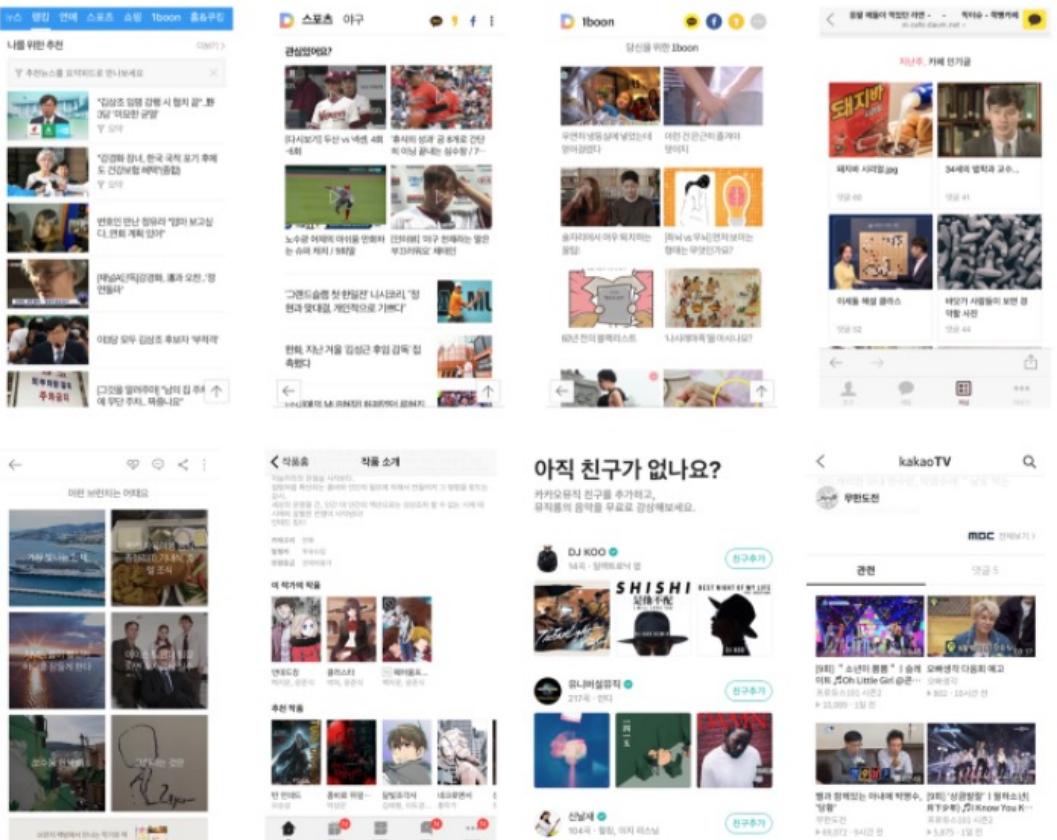
사용자의 정보 + 추가적 맥락

Unit 01 | 추천시스템 복습

추천시스템의 활용

토로스 (TOROS)

- 카카오의 자체 개발 인공지능 플랫폼
- 다음뉴스, 브런치, 다음웹툰, 카카오뮤직, ...



Unit 01 | 추천시스템 복습

추천시스템의 역사

Apriori 알고리즘
• 연관상품추천

2005 ~ 2010

2010 ~ 2015

2013 ~ 2017

2015 ~ 2017

2017 ~

Spark를 이용한 빅데이터
• FP-Growth
• Matrix Factorization

개인화 추천시스템
• Factorization Machine
• Hierarchical RNN
• 강화학습 + Re-Ranking
• 딥러닝

협업 필터링
• SVD
• Netflix 추천 대회

딥러닝을 이용한 추천시스템
• Collaborative Filtering + 딥러닝,
• Item2Vec, Doc2Vec
• YouTube Recommendation
• Wide & Deep Model

Unit 01 | 추천시스템 복습

Rule-based Model

연관분석 (Association Analysis)

- 아이템 간의 연관성을 찾아내는 알고리즘
- 연관이란?
 - ① 얼마나 자주 같이 구매가 되는가?
 - ② A 아이템을 구매하는 사람이 B 아이템을 구매하는가? (장바구니 분석)

Unit 01 | 추천시스템 복습

Rule-based Model

연관분석 (Association Analysis)

- support (지지도)

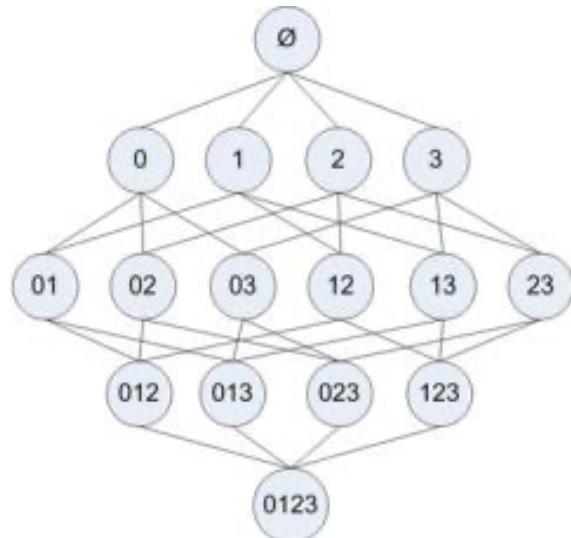
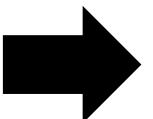
For the rule $A \rightarrow B$, $\text{support}(A) = P(A)$

- confidence (신뢰도)

$$\text{lift}(A \rightarrow B) = \frac{P(A, B)}{P(A) \cdot P(B)}$$

- lift (향상도)

$$\text{confidence}(A \rightarrow B) = \frac{P(A, B)}{P(A)}$$



지지도, 신뢰도, 향상도 높은 규칙 찾기 위해
가능한 모든 경우의 수를 탐색

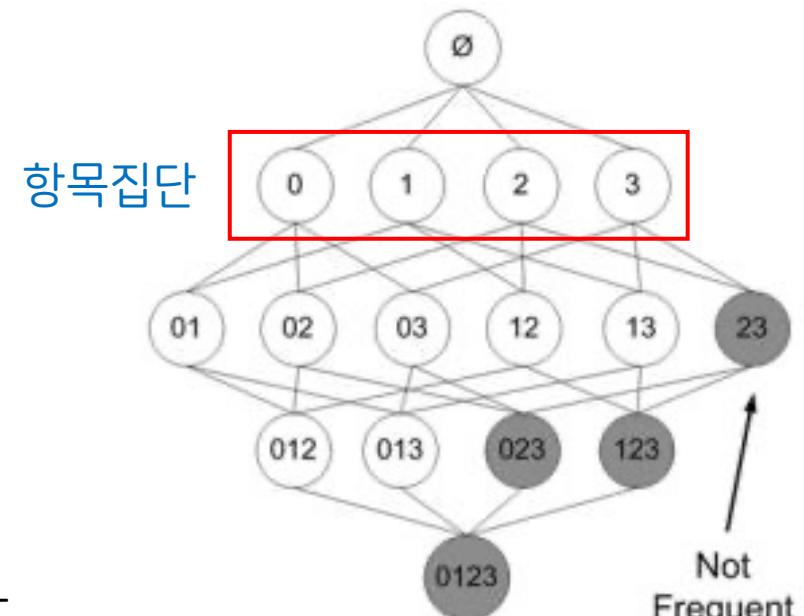
Unit 01 | 추천시스템 복습

Rule-based Model

Apriori 알고리즘

- 아이템 셋의 증가를 줄이기 위한 방법
- 빈번하지 않은 아이템 셋은 하위 아이템 셋 미리 제거

{2, 3}의 지지도 > {0, 2, 3}, {1, 2, 3}의 지지도



Unit 01 | 추천시스템 복습

Rule-based Model

Apriori 알고리즘

① k개의 아이템으로 단일항목집단 생성

우유, 양상추, 기저귀, 맥주, 주스

② 단일항목집단에서 최소 지지도 이상의 항목만 선택 최소 지지도 = 0.5

$P(\text{우유}) = 0.5$, $P(\text{양상추}) = 0.75$, $P(\text{기저귀}) = 0.75$, $\cancel{P(\text{주스}) = 0.25}$, $P(\text{맥주}) = 0.75$

③ 선택된 항목만을 대상으로 2개항목집단 생성

{우유, 양상추}, {우유, 기저귀}, {우유, 맥주}, {양상추, 기저귀}, {양상추, 맥주}, {기저귀, 맥주}

④ 2개항목집단에서 최소 지지도 혹은 신뢰도 이상의 항목만 선택

{우유, 양상추}: 0.25, {우유, 기저귀}: 0.5, {우유, 맥주}: 0.25, {양상추, 기저귀}: 0.5, {양상추, 맥주}: 0.75, {기저귀, 맥주}: 0.5

⑤ k개의 k-item frequent set을 생성할 때까지 반복

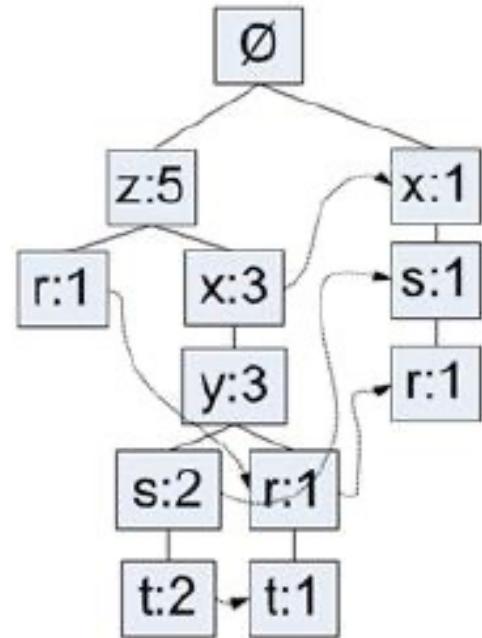
ID	우유	양상추	기저귀	주스	맥주
0	1	0	1	1	0
1	0	1	1	0	1
2	1	1	1	0	1
3	0	1	0	0	1

Unit 01 | 추천시스템 복습

Rule-based Model

FP-Growth 알고리즘

- FP Tree라는 구조를 사용하여 속도 측면의 단점 개선
- 동일하게 발생하는 아이템 셋을 찾는 데에 유리
- 아이템 간의 연관성을 찾는 것은 어려움



FP-Tree

Unit 01 | 추천시스템 복습

Rule-based Model

FP-Growth 알고리즘

- ① 각 아이템의 지지도 계산, 최소 지지도 이상의 아이템만 선택
- ② 모든 거래에서 빈도가 높은 아이템 순서대로 정렬

ID	우유	양상추	기저귀	주스	맥주
0	1	0	1	1	0
1	0	1	1	0	1
2	1	1	1	0	1
3	0	1	0	0	1

ID	아이템
0	우유, 기저귀, 주스
1	양상추, 기저귀, 맥주
2	우유, 양상추, 기저귀, 맥주
3	양상추, 맥주



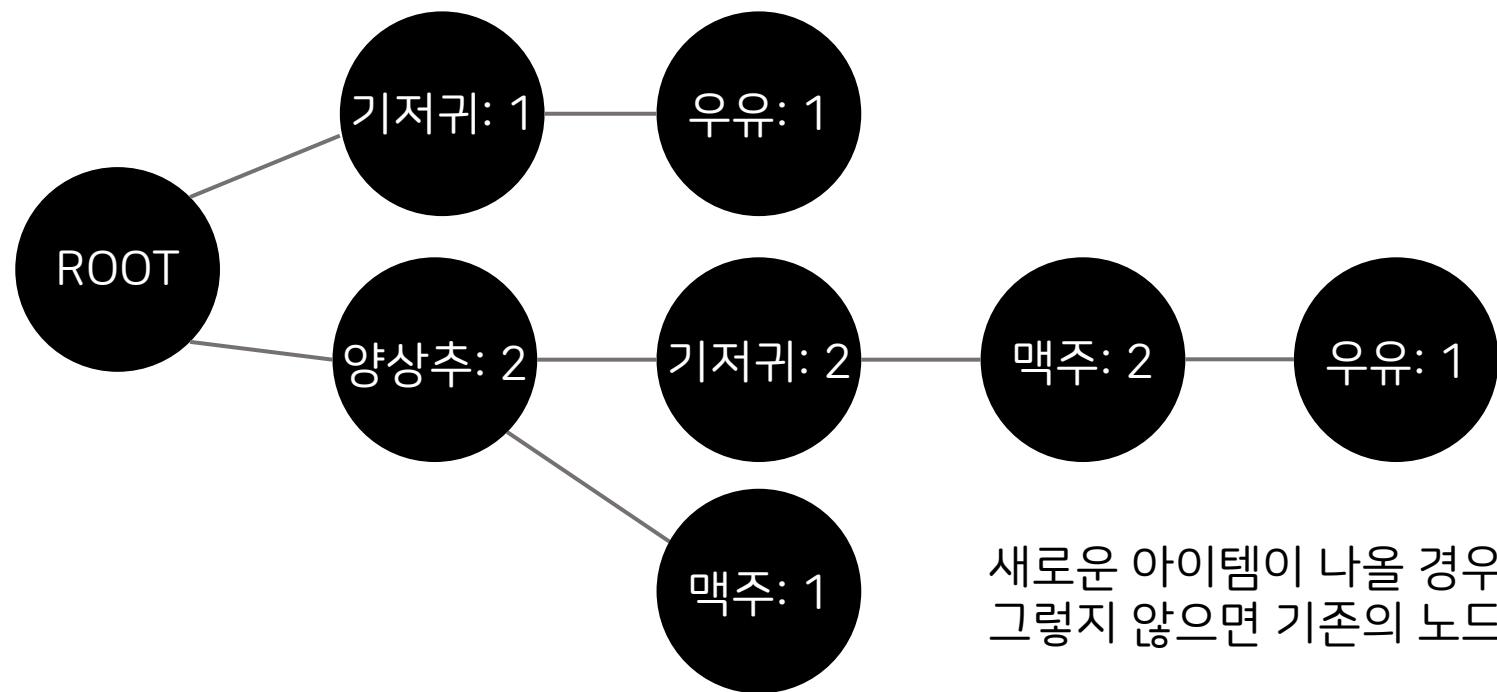
ID	아이템
0	기저귀, 우유
1	양상추, 기저귀, 맥주
2	양상추, 기저귀, 맥주, 우유
3	양상추, 맥주

Unit 01 | 추천시스템 복습

Rule-based Model

FP-Growth 알고리즘

- ③ 부모노드를 중심으로 자식노드로 거래 추가



새로운 아이템이 나올 경우에는 부모노드부터 시작,
그렇지 않으면 기존의 노드에서 확장

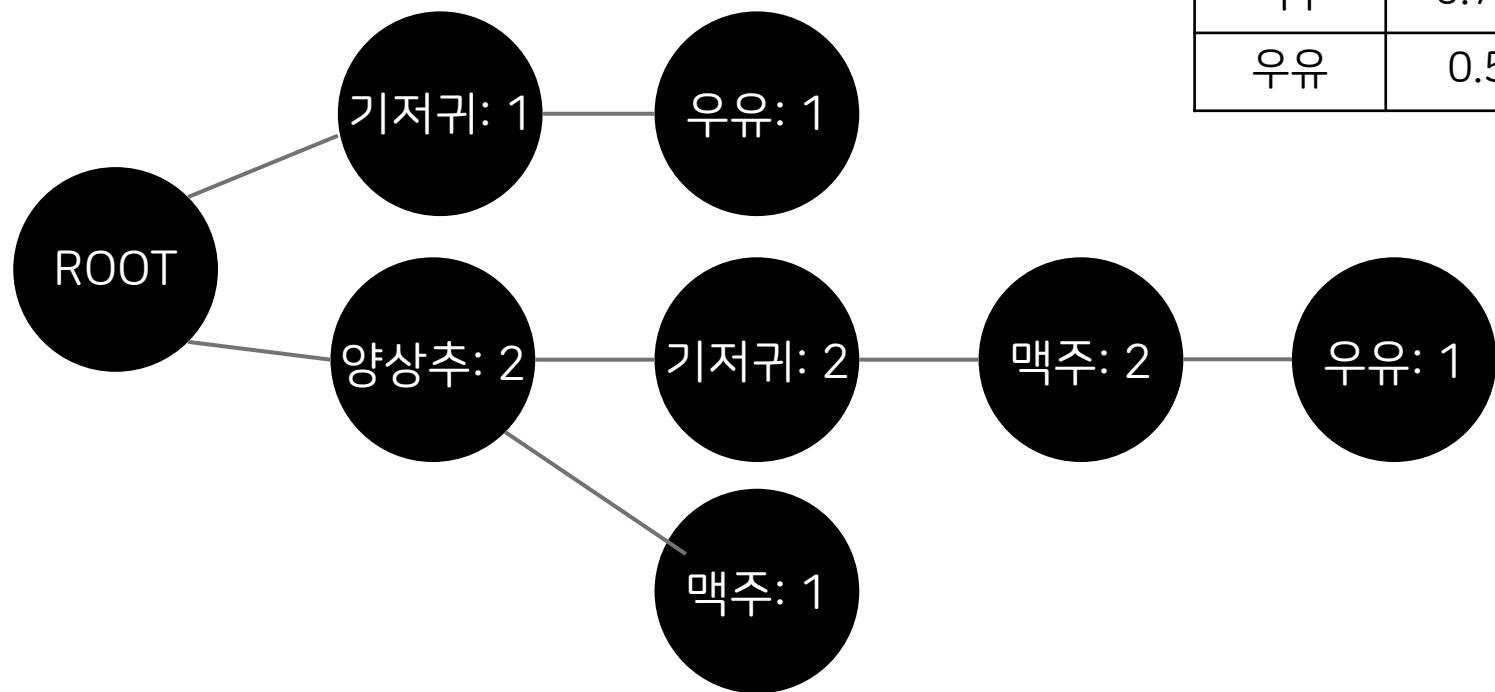
ID	아이템
0	기저귀, 우유
1	양상추, 기저귀, 맥주
2	양상추, 기저귀, 맥주, 우유
3	양상추, 맥주

Unit 01 | 추천시스템 복습

Rule-based Model

FP-Growth 알고리즘

- ④ 지지도가 낮은 순서부터 조건부 패턴 생성



아이템	지지도	Conditional Pattern bases
기저귀	0.75	{양상추}: 2
양상추	0.75	{}
맥주	0.75	{양상추, 기저귀}: 2, {양상추}: 1
우유	0.5	{양상추, 기저귀, 맥주}: 2, {기저귀}: 1

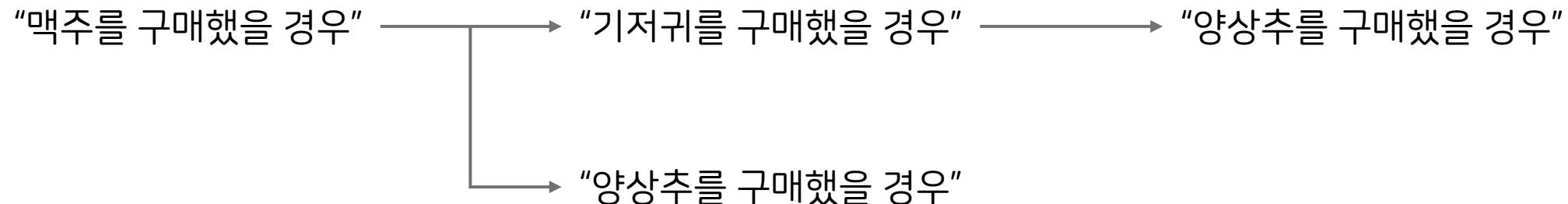
Unit 01 | 추천시스템 복습

Rule-based Model

FP-Growth 알고리즘

⑤ Conditional Pattern bases를 기반으로 패턴 생성

아이템	지지도	Conditional Pattern bases
기저귀	0.75	{양상추}: 2
양상추	0.75	{}
맥주	0.75	{양상추, 기저귀}: 2, {양상추}: 1
우유	0.5	{양상추, 기저귀, 맥주}: 2, {기저귀}: 1



Unit 01 | 추천시스템 복습

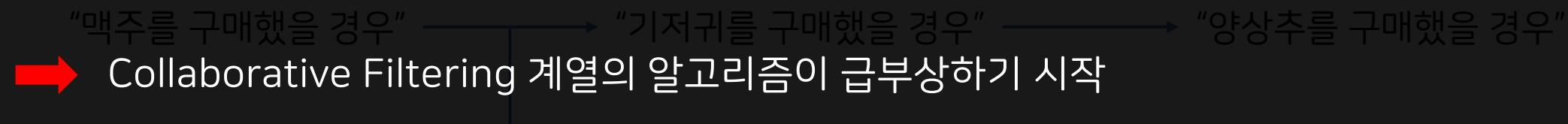
Rule-based Model

연관 분석 알고리즘의 한계

⑤ Conditional Pattern bases를 기반으로 패턴 생성

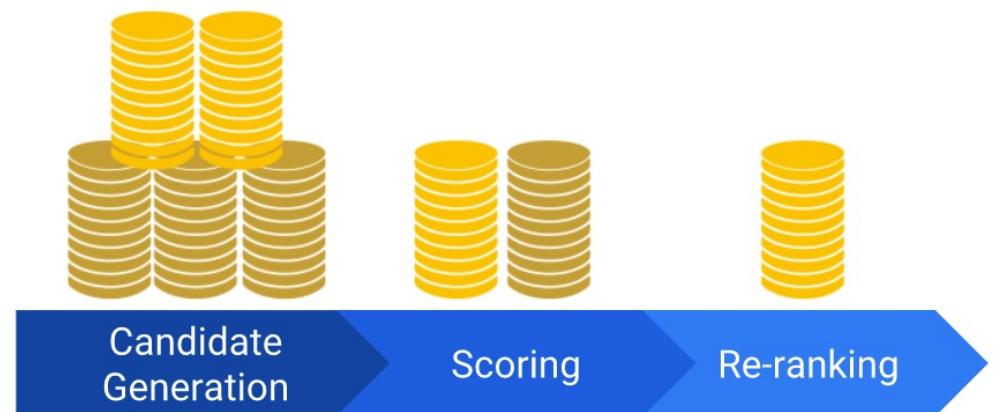
- 최소 지지도 선정이 어려움
- 인과를 알기 어려움 (Causal Inference)
- 메모리와 시간을 많이 소비하는 알고리즘

아이템	지지도	Conditional Pattern bases
기저귀	0.75	{양상추}: 2
양상추	0.75	{}
맥주	0.75	{양상추, 기저귀}: 2, {양상추}: 1
우유	0.5	{양상추, 기저귀, 맥주}: 2, {기저귀}: 1



Unit 01 | 추천시스템 복습

추천시스템의 Overview



① Candidate Generation

거대한 말뭉치에서 유저 정보를 토대로
유사도가 높은 후보 하위 집합 생성

② Scoring

사용자에게 표시할 항목을 선택하기 위해
후보에 점수와 순위 매김

③ Re-ranking

추가 feature를 더하여 성능 개선 후 리스트 생성

Unit 01 | 추천시스템 복습

Candidate Generation Overview

- 쿼리가 주어지면 연관된 후보를 생성

Content-based Filtering	아이템 간의 유사도를 이용하여 유저가 좋아할 만한 아이템 추천 ex) 유저 A가 고양이 영상을 여러 개 시청하면, 다른 동물 영상을 추천
-------------------------	---

Collaborative Filtering	쿼리와 아이템 간의 유사도를 동시에 고려하여 추천 ex) 유저 A와 유저 B가 비슷한 취향을 갖고 있을 때, 유저 B가 좋아하는 아이템을 유저 A에게도 추천
-------------------------	--

Unit 01 | 추천시스템 복습

Candidate Generation Overview

- 임베딩 공간

각각의 아이템과 쿼리를 공통 임베딩 공간 $E = \mathbb{R}^d$ 로 매핑

일반적으로 임베딩 공간은 저차원 (corpus 크기 $> d$)

아이템이나 쿼리 셋의 latent structure를 포착

ex) 같은 유저가 시청한 비슷한 유튜브 영상들은 임베딩 공간에서 서로 **가깝게** 매핑



유사도

Unit 01 | 추천시스템 복습

Candidate Generation Overview

- 유사도 측정

$$s : E \times E \rightarrow \mathbb{R}$$

임베딩 쌍 임베딩 공간

쿼리 임베딩 $q \in E$  아이템 임베딩 $x \in E$

$$s(q, x)$$

높은 유사도의 임베딩 반환

Unit 01 | 추천시스템 복습

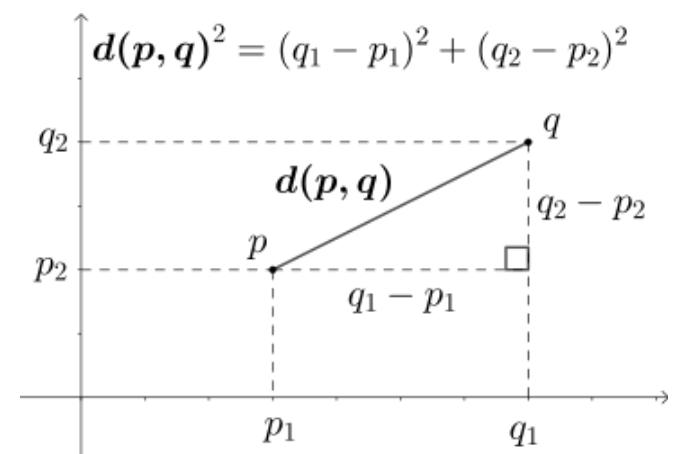
유사도 함수

유클리디안 유사도

- 두 점의 직선거리를 구하여 유사도 계산
- 계산이 쉬움
- p 와 q 의 분포가 다르거나 범위가 다른 경우 상관성을 놓침

$$\text{유클리디안 유사도} = \frac{1}{\text{유클리디안 거리} + 1e - 5}$$

$$\|p - q\| = \sqrt{(p - q) \cdot (p - q)} = \sqrt{\|p\|^2 + \|q\|^2 - 2pq}$$



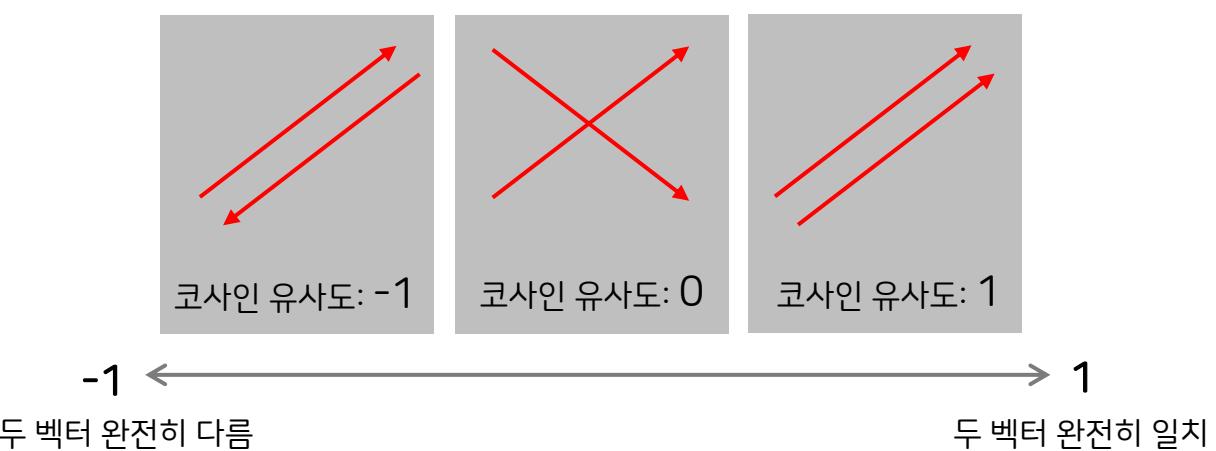
Unit 01 | 추천시스템 복습

유사도 함수

코사인 유사도

- 두 벡터 간의 사이각을 구하여 유사도 계산
- 벡터의 크기가 중요하지 않은 경우에 거리를 측정하기 위한 지표로 사용
- 벡터의 크기가 중요한 경우에 대해서 잘 작동하지 않음

$$\cos(\vec{q} \cdot \vec{d}) = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}| |\vec{d}|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2} \sqrt{\sum_{i=1}^{|V|} d_i^2}}$$



Unit 01 | 추천시스템 복습

유사도 함수

피어슨 유사도

- 두 변수 간의 선형관계 파악
- 코시-슈바르츠 부등식에 의해 1과 -1 사이의 값을 가짐
- 1은 완벽한 양의 선형상관관계, -1은 완벽한 음의 선형상관관계 의미

$$r_{XY} = \frac{\text{공분산}}{\text{표준편차} \cdot \text{표준편차}} = \frac{\sum_i^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_i^n (X_i - \bar{X})^2} \sqrt{\sum_i^n (Y_i - \bar{Y})^2}}$$

X와 Y가 함께 변하는 정도

X와 Y가 따로 변하는 정도

Unit 01 | 추천시스템 복습

유사도 함수

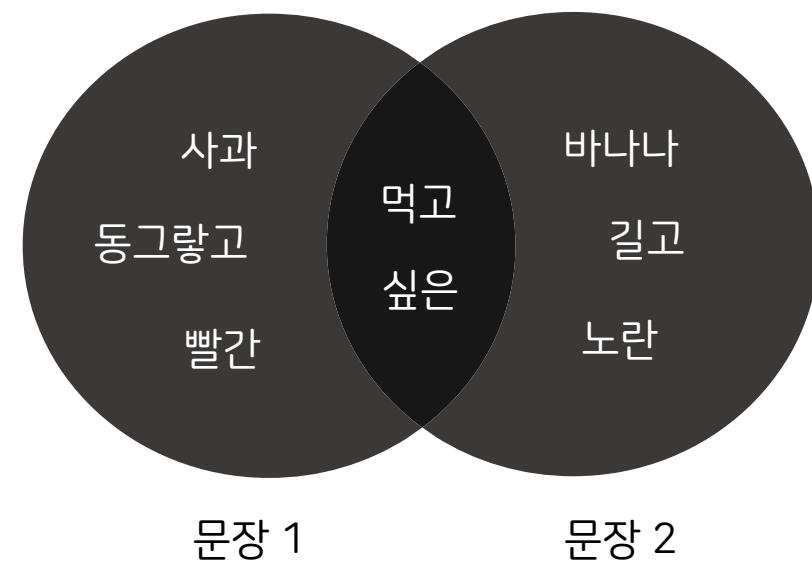
자카드 유사도

- 두 문장을 단어의 집합으로 만든 뒤, 전체 단어를 바탕으로 교집합 크기를 합집의 크기로 나눔
- 두 문장 사이의 단어가 많이 겹칠수록 유사도 높음

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

문장 1: 먹고 싶은 동그랗고 빨간 사과

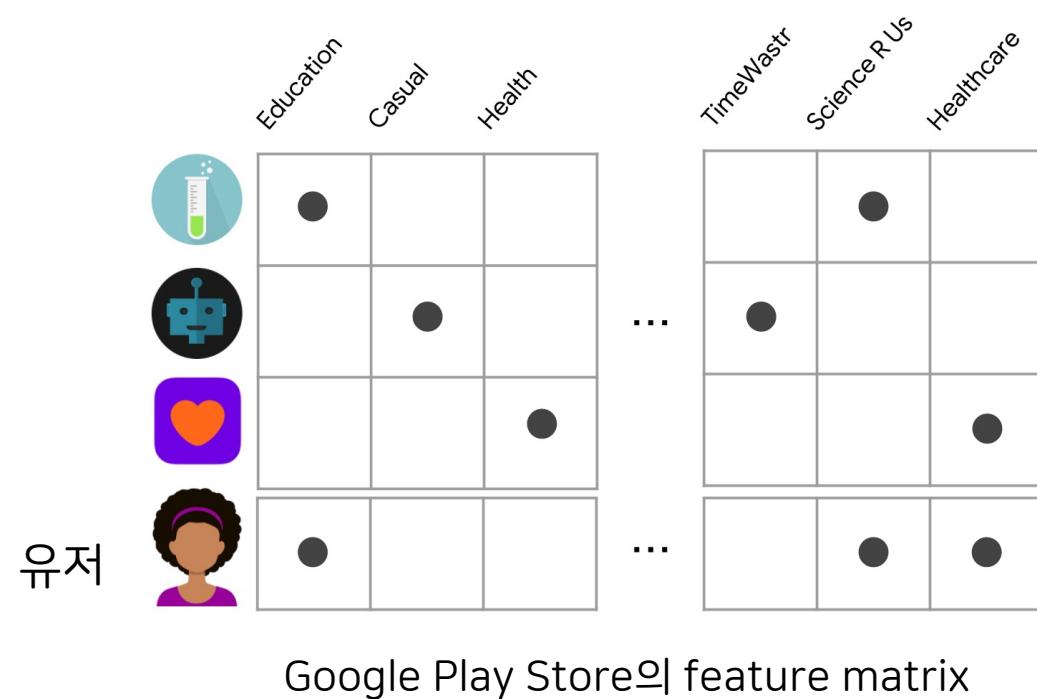
문장 2: 먹고 싶은 길고 노란 바나나



Unit 01 | 추천시스템 복습

Content-based Model

- 유저의 이전 행동이나 explicit feedback을 기반
- 아이템 feature를 이용하여 유저가 좋아하는 아이템과 비슷한 아이템 추천



Unit 01 | 추천시스템 복습

Content-based Model

TF-IDF

- 통계 기반 - 빈도수
 - TF (단어 빈도): 특정 문서 내에서 특정 단어가 얼마나 자주 등장하는지
 - DF (역문서 빈도): 전체 문서에서 특정 단어가 얼마나 자주 등장하는지
- } 문서 내 단어의 가중치 계산

그러나 대규모 corpus를 다룰 때 메모리 상의 문제가 발생하고, 한 번에 학습 데이터 전체를 진행하기 때문에 큰 작업을 처리하기 어려우며, 학습을 통해서 개선하기가 어려움!

Unit 01 | 추천시스템 복습

Content-based Model

Word2Vec (CBOW, Skip-gram)

- 추론 기반 - 주변 단어가 주어졌을 때 중심 단어에 무슨 단어가 들어가는지 추측
- CBOW: 주변 단어를 통해 중심 단어를 채우는 방법
- Skip-gram: 중심 단어를 통해 주변 단어를 채우는 방법



더 어려운 태스크 수행 가능, 하나의 문장에서 더 많은 데이터가 나옴

Unit 01 | 추천시스템 복습

Content-based Model

장점

- 다른 유저들의 정보가 필요하지 않기 때문에, 많은 수의 유저들을 측정할 수 있음
- 유저의 구체적인 선호도를 포착하여 섬세한 추천 가능

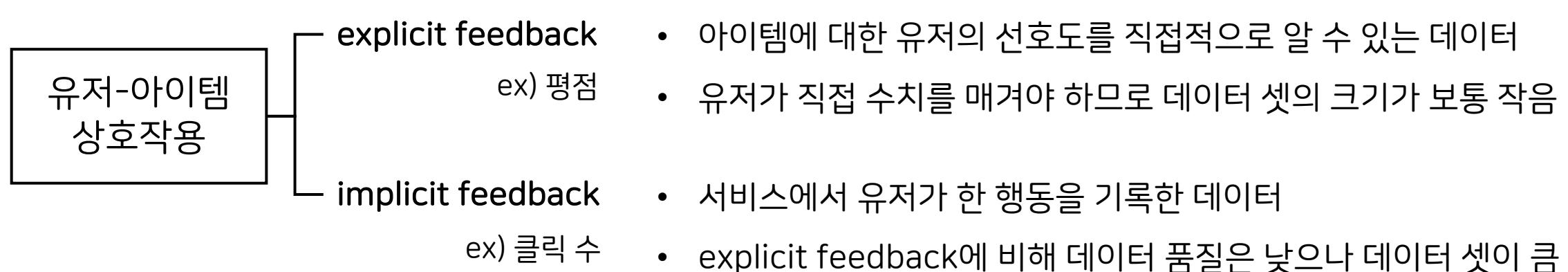
단점

- feature representation이 어느 정도는 hand-engineering 해야 하기 때문에 도메인 지식이 많아야 모델 또한 비슷한 성능을 낼 수 있음
- 이미 존재하는 선호도에 기반해서 추천하기 때문에 이를 확장하는 데 한계가 있음

Unit 01 | 추천시스템 복습

Collaborative Filtering

- 유저-아이템 상호작용을 토대로 한 유사도 기반
- 유저 B의 선호도를 기반으로 유저 A에게 아이템을 추천할 수 있으므로 더 우연적 추천 가능
- hand-engineering에 의존하지 않고 임베딩을 스스로 학습할 수 있음
- KNN, Latent Factor Model, ALS



Unit 01 | 추천시스템 복습

Collaborative Filtering

- user-item matrix

explicit feedback
(rating)



	Argo	Seven	The Usual Suspects
John	5	1	3
Tom	?	?	?
Alice	4	?	3
			5

? 채우는 matrix completion

implicit feedback
(interaction 유무)



	Argo	Seven	The Usual Suspects
John	1	1	1
Tom	0	0	0
Alice	1	0	1
			1

주어진 matrix에서 유저의 선호 패턴 학습

Unit 01 | 추천시스템 복습

Collaborative Filtering

장점

- 도메인 지식 불필요
- 유저가 새로운 선호도를 발견하게 함
- 모델 훈련 시 feedback matrix만 있으면 되므로 맥락적 feature 불필요

단점

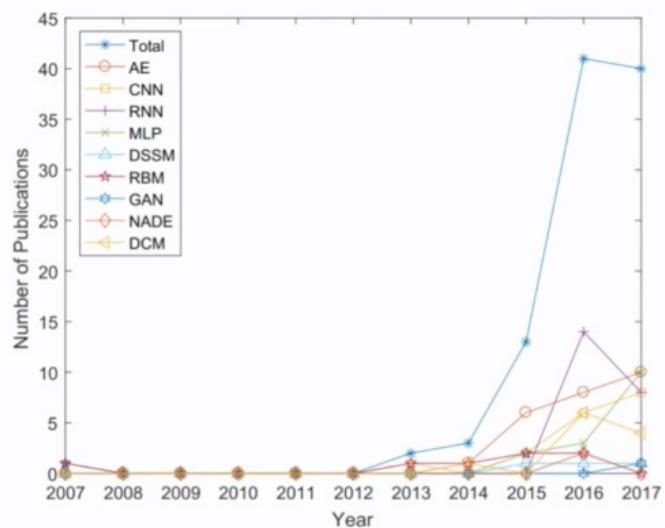
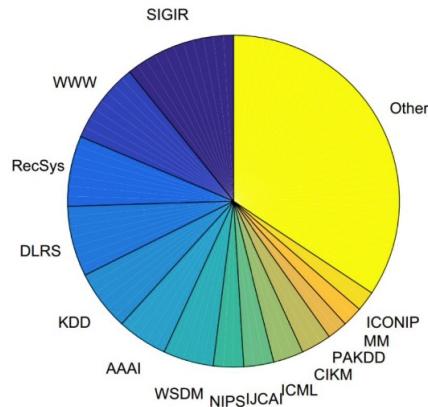
- 학습 할 때 없었던 새로운 아이템에 취약 (cold-start problem)
- 쿼리나 아이템 ID 이외의 side feature를 포함하기 어려움

Unit 01 | 추천시스템 복습

딥러닝 기반 추천시스템

기존에 내적으로 표현하던 유저-아이템 상호작용을 임의의 함수 형식 $f(U, V)$ 로 확장

- 비선형 변환 가능 → 더 복잡한 유저-아이템 상호작용 파악 가능
- representation learning → hand-craft feature design 줄임, 다차원적 정보 처리 가능
- 시퀀스 모델링 → RNN, CNN을 이용하여 성능 향상
- 유연성 → 딥러닝 환경 활용 가능



Unit 01 | 추천시스템 복습

딥러닝 기반 추천시스템

Deep Learning based Recommender System

Neural Network Model

Model using Single DL Technique

MLP AE CNN RNN DSSM RBM NADE GAN

Integration Model

Integrate DL with Traditional RS

Loosely Coupled Tightly Coupled

Deep Composite Model

Recommend Rely Solely on DL

02 MLP-based Matrix Factorization

Neural Collaborative Filtering

Deep Neural Networks for YouTube Recommendation

Unit 02 | MLP-based Matrix Factorization

Multilayer Perceptron based Matrix Factorization

비선형적 유저-아이템 상호작용을 모델링할 수 있음

- 1) Neural Collaborative Filtering
- 2) Deep Neural Networks for YouTube Recommendation

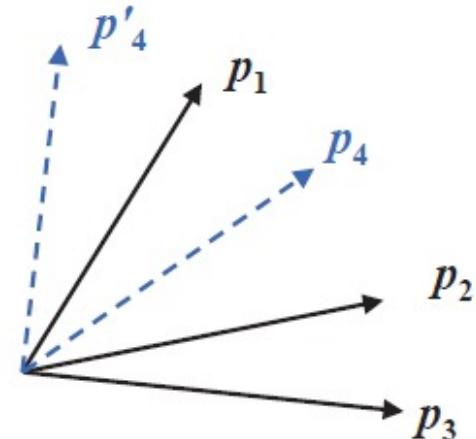
Unit 02 | MLP-based Matrix Factorization

Neural Collaborative Filtering

기존 Matrix Factorization의 문제점

내적 등의 선형 모델은 유저와 아이템 간의 복잡한 관계를 표현하는 데 한계가 있음

	i_1	i_2	i_3	i_4	i_5
u_1	1	1	1	0	1
u_2	0	1	1	0	0
u_3	0	1	1	1	0
u_4	1	0	1	1	1



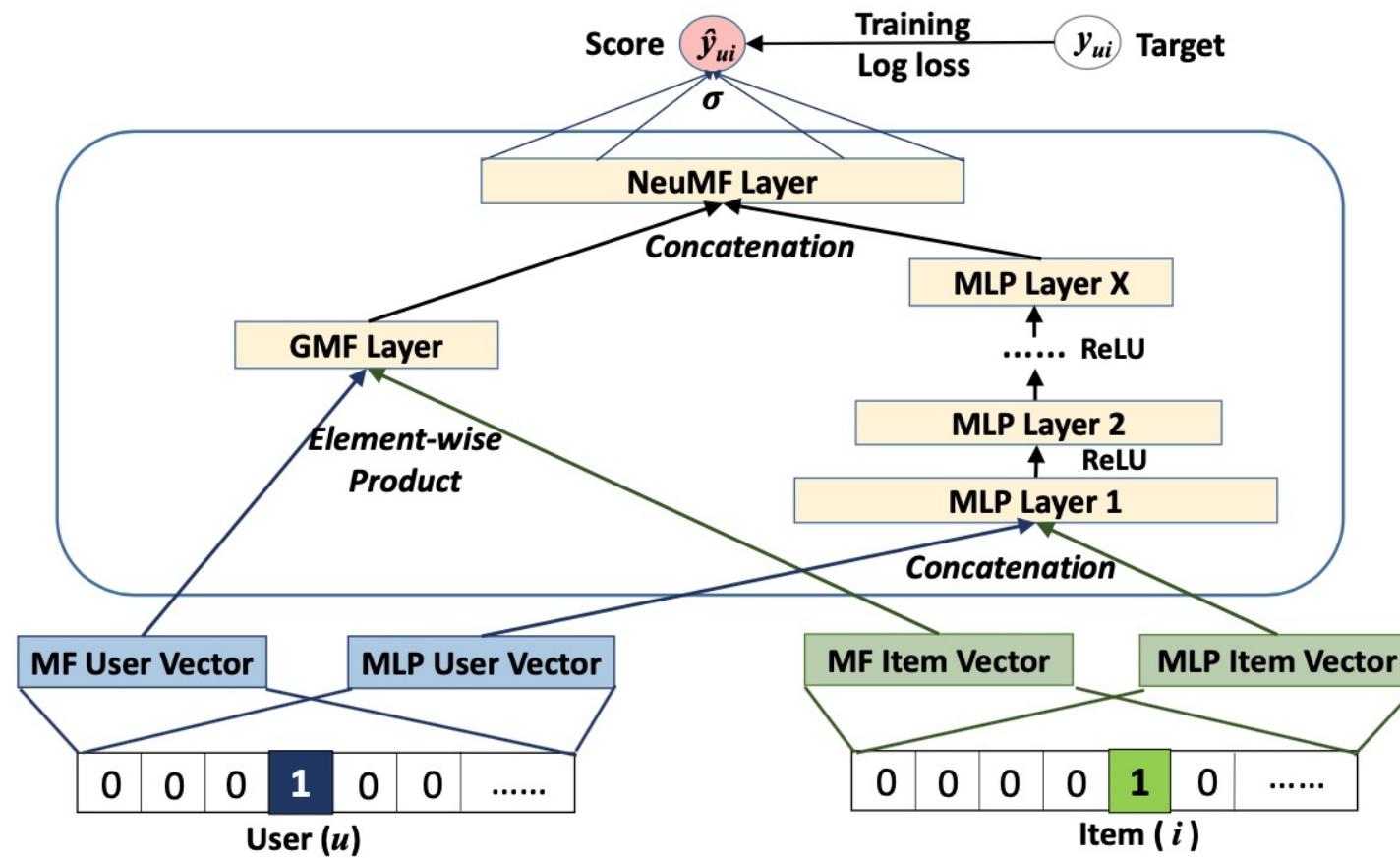
$$\begin{aligned}s_{23}(0.66) &> s_{12}(0.50) > s_{13}(0.40) \\s_{41}(0.60) &> s_{43}(0.40) > s_{42}(0.20)\end{aligned}$$

새로운 유저 4의 관계를 표현할 수 없음

Unit 02 | MLP-based Matrix Factorization

Neural Collaborative Filtering

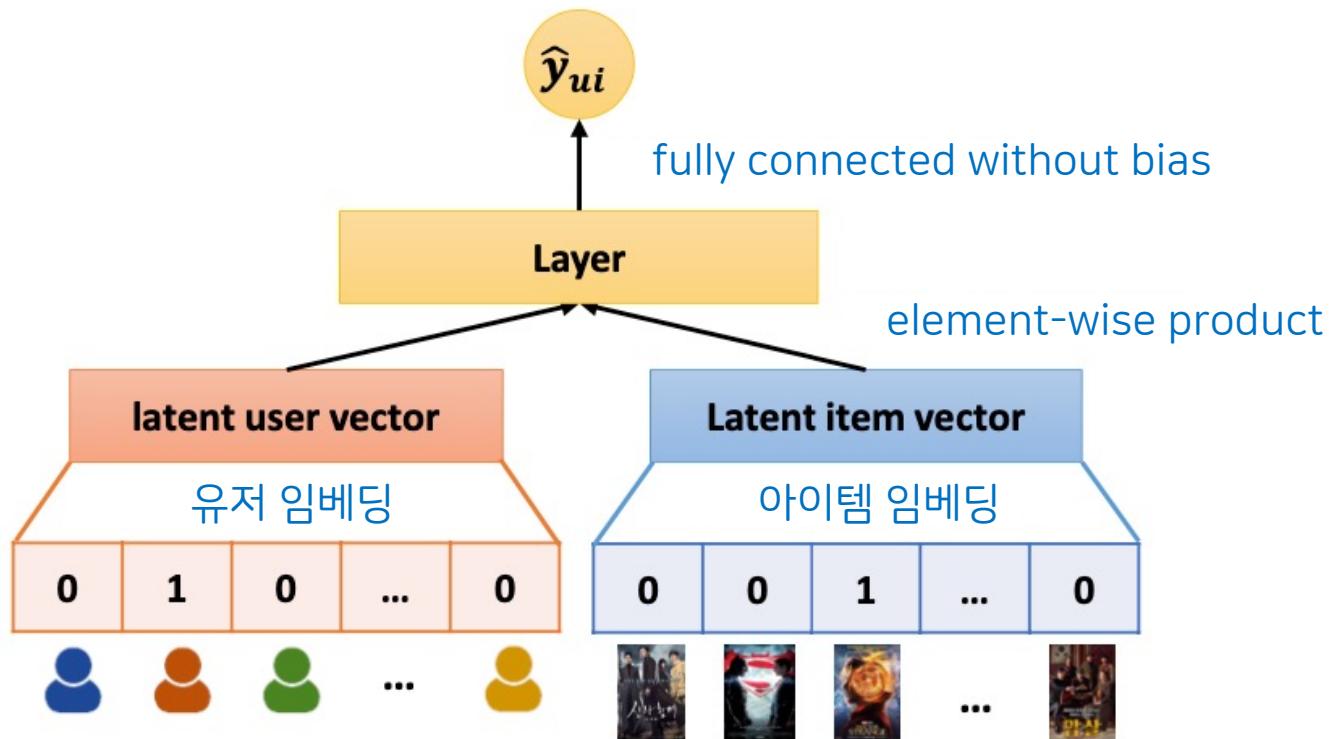
Model Overview



Unit 02 | MLP-based Matrix Factorization

Neural Collaborative Filtering

Generalized Matrix Factorization (GMF)



Unit 02 | MLP-based Matrix Factorization

Neural Collaborative Filtering

Generalized Matrix Factorization (GMF)

① Input Layer

input으로 유저와 아이템의 one-hot vector 넣음



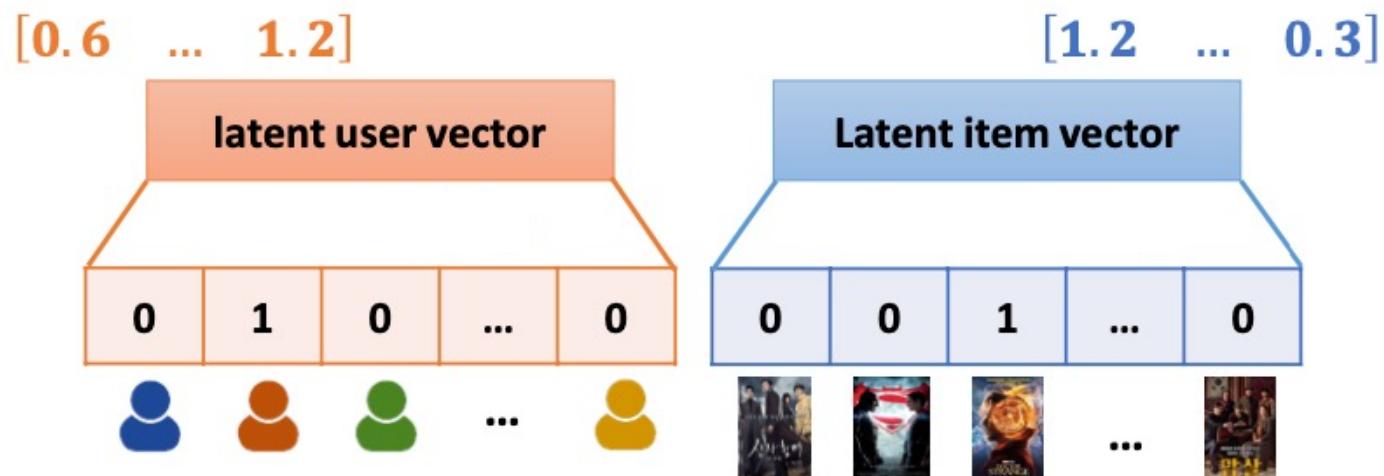
Unit 02 | MLP-based Matrix Factorization

Neural Collaborative Filtering

Generalized Matrix Factorization (GMF)

② Embedding Layer

input 단계의 sparse 벡터를 dense 벡터로 맵핑



Unit 02 | MLP-based Matrix Factorization

Neural Collaborative Filtering

Generalized Matrix Factorization (GMF)

② Embedding Layer

$$\begin{array}{c}
 \text{유저 임베딩} \\
 \left[\begin{matrix} 0 & 1 & 0 & \cdots & 0 \end{matrix} \right] \times
 \end{array}
 \begin{array}{c}
 \text{1}\times m \text{ 벡터} \\
 \left[\begin{matrix} 0.6 & 0.8 & \cdots & 0.6 \\ 0.6 & 0.4 & \cdots & 1.2 \\ 0.2 & 0.5 & \cdots & 0.2 \\ \vdots & & & \vdots \\ 0.7 & 0.5 & \cdots & 0.3 \end{matrix} \right]
 \end{array}
 \begin{array}{c}
 \text{m}\times k \text{ 벡터} \\
 = \left[\begin{matrix} 0.6 & 0.4 & \cdots & 1.2 \end{matrix} \right]
 \end{array}
 \begin{array}{c}
 \text{1}\times k \text{ 벡터} \\
 \text{유저 latent vector} \\
 (U)
 \end{array}$$

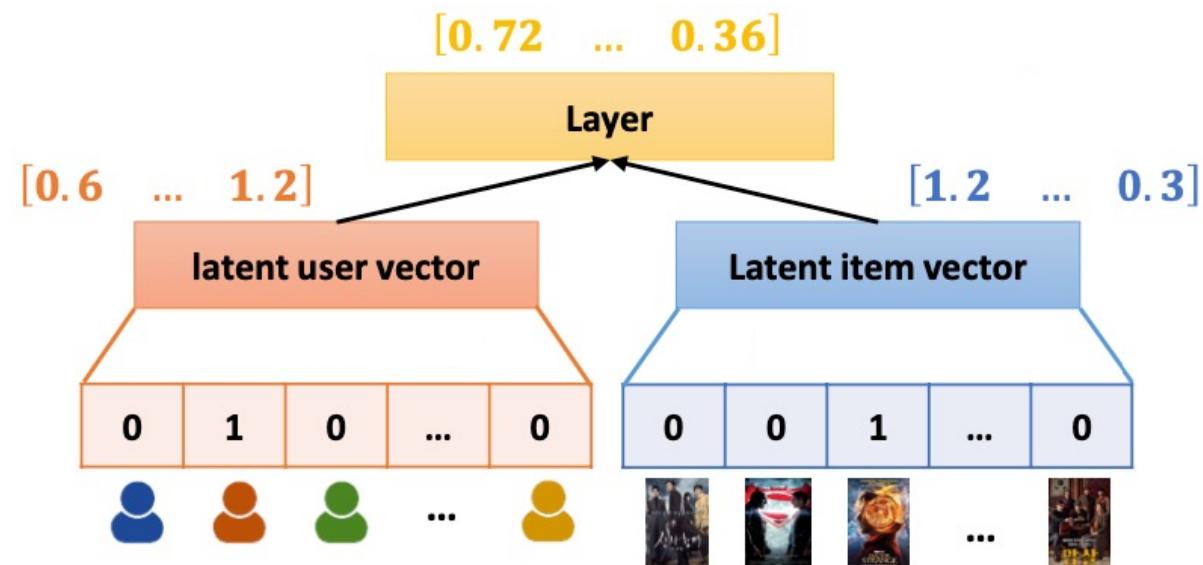
Unit 02 | MLP-based Matrix Factorization

Neural Collaborative Filtering

Generalized Matrix Factorization (GMF)

③ Element-wise Product

k차원의 feature에 대하여 대응하는 값끼리 곱해서 더함



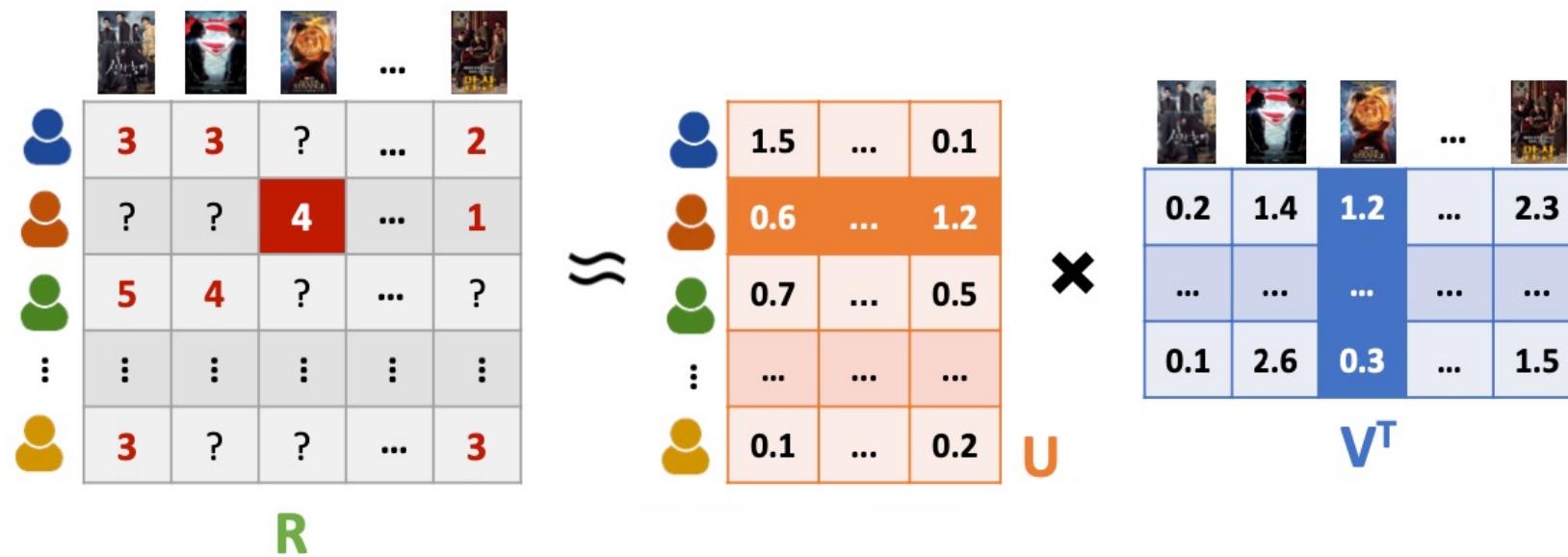
Unit 02 | MLP-based Matrix Factorization

Neural Collaborative Filtering

Generalized Matrix Factorization (GMF)

③ Element-wise Product

그러나 더할 때 각각의 값에 **가중치**를 매겨서 더함 → 원하는 특징에 따라 조정 가능하게 일반화



Unit 02 | MLP-based Matrix Factorization

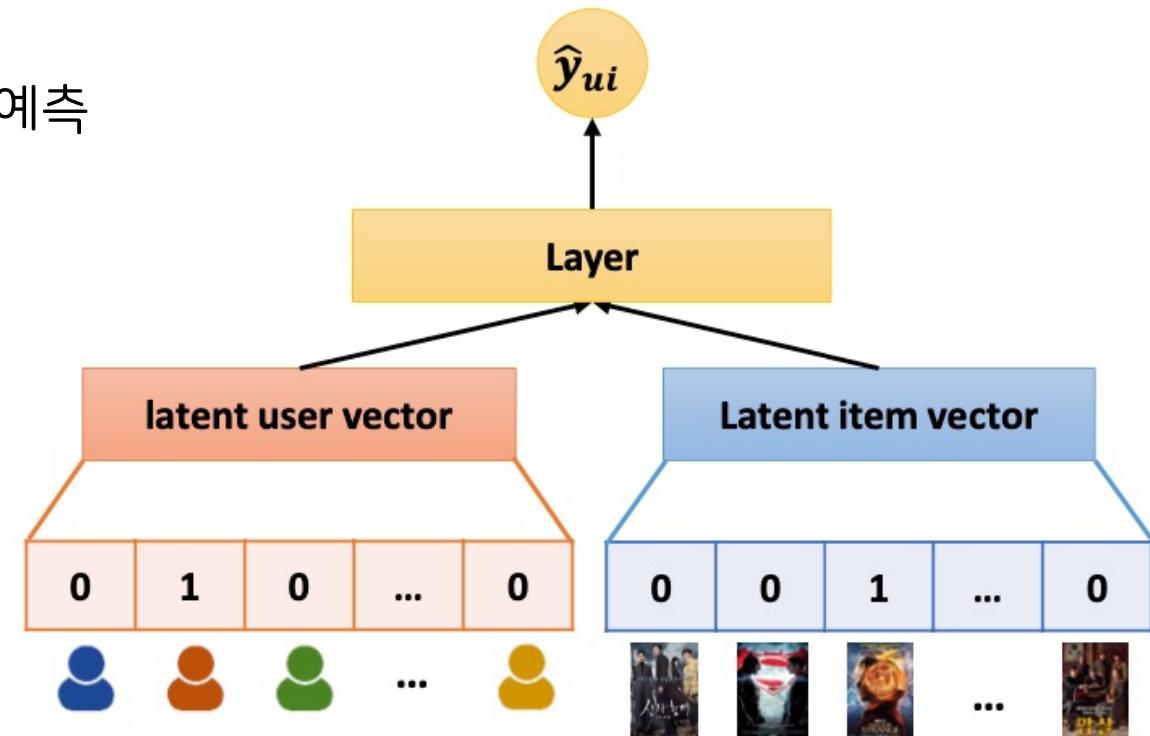
Neural Collaborative Filtering

Generalized Matrix Factorization (GMF)

④ Passing Fully Connected Layer

가중치를 학습하여 유저-아이템 행렬의 값 예측

				...	
	3	3	?	...	2
	?	?	4	...	1
	5	4	?	...	?
:	:	:	:	...	:
	3	?	?	...	3



Unit 02 | MLP-based Matrix Factorization

Neural Collaborative Filtering

Generalized Matrix Factorization (GMF)

④ Passing Fully Connected Layer

가중치를 학습하여 유저-아이템 행렬의 값 예측

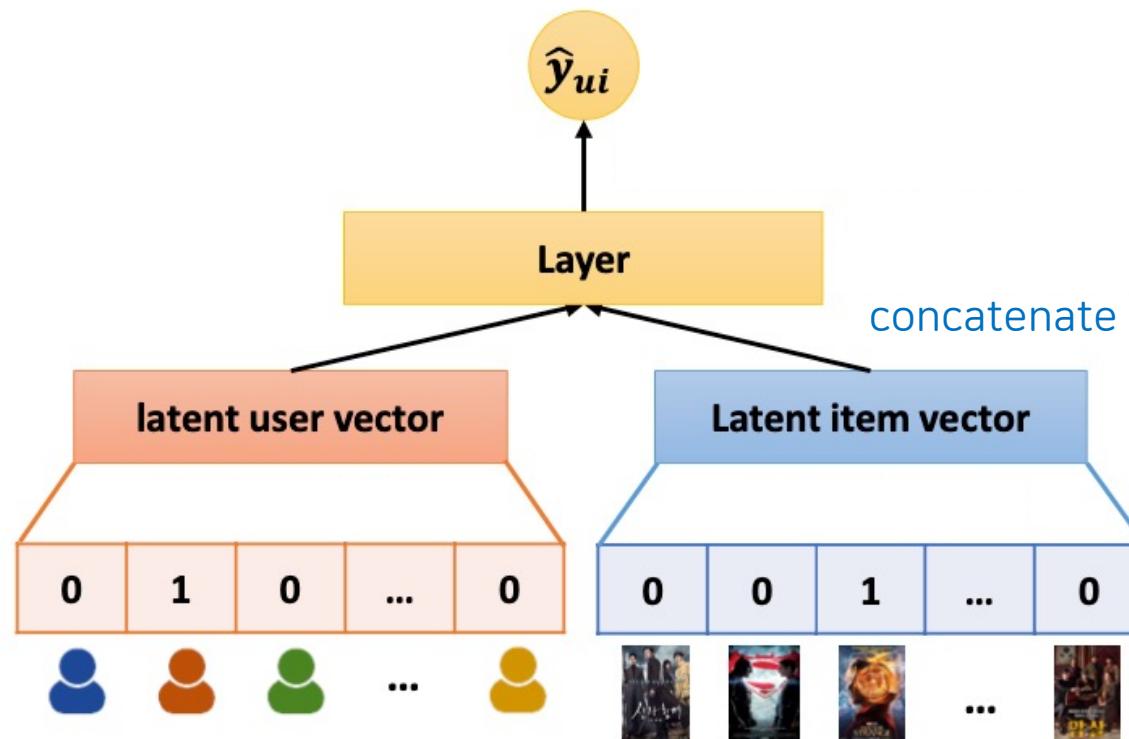
GMF는 element-wise product하여 선형구조를 가지게 됨



Unit 02 | MLP-based Matrix Factorization

Neural Collaborative Filtering

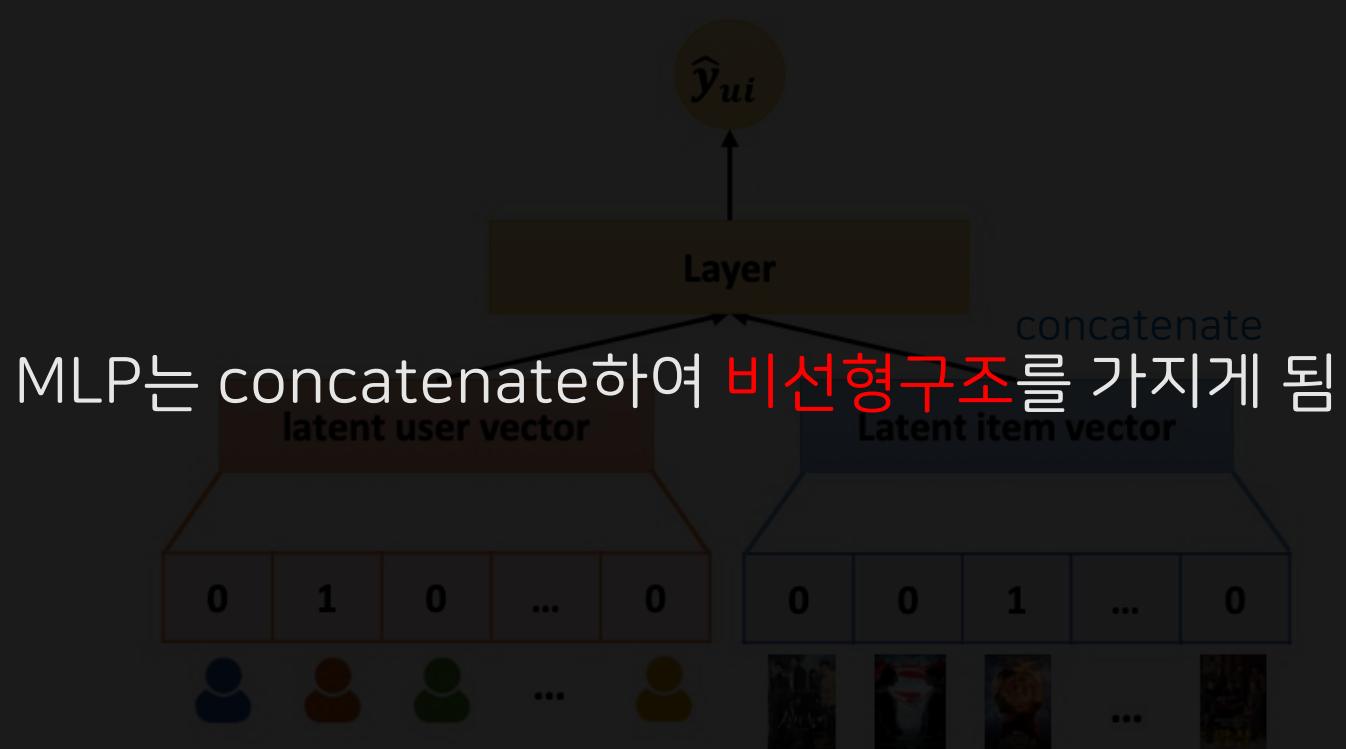
Multilayer Perceptron



Unit 02 | MLP-based Matrix Factorization

Neural Collaborative Filtering

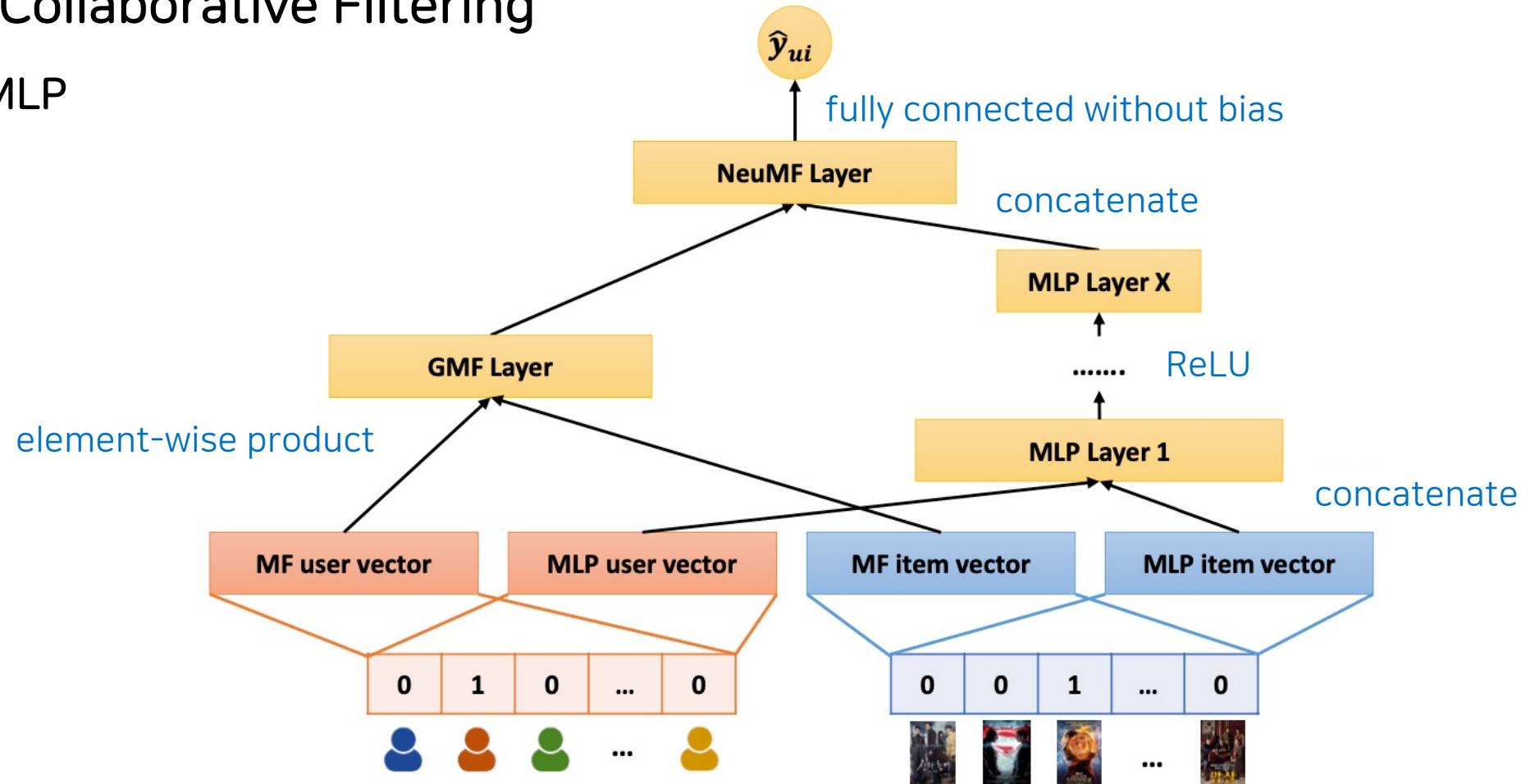
Multilayer Perceptron



Unit 02 | MLP-based Matrix Factorization

Neural Collaborative Filtering

GMF + MLP



Unit 02 | MLP-based Matrix Factorization

Neural Collaborative Filtering

의의

- 유저-아이템 상호작용을 표현하기 위해 Matrix Factorization의 선형성과 Multilayer Perceptron의 비선형성을 결합 (Neural Matrix Factorization)
- 선형 공간에 기반한 기존 모델들이 갖는 한계를 Deep Neural Network를 도입하여 해결

Unit 02 | MLP-based Matrix Factorization

Deep Neural Networks for YouTube Recommendation

기존 YouTube 추천시스템의 문제점

- ① Scale: 엄청난 양의 데이터 & 제한된 컴퓨팅 파워
 - ② Freshness: 새로운 컨텐츠의 빠른 적용
 - ③ Noise: 낮은 질의 메타 데이터, implicit feedback
- Candidate Generation Model + Ranking Model + A/B 테스트를 통한 실제 환경 개선

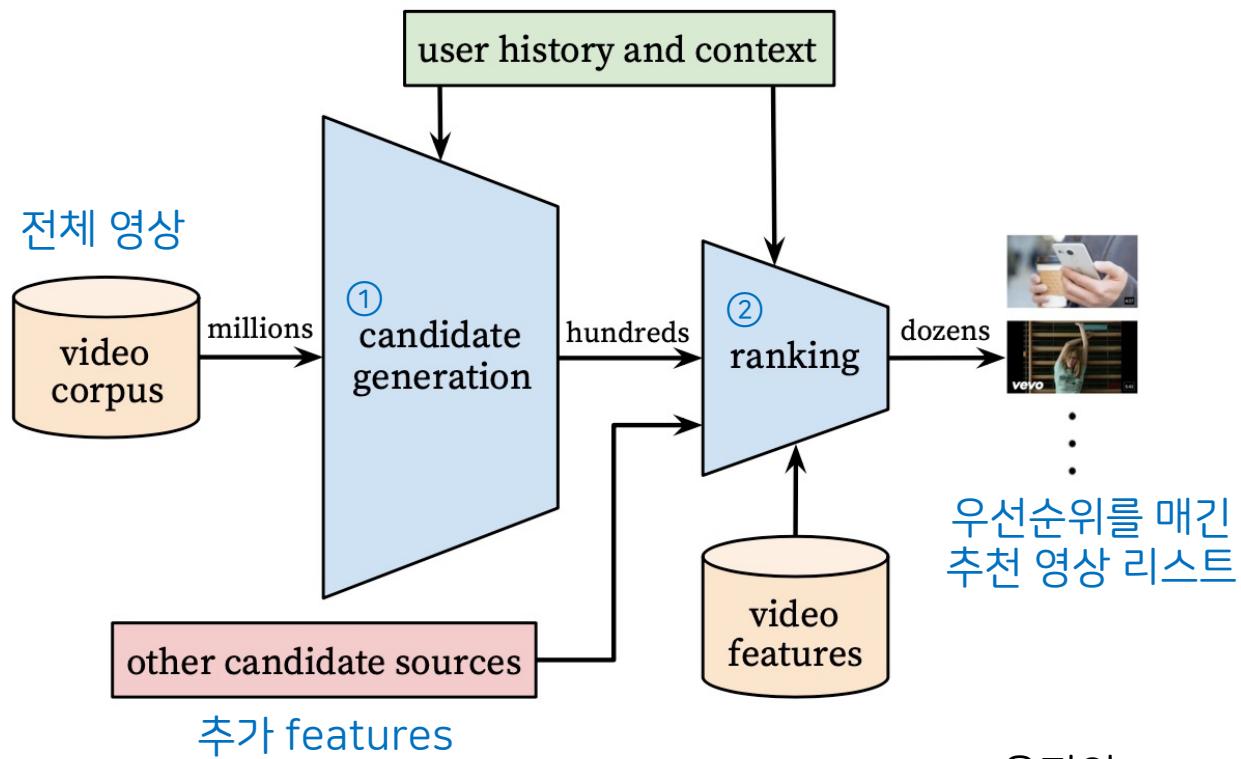
Unit 02 | MLP-based Matrix Factorization

Deep Neural Networks for YouTube Recommendation

Model Overview

① Candidate Generation

유저 정보를 토대로
유사도가 높은 후보 생성



② Ranking

추가 feature를 더하여
성능 개선 후 리스트 생성

우선순위를 매긴
추천 영상 리스트

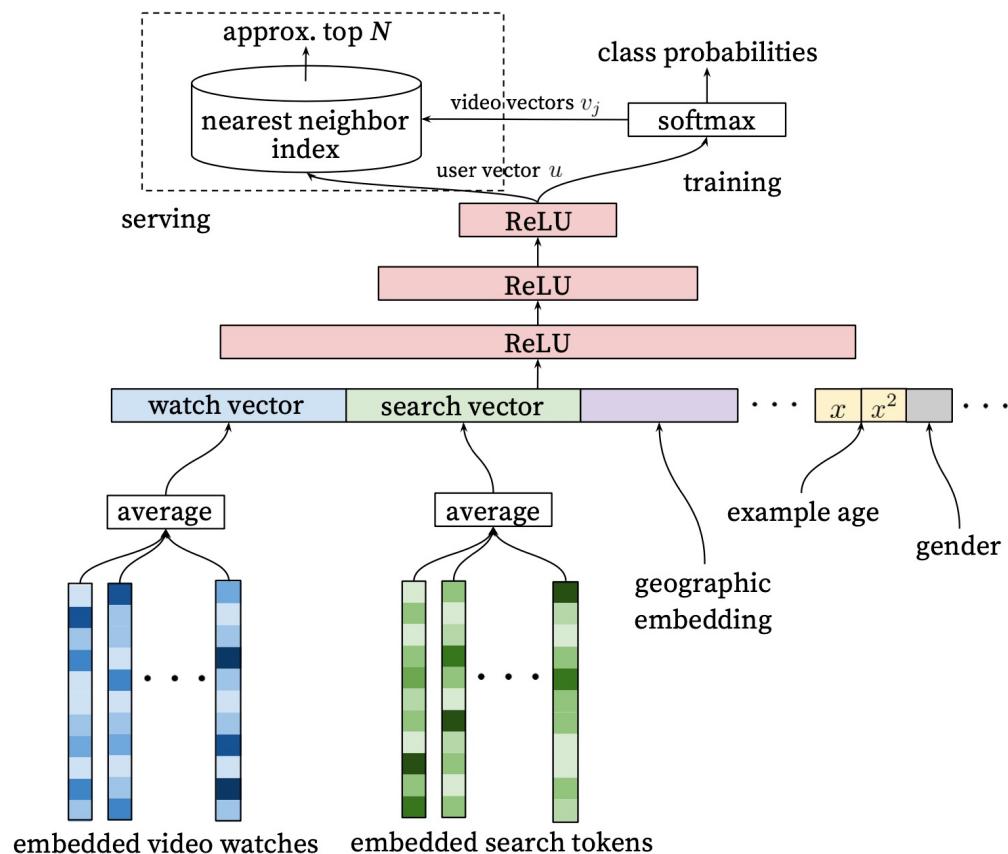
+ A/B 테스트

유저의 engagement를 측정하기 위한 척도
(CTR vs. 감상 시간)

Unit 02 | MLP-based Matrix Factorization

Deep Neural Networks for YouTube Recommendation

Candidate Generation



Unit 02 | MLP-based Matrix Factorization

Deep Neural Networks for YouTube Recommendation

① Candidate Generation

Recommendation as Classification

- Extreme Multiclass Classification
- user, context가 주어지면 특정 시간에 이 영상을 볼 확률

$$P(w_t = i|U, C) = \frac{e^{v_i u}}{\sum_{j \in V} e^{v_j u}}$$

v_j : context embedding
 u : user embedding

→ 유저 정보를 기반으로 한 user embedding을 하고, softmax에서 잘 분류되도록 학습

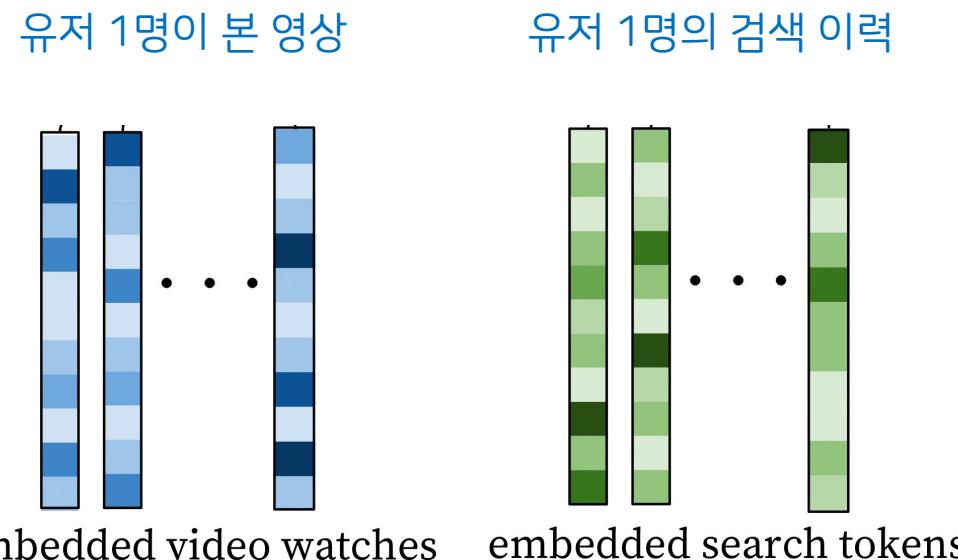
Unit 02 | MLP-based Matrix Factorization

Deep Neural Networks for YouTube Recommendation

① Candidate Generation

① Embedding

- Video Embedding & Search Token Embedding
- 전체 모델의 Backpropagation을 통해 임베딩도 함께 학습



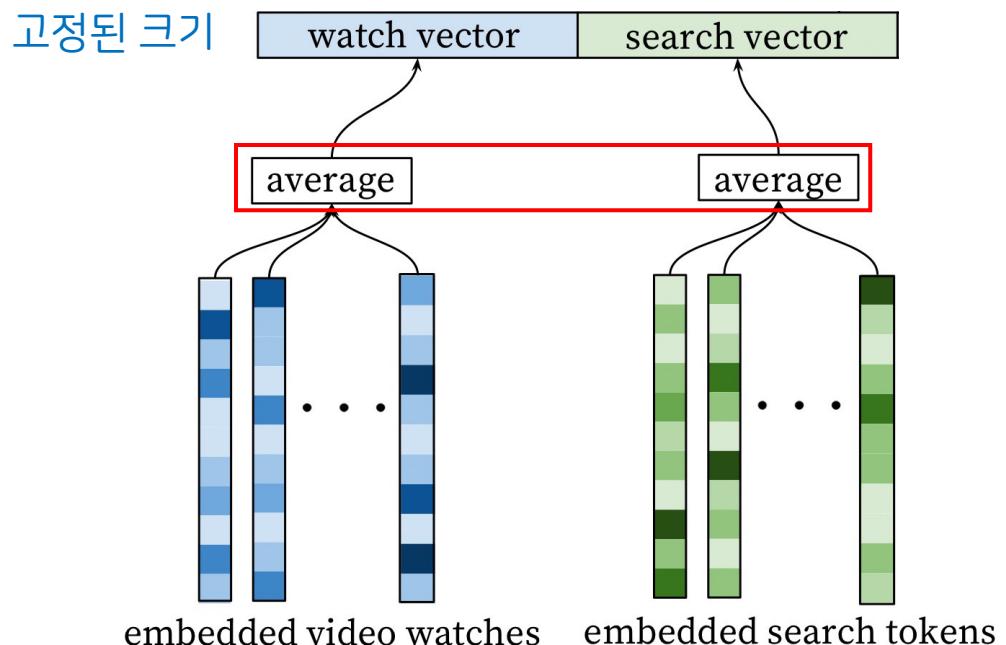
Unit 02 | MLP-based Matrix Factorization

Deep Neural Networks for YouTube Recommendation

① Candidate Generation

② Combining

- 고정된 크기의 input으로 바꿈
- 여러 영상의 embedding vector를 average



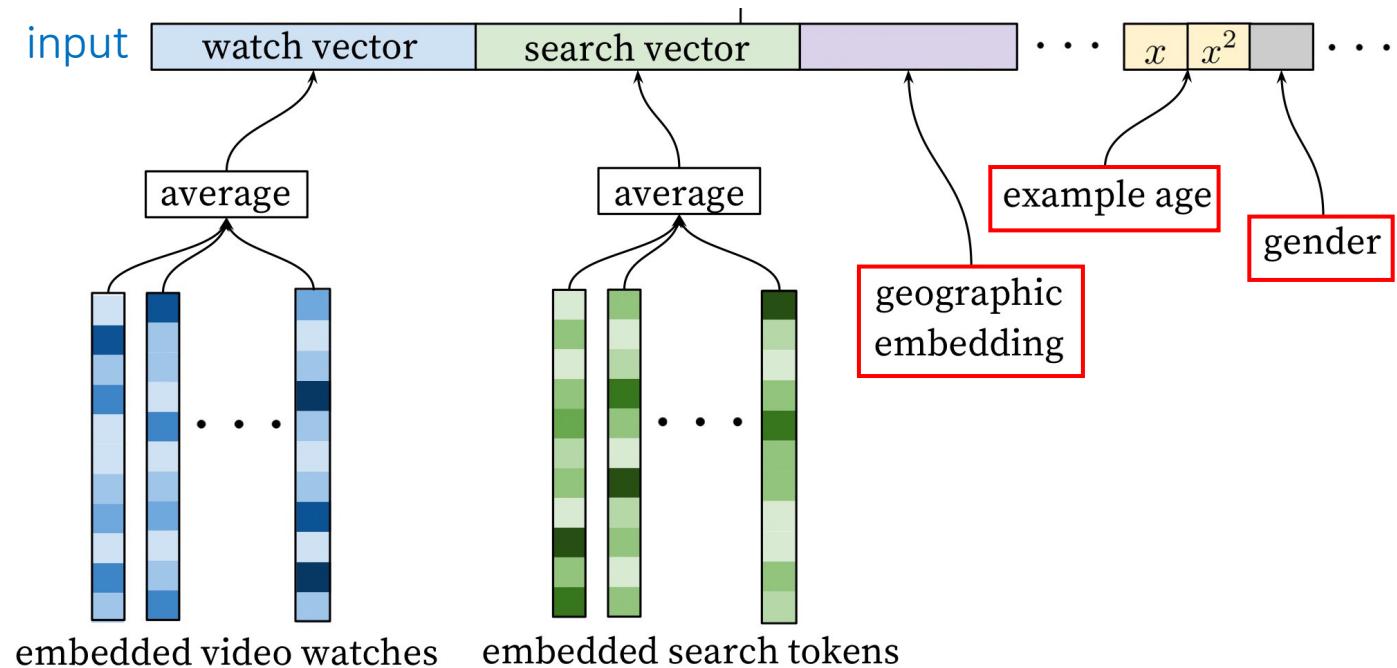
Unit 02 | MLP-based Matrix Factorization

Deep Neural Networks for YouTube Recommendation

① Candidate Generation

③ Additional Features

- 다른 feature 옆에 concatenate



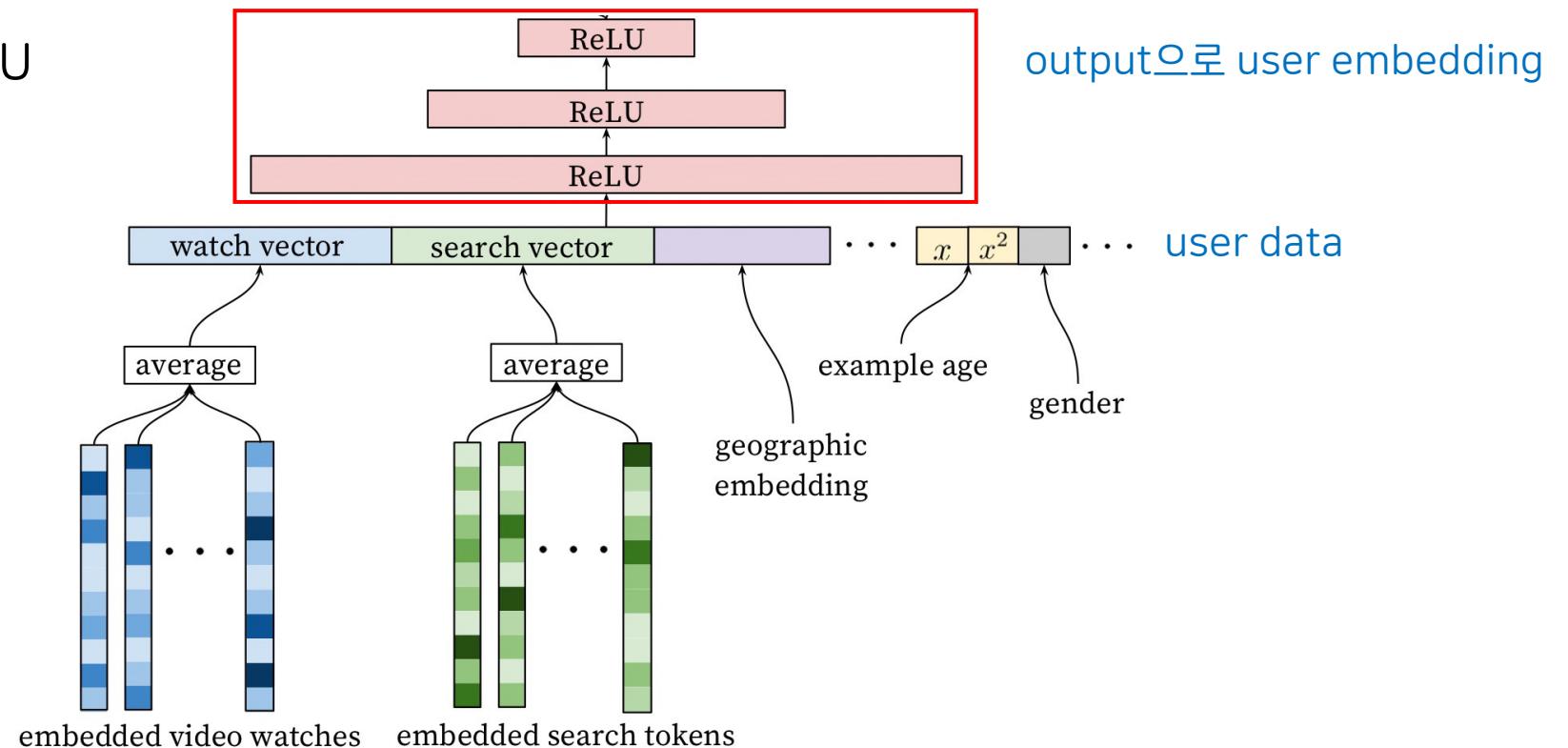
Unit 02 | MLP-based Matrix Factorization

Deep Neural Networks for YouTube Recommendation

① Candidate Generation

④ ReLU Stack

- fully connected ReLU



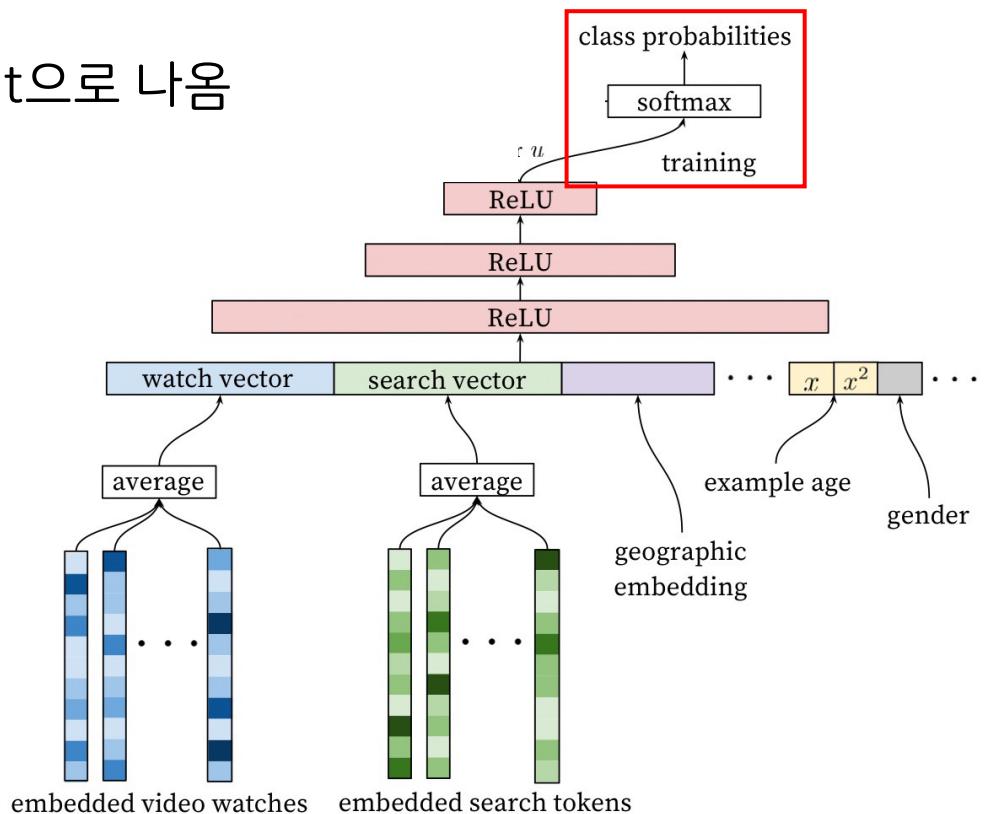
Unit 02 | MLP-based Matrix Factorization

Deep Neural Networks for YouTube Recommendation

⑤ Softmax Prediction

- 어떤 영상을 볼 것으로 예상되는지 영상별 가중치가 output으로 나옴
- Negative Sampling

① Candidate Generation



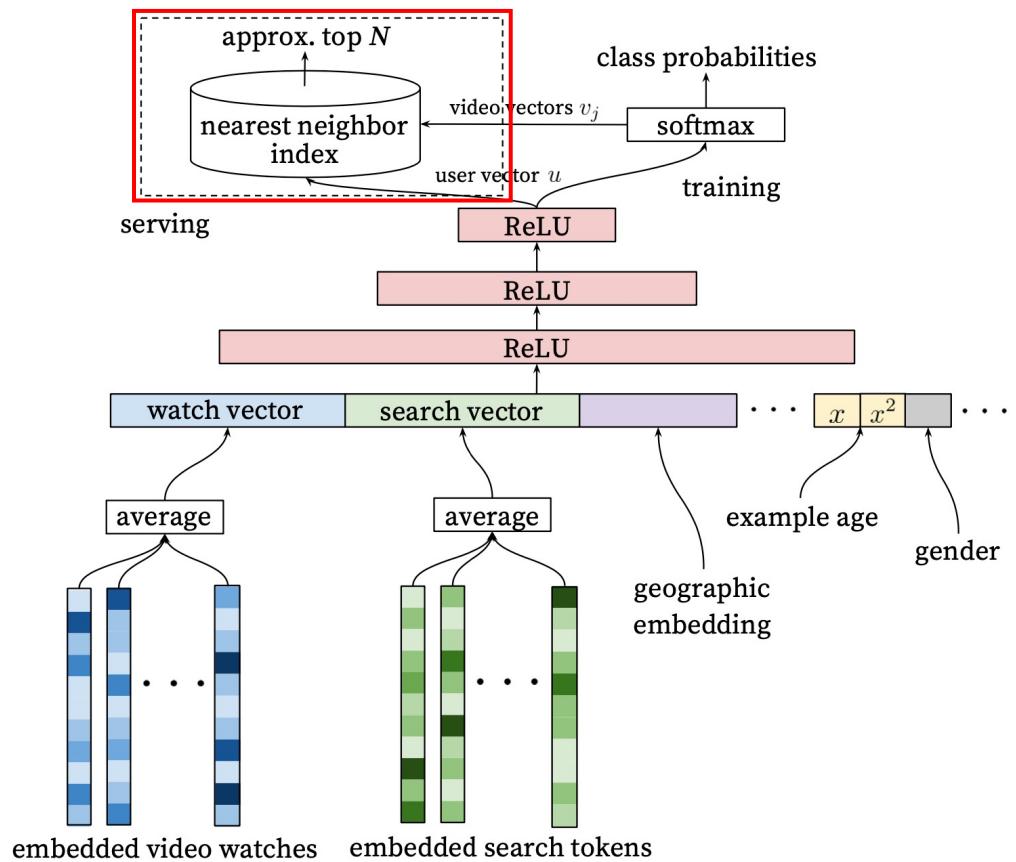
Unit 02 | MLP-based Matrix Factorization

Deep Neural Networks for YouTube Recommendation

⑥ Serving

- 요청이 왔을 때 상위 N개의 영상 추천
 - Dot-product space에서 Nearest Neighbor Index를 뽑아서 가장 가까운 아이템 찾음
- 여전히 실제 환경과 차이가 있으므로 추가 feature

① Candidate Generation



Unit 02 | MLP-based Matrix Factorization

Deep Neural Networks for YouTube Recommendation

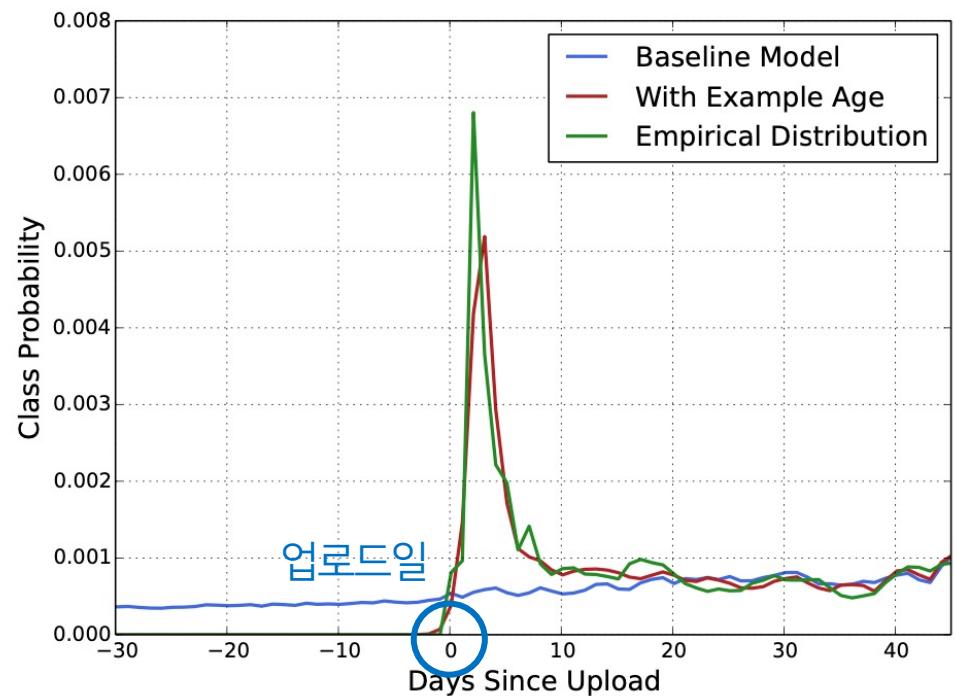
① Candidate Generation

"Example Age" Feature

- 히스토리 데이터를 기반으로 학습하면 오래된 아이템들이 더 많이 추천받음
- freshness의 중요성

그 외 features

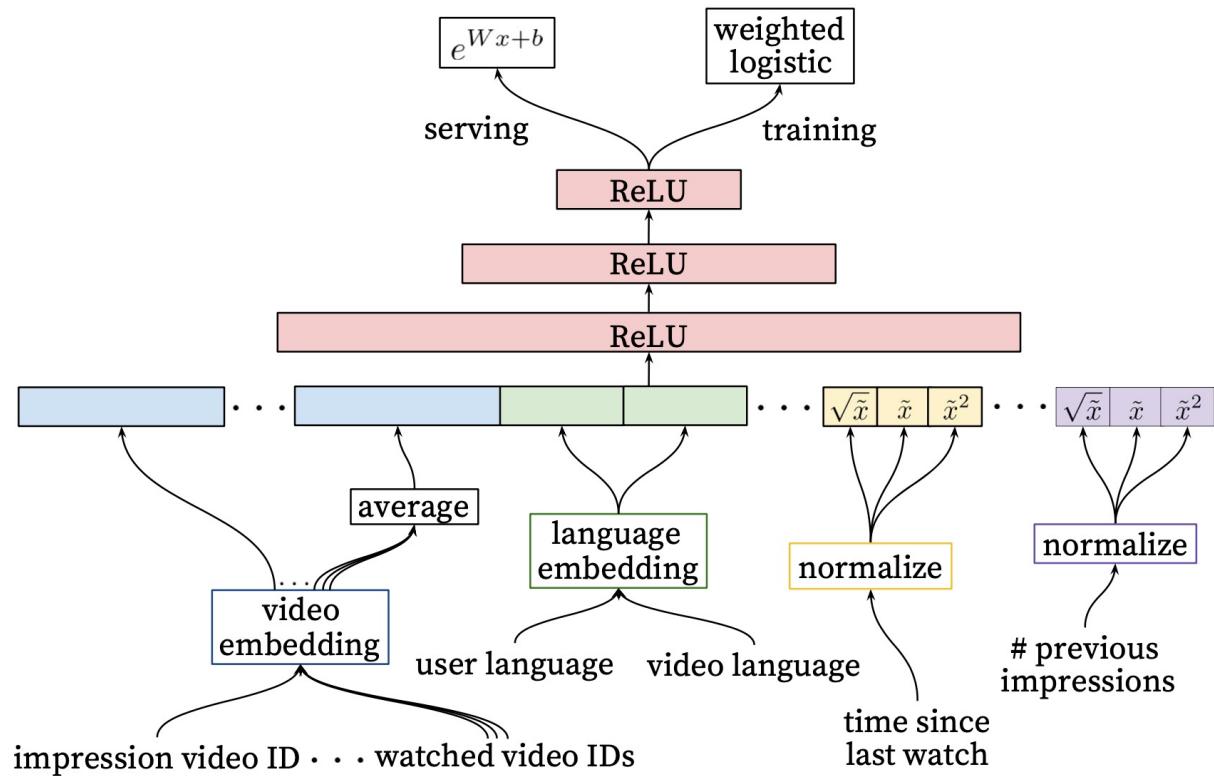
- 모든 영상 시청 이력을 확인해야 편향이 생기지 않음
- 학습에 사용할 유저별 영상 횟수를 고정
- 새로운 검색 쿼리에 즉시 추천 엔진을 반영하지 않음
- 비대칭적 감상 패턴 적용해서 학습



Unit 02 | MLP-based Matrix Factorization

Deep Neural Networks for YouTube Recommendation

Ranking Network

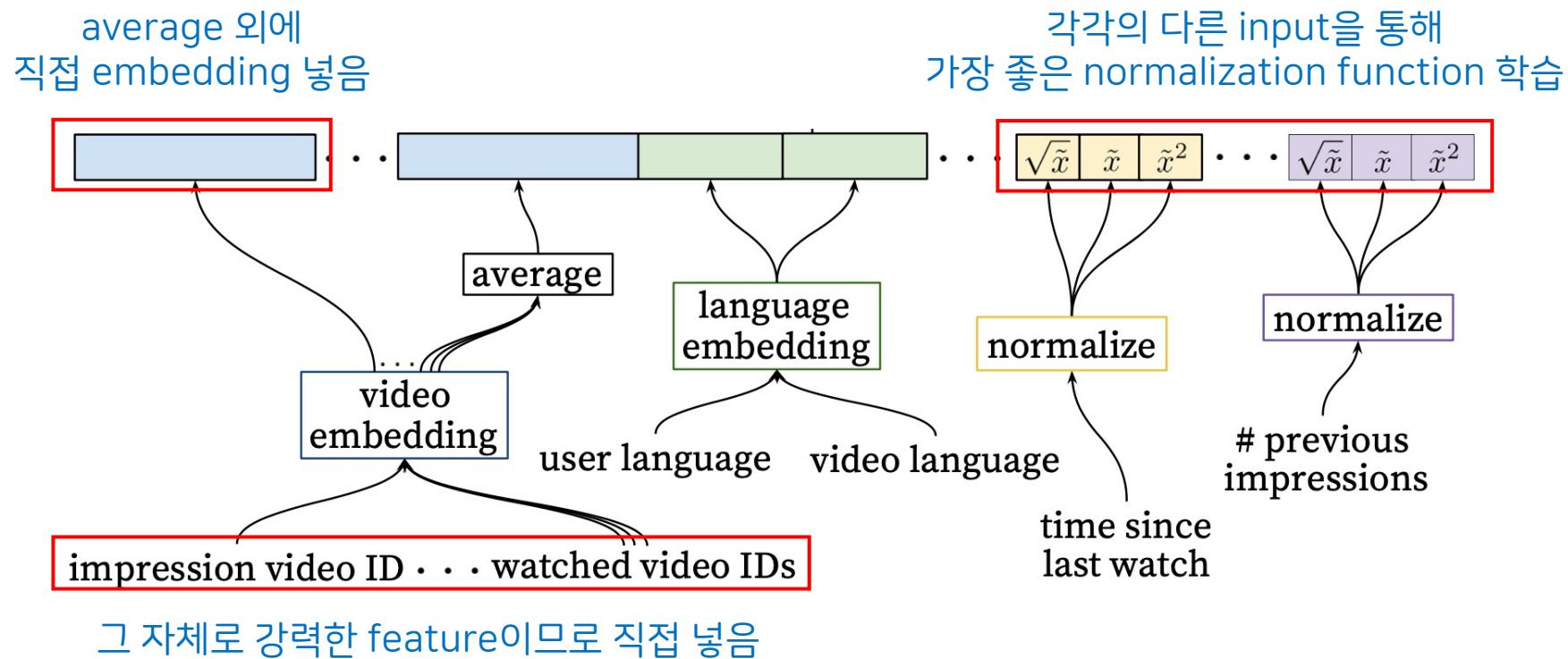


Unit 02 | MLP-based Matrix Factorization

Deep Neural Networks for YouTube Recommendation

② Ranking

Embedding



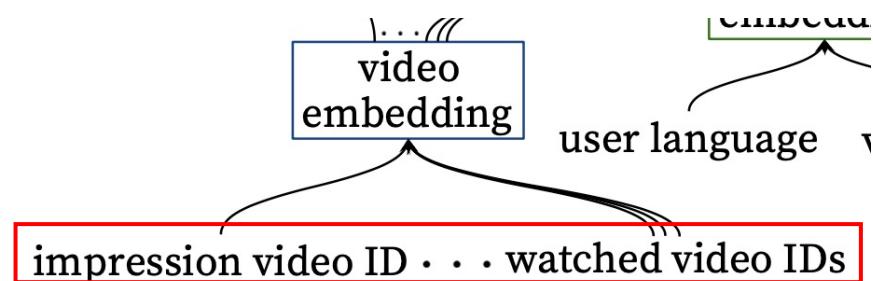
Unit 02 | MLP-based Matrix Factorization

Deep Neural Networks for YouTube Recommendation

② Ranking

Feature Engineering

- feature 분류
 - univalent vs. multivalent
 - impression vs. query
- feature가 매우 많아서 미리 가공
- 딥러닝이지만 엔지니어가 직접 가공



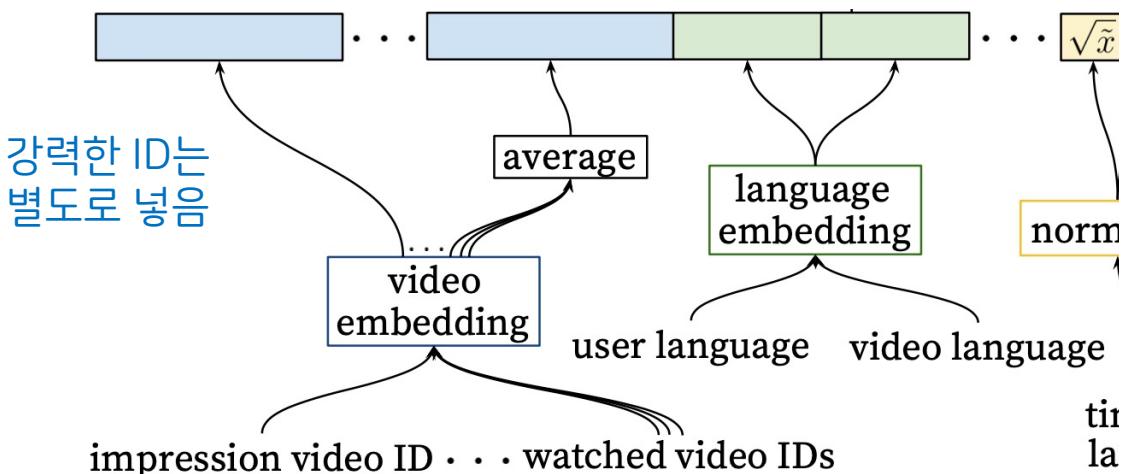
Unit 02 | MLP-based Matrix Factorization

Deep Neural Networks for YouTube Recommendation

② Ranking

Embedding Categorical Features

- unique ID space (“vocabulary”) 따로 임베딩
- 이 중에 impression 기준으로 Top N 영상 및 검색 이력 임베딩
- vocabulary에 없으면 zero embedding



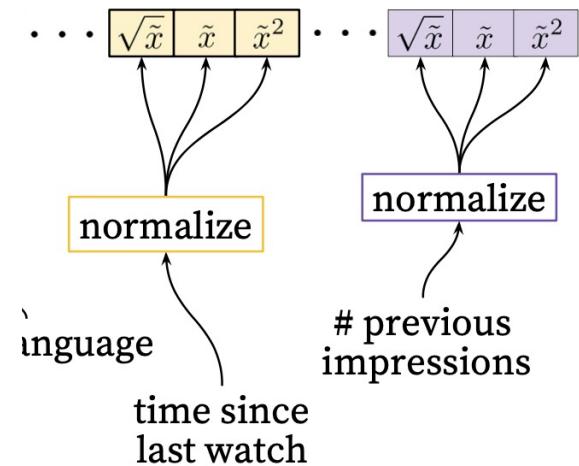
Unit 02 | MLP-based Matrix Factorization

Deep Neural Networks for YouTube Recommendation

② Ranking

Normalizing Continuous Features

- 값 x 를 $[0, 1)$ 에 들어오도록 scaling
- $\tilde{x}, \tilde{x}^2, \sqrt{\tilde{x}}$ 도 네트워크에 넣음



Unit 02 | MLP-based Matrix Factorization

Deep Neural Networks for YouTube Recommendation

② Ranking

Modeling Expected Watch Time

- 추천한 영상의 감상 시간 예측
- 감상 시간은 안 봤으면 0, 봤으면 본 시간을 값으로 넣음
- Weighted Logistic Regression: 감상한 영상을 감상 시간으로 가중치

$$\frac{\sum T_i}{N - k}$$



$$E[T](1 + P)$$

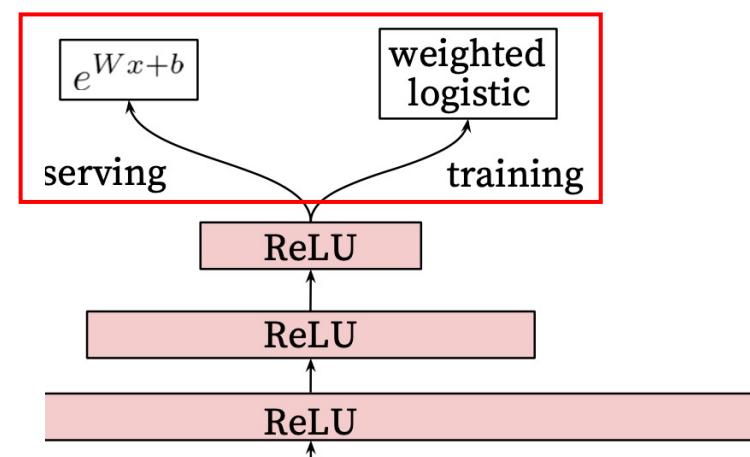
N: training example 수

k: positive expression 수

 T_i : i번째 노출에 감상 시간

P: click probability

E[T]: expected watch time



Unit 02 | MLP-based Matrix Factorization

Deep Neural Networks for YouTube Recommendation

의의

- 기존의 방법인 Matrix Factorization보다 성능 향상
- 딥러닝으로만 feature 학습하기가 어려워서 hand-written feature 포함
- 영상의 나이 잘 작동
- 사용자의 과거 행동 패턴을 잘 설명하는 feature 중요
- 감상 시간별로 가중치

03 RNN-based Model

Session-based Recommendations with Recurrent Neural Networks

Neural Attentive Session-based Recommendation

Unit 03 | RNN-based Model

Recurrent Neural Network based Matrix Factorization

순차적 데이터 처리에 유용한 RNN을 사용

상호작용의 일시적 역학, 유저 행동의 순차적 패턴, 순차적 신호를 가진 side information 처리

- 1) Session-based Recommendations with Recurrent Neural Networks
- 2) Neural Attentive Session-based Recommendation

Unit 03 | RNN-based Model

Session-based Recommendations with Recurrent Neural Networks

기존 Session-based의 어려움

- 유저가 로그인하지 않은 환경에서는 latent vector 사용이 어려움
 - session이나 cookie mechanism이 있지만 데이터가 매우 적음
- 유저의 순서 정보를 이용해서 추천시스템에 적용할 수 있는 방법을 RNN 기반으로 제안

Unit 03 | RNN-based Model

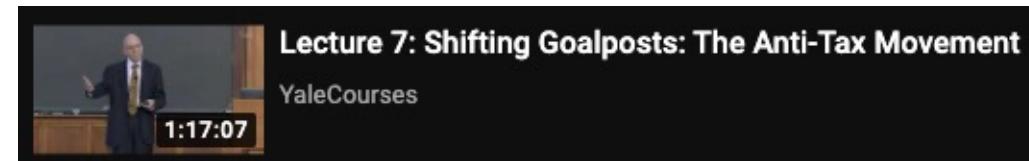
Session-based Recommendations with Recurrent Neural Networks

기존 Session-based의 어려움

- 어떤 아이템을 경험했는가? + 가장 최근에 어떤 아이템을 클릭했는가?



session input



recommendation

Unit 03 | RNN-based Model

Session-based Recommendations with Recurrent Neural Networks

Markov Decision Process

- 순차적 확률 기반 결정 문제
 - $\langle S, A, Rwd, tr \rangle$ 에서 action은 recommendation이 됨
- ※ 모든 가능한 유저 선택의 시퀀스를 고려하려고 하면 상태 공간이 감당할 수 없어짐

General Factorization Framework (GFF)

- 세션 데이터를 추천에 이용 가능
 - 아이템 자체 & 세션의 일부로서의 아이템의 latent representation
- ※ 그러나 세션 안에서는 순서를 고려하지 않음

Unit 03 | RNN-based Model

Session-based Recommendations with Recurrent Neural Networks

Gated Recurrent Units (GRU)

① Reset Gate

$$r^{(t)} = \sigma(W_r h^{(t-1)} + U_r x^{(t)})$$

② Update Gate

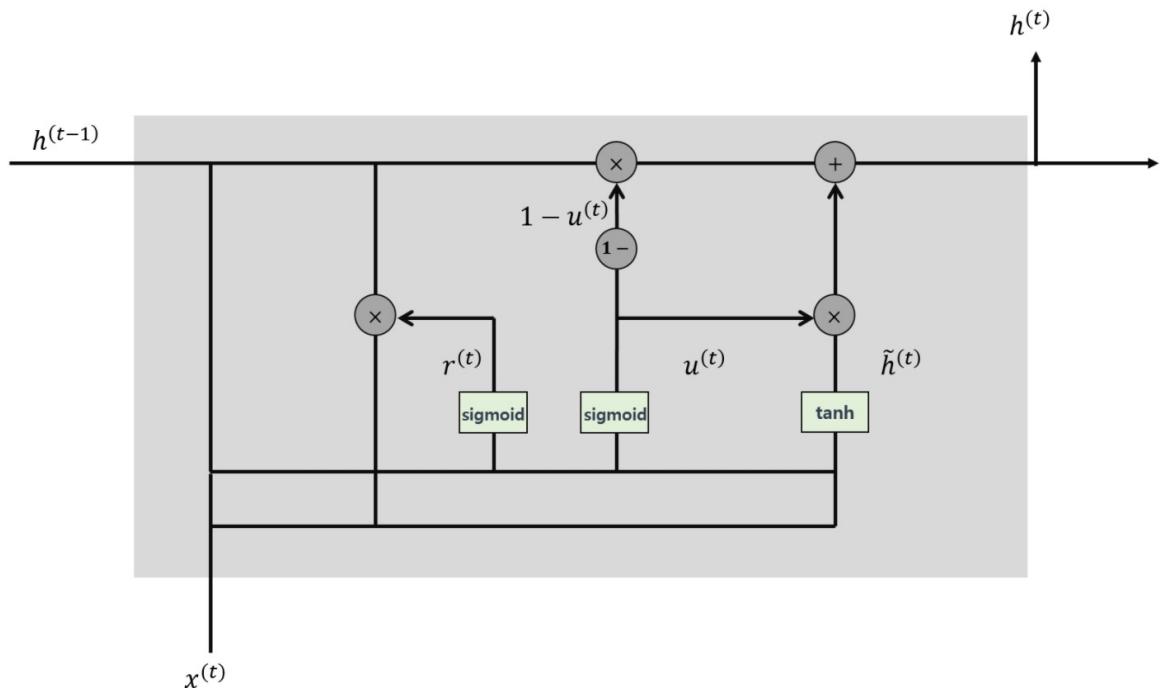
$$u^{(t)} = \sigma(W_u h^{(t-1)} + U_u x^{(t)})$$

③ Candidate

$$\tilde{h}^{(t)} = \tanh(W h^{(t-1)} * r^{(t)} + U x^{(t)})$$

④ Hidden State 계산

$$h^{(t)} = (1 - u^{(t)}) * h^{(t-1)} + u^{(t)} * \tilde{h}^{(t)}$$

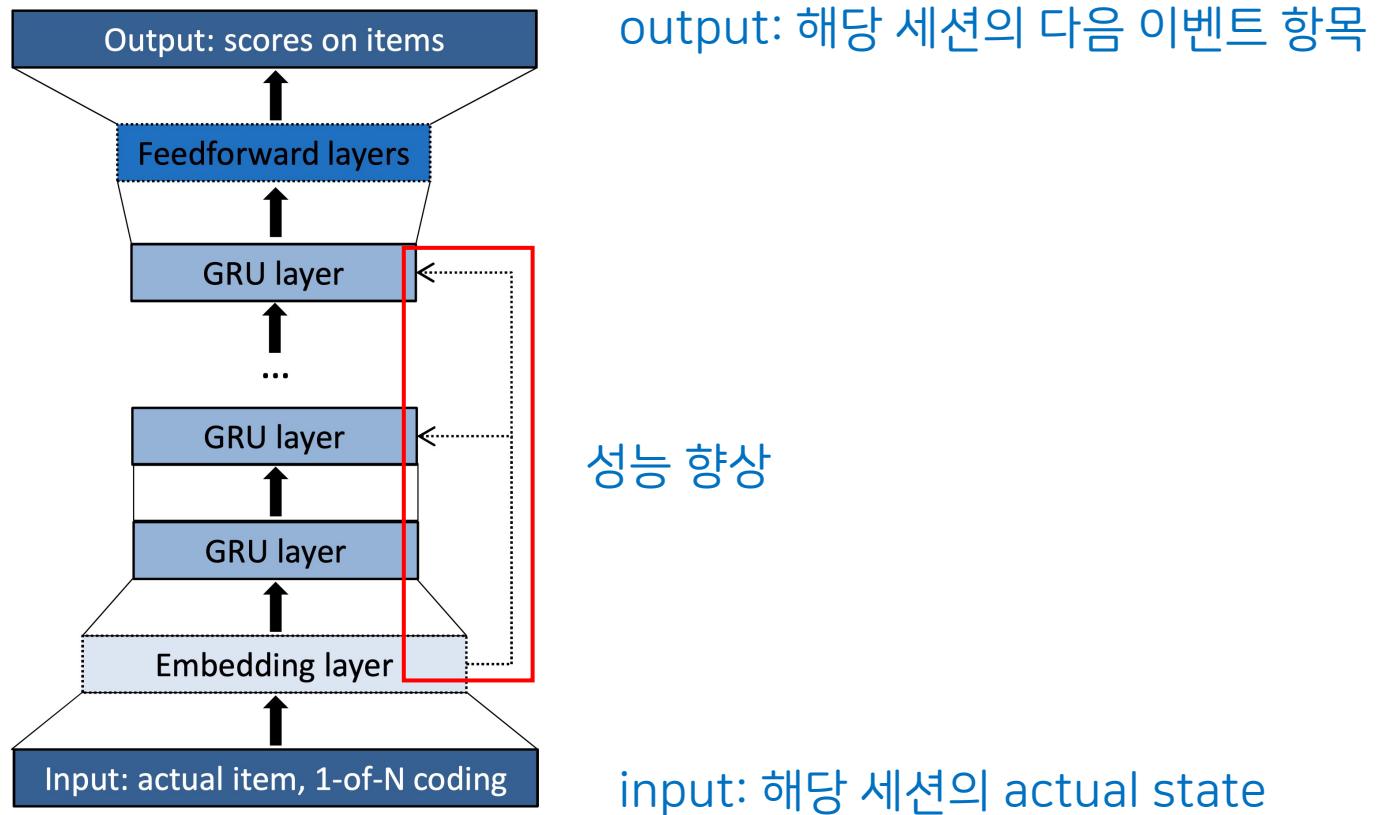


LSTM보다 학습할 가중치가 적음

Unit 03 | RNN-based Model

Session-based Recommendations with Recurrent Neural Networks

Model Overview



Unit 03 | RNN-based Model

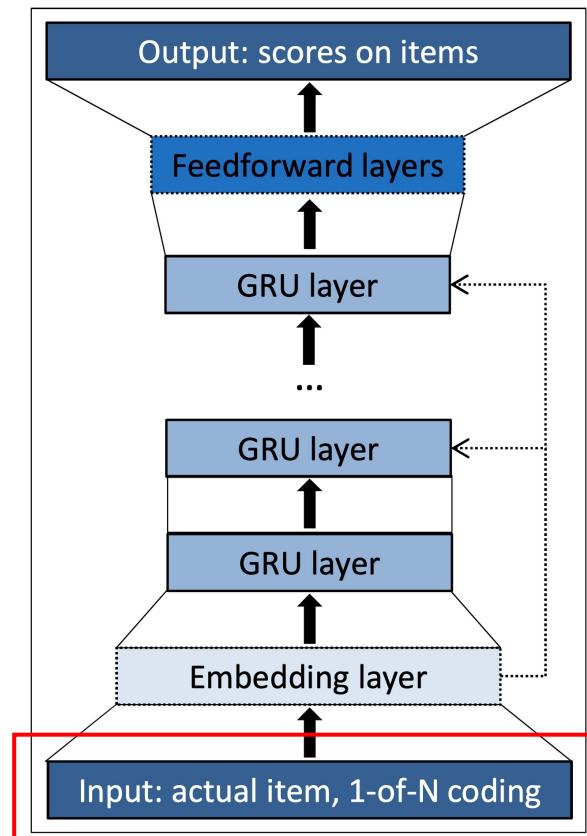
Session-based Recommendations with Recurrent Neural Networks

GRU4Rec

① Input Layer

input으로 Actual State of Session 넣음

- Actual State of Session
 - Actual Event의 아이템 : 1-of-N encoding vector $(0, 0, 1, \dots, 0)$
 - 세션 안의 이벤트 : weighted sum vector $(0, 0.6, 0.4, \dots, 0)$
더 일찍 일어난 사건은 discount 됨



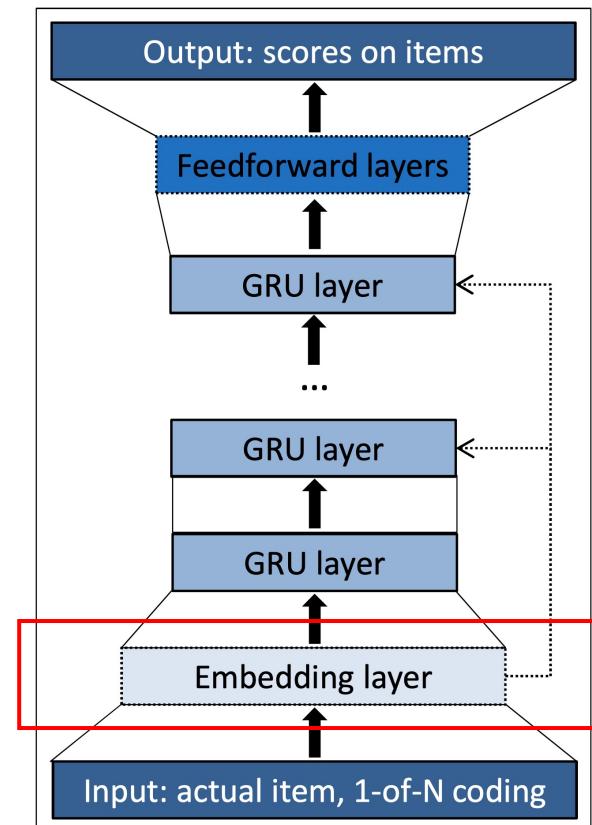
Unit 03 | RNN-based Model

Session-based Recommendations with Recurrent Neural Networks

GRU4Rec

② Embedding Layer

RNN 특성상 long term에 약하기 때문에 안정성을 위해 정규화



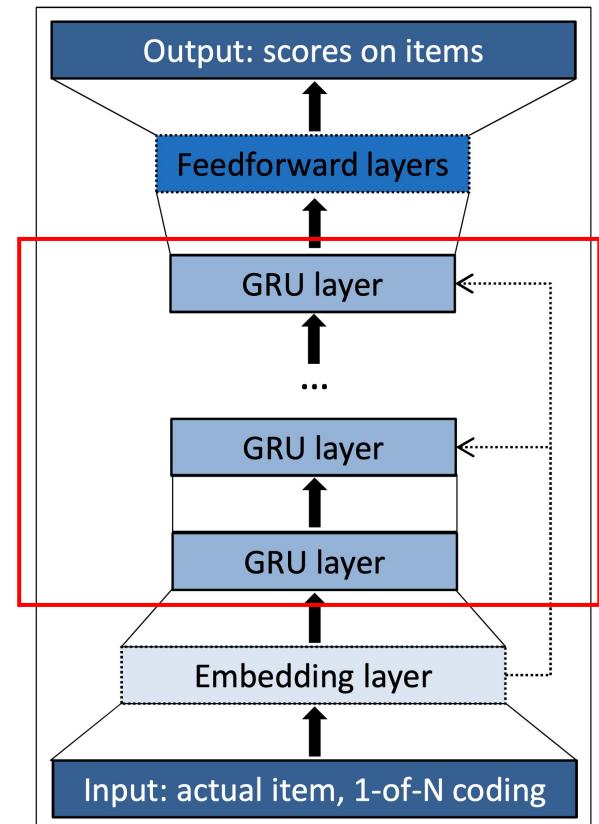
Unit 03 | RNN-based Model

Session-based Recommendations with Recurrent Neural Networks

GRU4Rec

③ GRU Layer

아이템에 대한 선호도 예측하는 학습



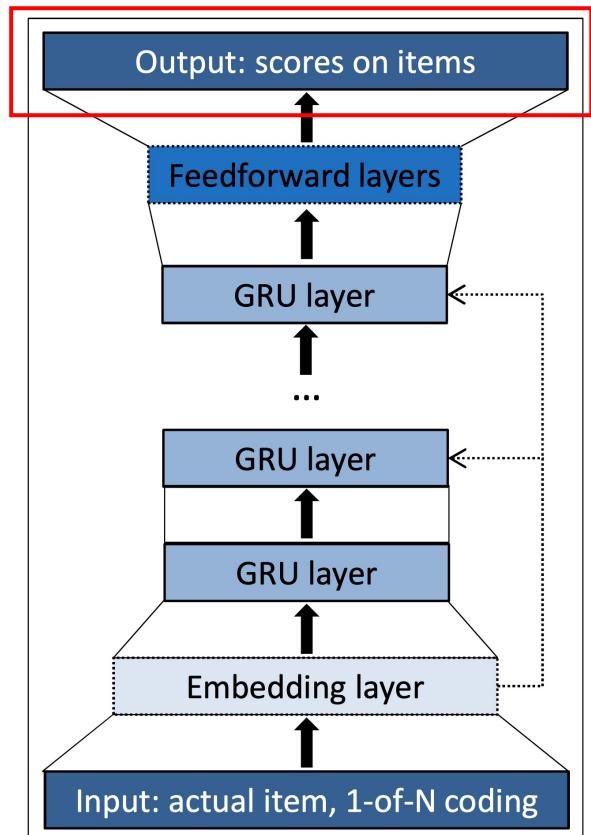
Unit 03 | RNN-based Model

Session-based Recommendations with Recurrent Neural Networks

GRU4Rec

④ Output Layer

해당 세션의 다음 이벤트 아이템 출력

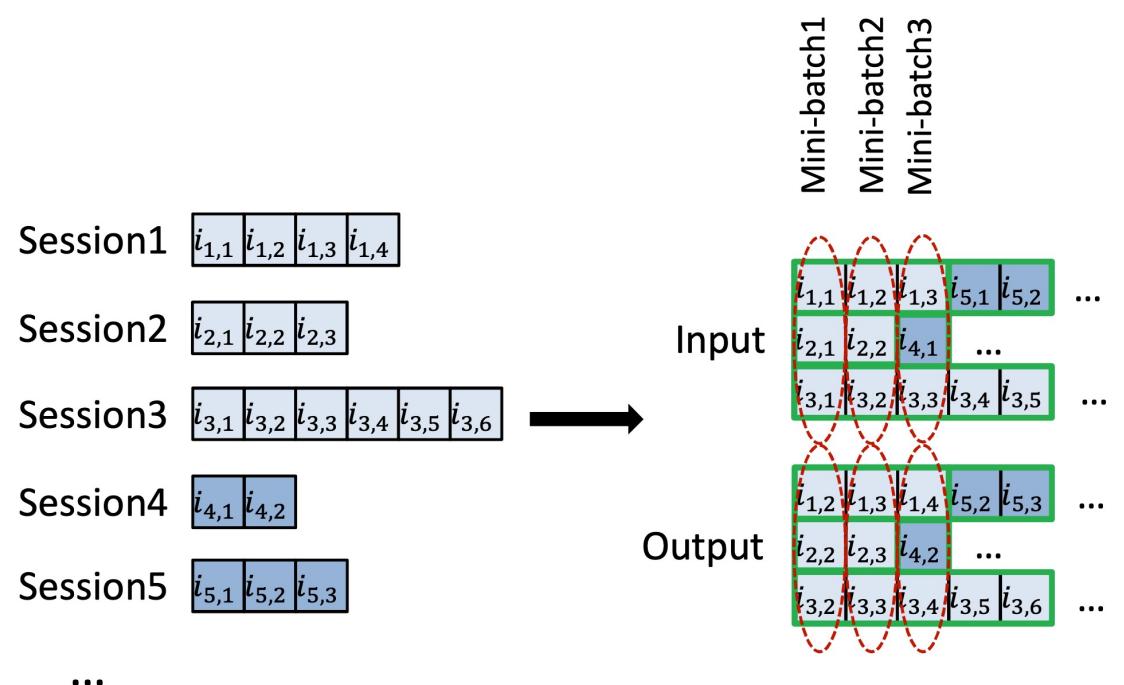


Unit 03 | RNN-based Model

Session-based Recommendations with Recurrent Neural Networks

Session-parallel Mini-batch

- 세션 병렬 방식의 Mini-batch로 input을 넣음
- 다음 세션 정보가 들어올 때 hidden state 초기화

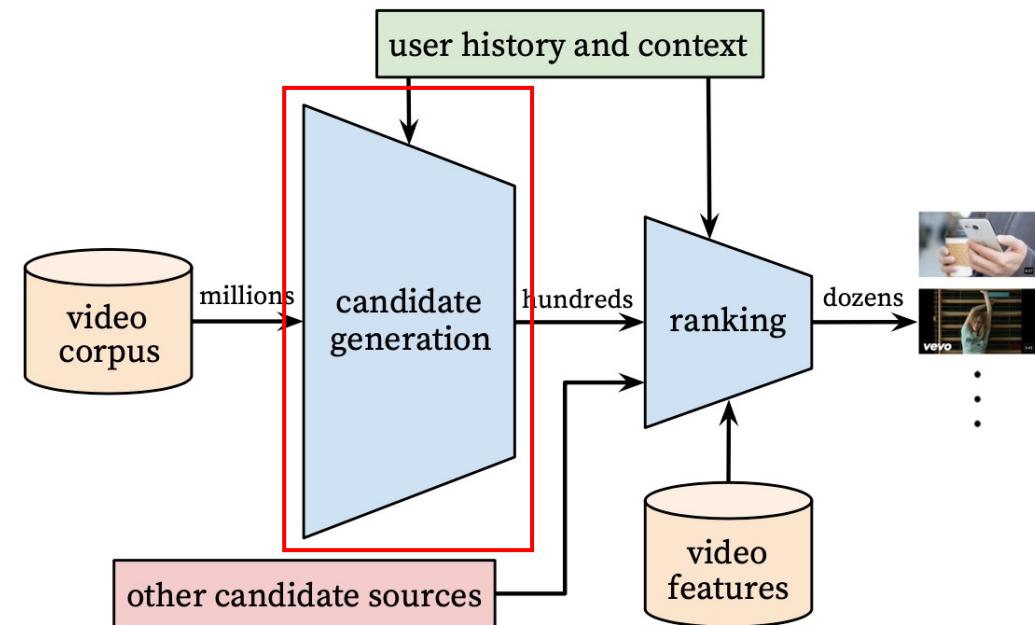


Unit 03 | RNN-based Model

Session-based Recommendations with Recurrent Neural Networks

Sampling on the Output

- output을 샘플링해서 하위 집합만 연산
- Negative Sampling도 함께 시행
- 둘 다 popularity-based sampling



Unit 03 | RNN-based Model

Session-based Recommendations with Recurrent Neural Networks

Pairwise Ranking

- positive & negative 쌍으로 비교
- loss를 통해 positive item의 rank가 negative item의 rank보다 낮게 학습
- Bayesian Personalized Ranking(BPR)과 TOP1 사용

$$\mathcal{L}_s = \frac{1}{S} \sum_{j=1}^S \sigma(\hat{r}_{sj} - \hat{r}_{si}) + \sigma(\hat{r}_{sj}^2)$$

TOP1 BPR

Unit 03 | RNN-based Model

Session-based Recommendations with Recurrent Neural Networks

의의

- GRU라는 RNN을 추천시스템에 적용
- session-parallel mini-batch과 ranking loss function 이용한 session-based 추천시스템

Unit 03 | RNN-based Model

Neural Attentive Session-based Recommendation

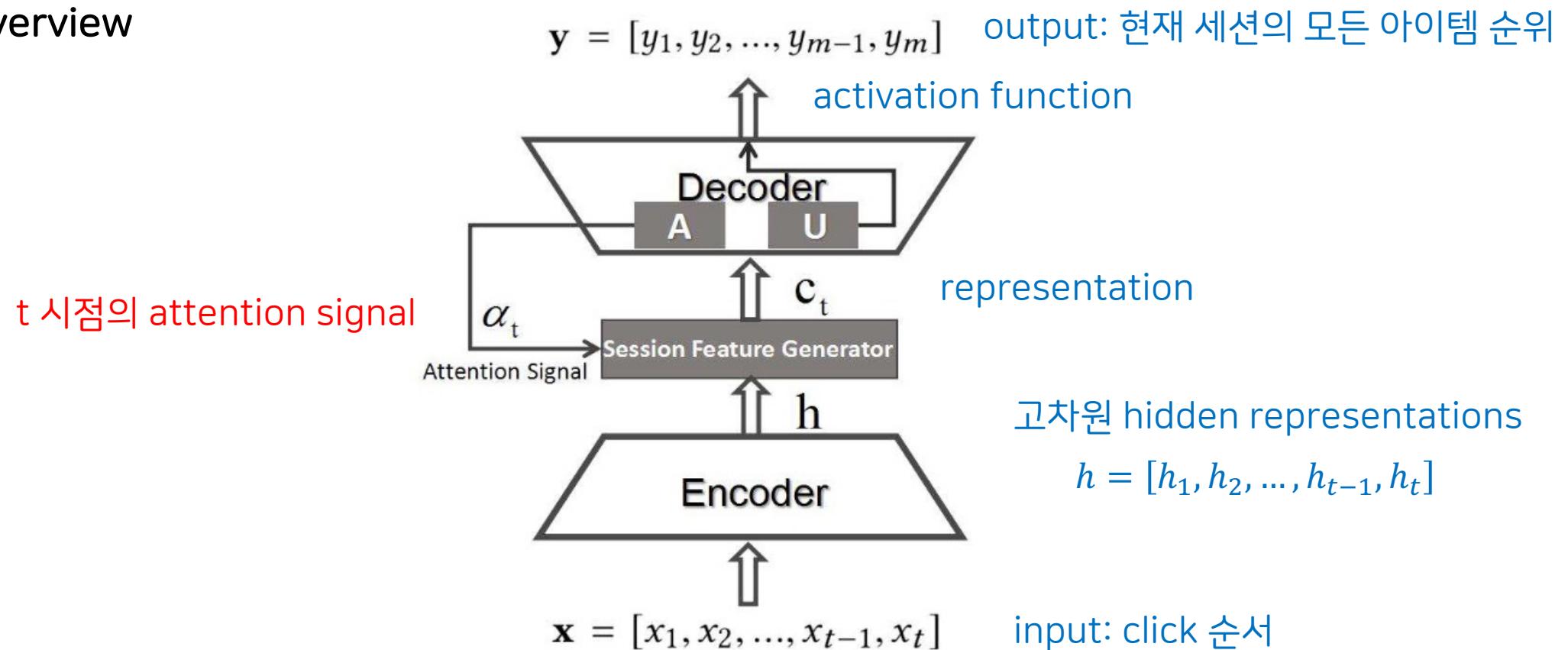
기존 RNN-based Model의 문제점

- 최근 순차적 정보만 고려하면 유저의 주요 목적을 놓치게 됨
- 유저의 순차적 행동 & 현재 세션에서의 주요 목적을 함께 고려

Unit 03 | RNN-based Model

Neural Attentive Session-based Recommendation

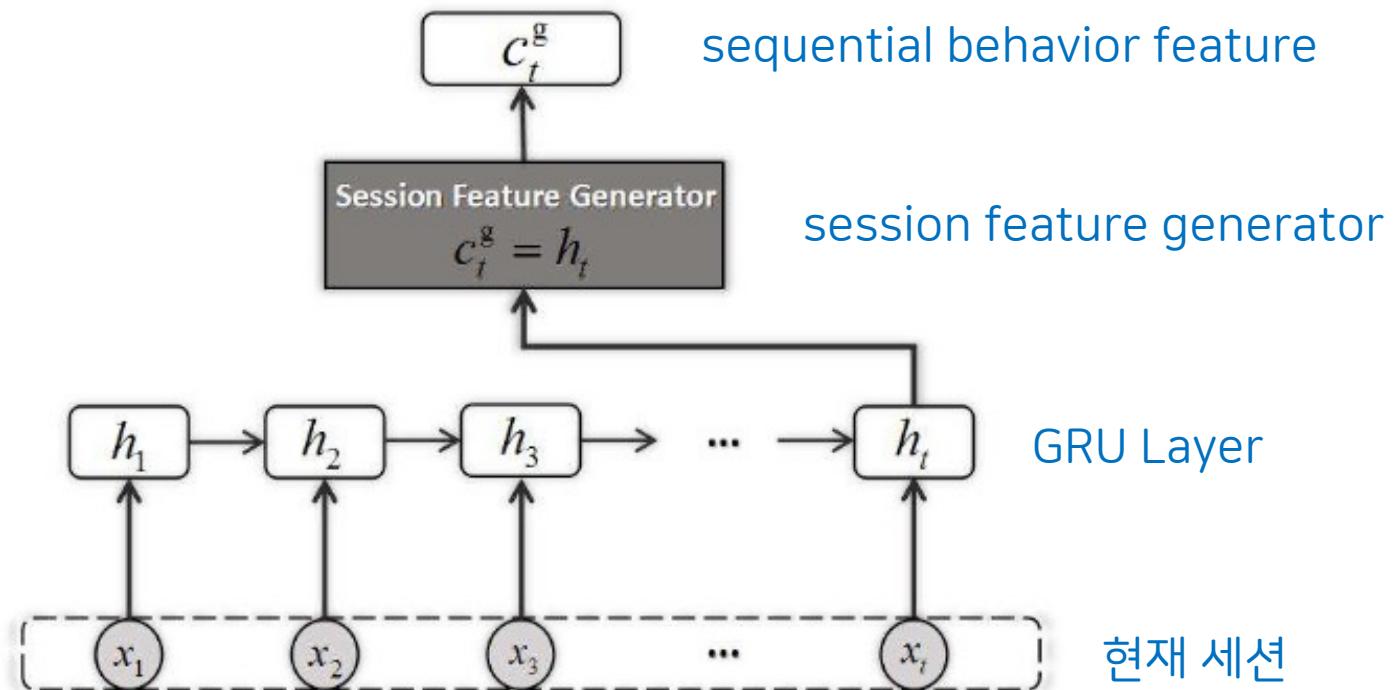
Model Overview



Unit 03 | RNN-based Model

Neural Attentive Session-based Recommendation

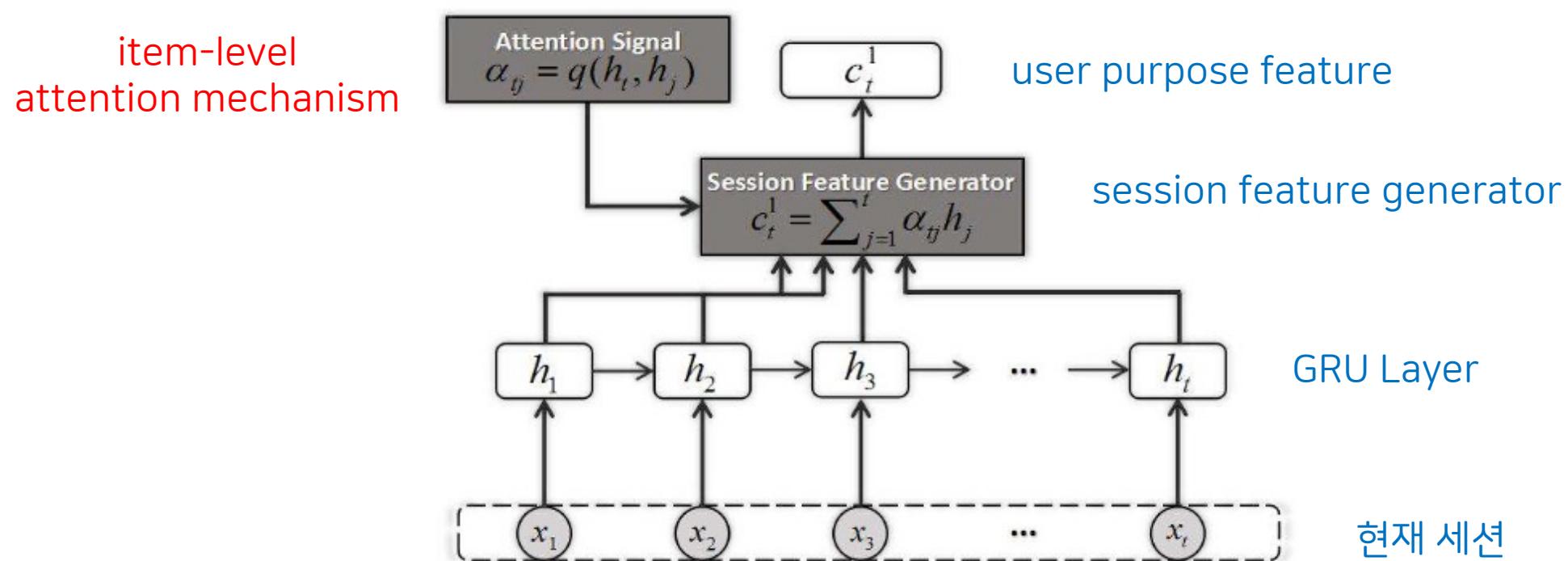
Global Encoder



Unit 03 | RNN-based Model

Neural Attentive Session-based Recommendation

Local Encoder



Unit 03 | RNN-based Model

Neural Attentive Session-based Recommendation

Item-level Attention Mechanism

- 디코더가 입력 시퀀스의 다른 부분들을 다양하게 고르고 선형결합 할 수 있도록 함

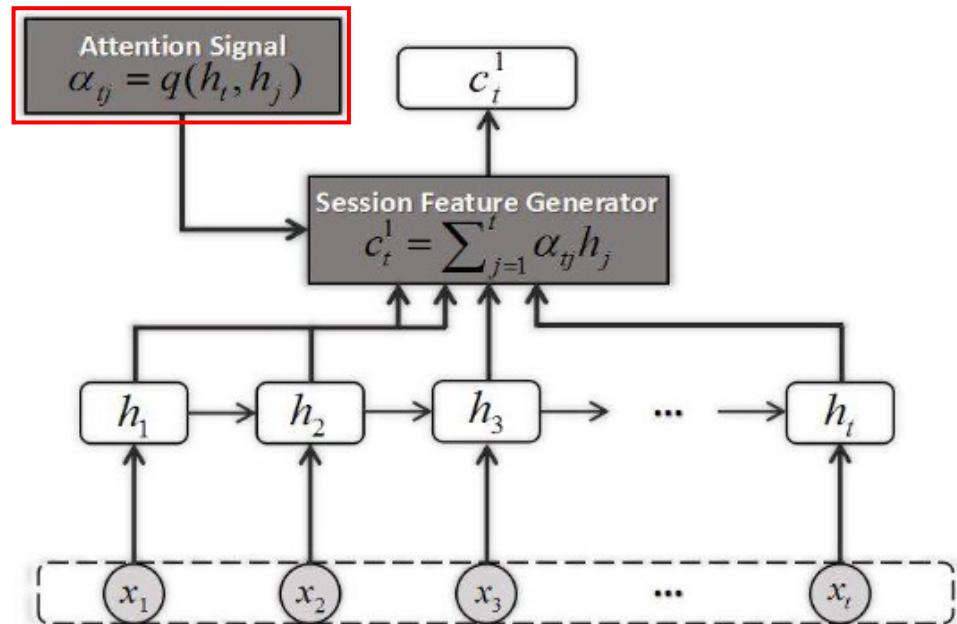
$$c_t^1 = \sum_{j=1}^t \alpha_{tj} h_j$$

$$\alpha_{tj} = q(h_t, h_j)$$

hidden states의 함수

인풋 시퀀스의 어느 부분을 강조할지 결정

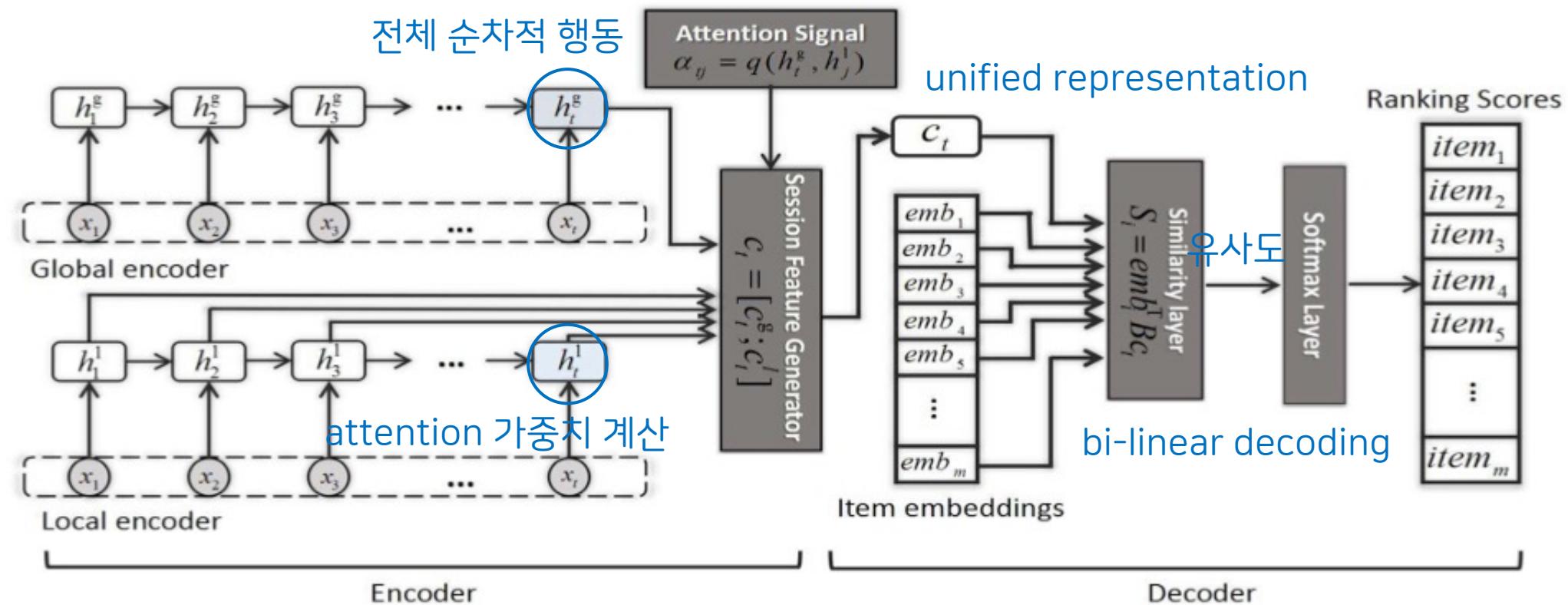
$$q(h_t, h_j) = v^T \sigma(A_1 h_t + A_2 h_j) \quad h_t \text{와 } h_j \text{의 유사도 계산}$$



Unit 03 | RNN-based Model

Neural Attentive Session-based Recommendation

NARM



Unit 03 | RNN-based Model

Neural Attentive Session-based Recommendation

의의

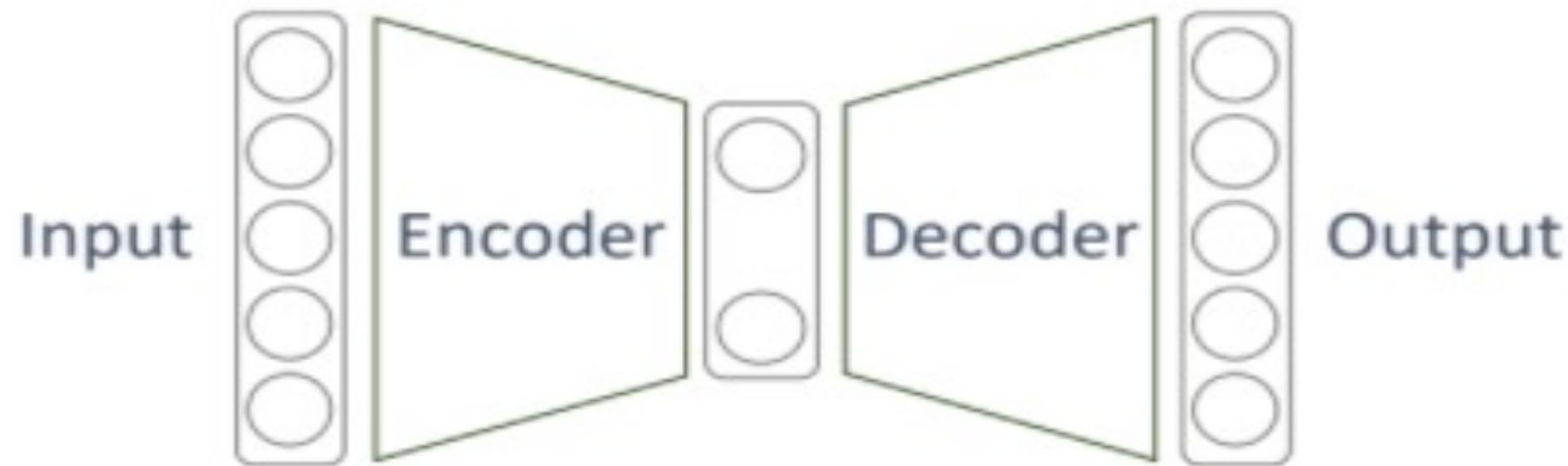
- 현재 상태의 유저의 순차적 행동과 주요 목적을 함께 고려 (local information + global information)
- bi-linear decoder을 통해 연산량을 줄이고 성능 향상
- attention mechanism 적용
- 긴 세션에서 더 좋은 성능

04 Autoencoder-based Model

Unit 04 | Autoencoder-based Model

AutoEncoder 란?

- Input과 output이 똑같은 형태의 모델
- 코드를 압축하는 Encoder와 압축된 코드를 해석하는 Decoder로 이루어짐



Unit 04 | Autoencoder-based Model

AutoEncoder의 4가지 키워드

- Unsupervised Learning -> 오직 input Data 만을 사용하기 때문에!
- Manifold Learning -> Encoder가 차원 축소의 역할을 수행하기 때문에!
- Generative Model Learning -> Decoder가 input Data로 복원하기 때문에!
- Maximum Likelihood Density Estimation -> 학습 방법이 MLE와 동일하기 때문에!

Unit 04 | Autoencoder-based Model

Encoder가 차원을 축소해준다고? -> Latent Space를 찾을 수 있지 않을까?

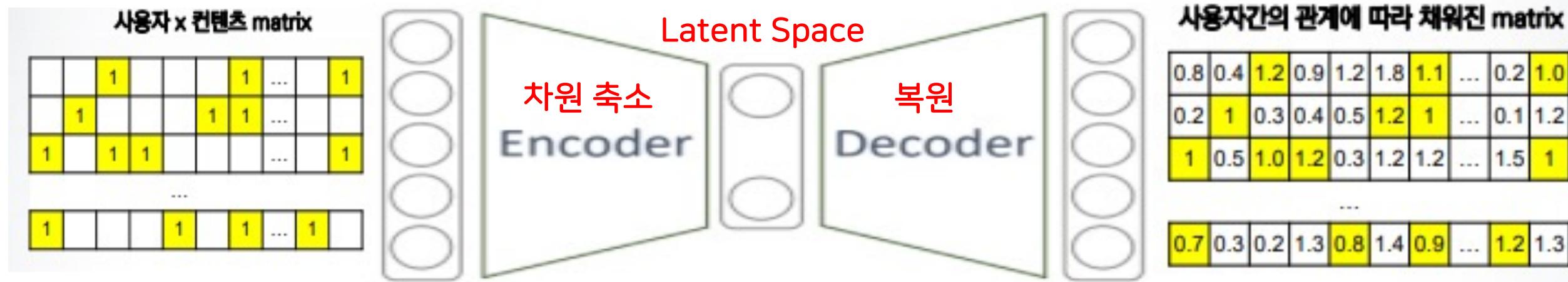
Decoder가 input Data로 복원해준다고? -> Latent Model과 유사한 것 같은데?

그러면 AutoEncoder를 추천시스템에 활용해보자!

Unit 04 | Autoencoder-based Model

AutoEncoder를 활용한 추천시스템

Autoencoder의 차원 축소-복구 과정을 이용한 컨텐츠 추천



- 비선형 활성 함수를 활용해 사용자 간의 관계를 모델링하기 때문에 성능이 좋음
- 단순 Matrix Factorization에 비해 sparse matrix에서 좋은 성능을 보임
- 평점 추천, 순위 추천, 클릭 여부, 시청 여부 등에 모두 활용 가능

Unit 04 | Autoencoder-based Model

AutoEncoder의 발전된 형태도 사용할 수 있지 않을까?

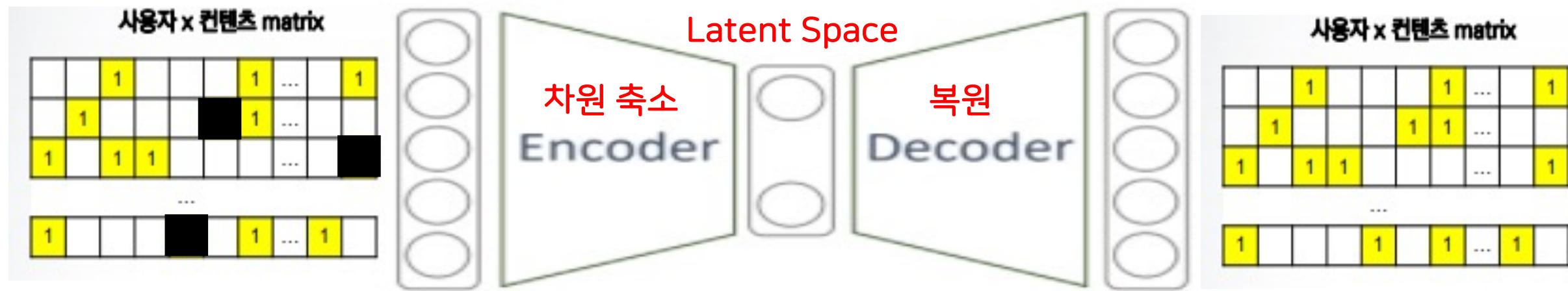
Unit 04 | Autoencoder-based Model

물론 사용 가능하며 성능도 더 우수!

Unit 04 | Autoencoder-based Model

Denoising AutoEncoder 를 활용한 추천시스템

noise가 추가된 Input data로 noise가 없는 input data로 복원시키는 방식

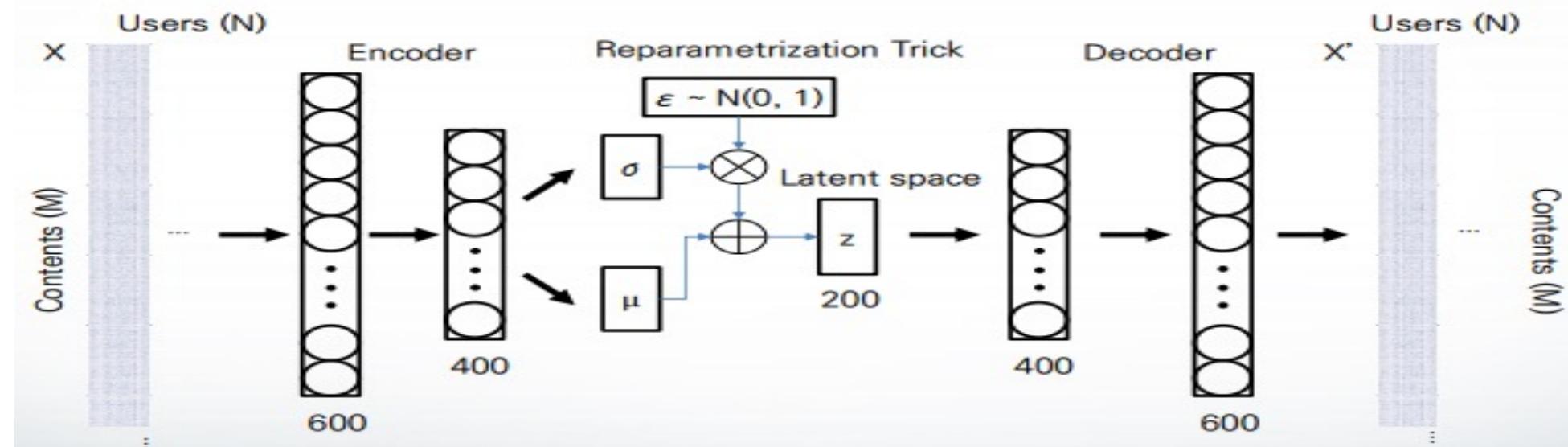


- 일부 데이터에 noise가 들어가기 때문에 조금 더 강건한 형태의 모델을 얻을 수 있음
- 실제로 기본 AE보다 더 우수한 성능을 보임

Unit 04 | Autoencoder-based Model

Variational AutoEncoder 를 활용한 추천시스템

축소된 차원으로 Z를 샘플링 하여 Z를 정규 분포에 가깝게 만드는 방식



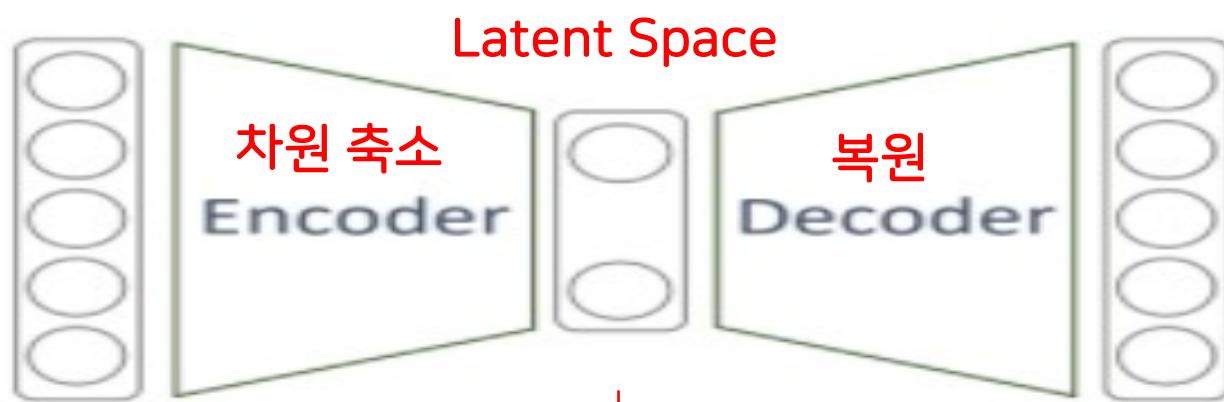
- Latent Space가 다루기 쉬워지고 컨트롤할 수 있다.
- Latent Space가 정규 분포와 유사해지기 때문에 Sparse한 부분이 줄어들어 조금 더 유용한 Space를 얻을 수 있다.
- 실제로 기본 AE보다 더 우수한 성능을 보임

Unit 04 | Autoencoder-based Model

Latent Space를 활용한 사례

전처리된 컨텐츠 matrix

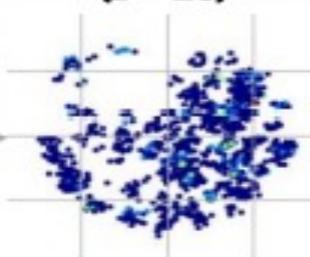
컨텐츠 메타정보
컨텐츠 이미지
컨텐츠 스크립트
.....



복원된 컨텐츠 matrix

컨텐츠 메타정보
컨텐츠 이미지
컨텐츠 스크립트
.....

컨텐츠 벡터 공간
($z = 20$)



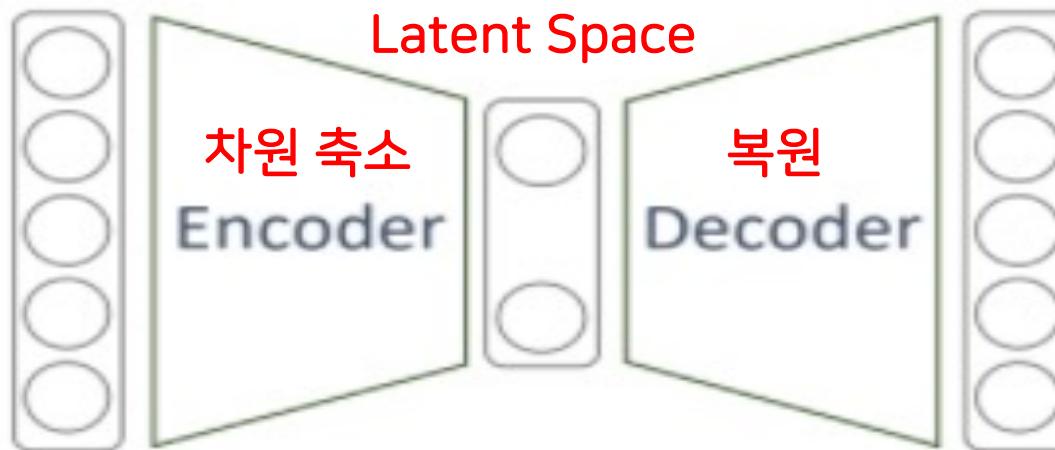
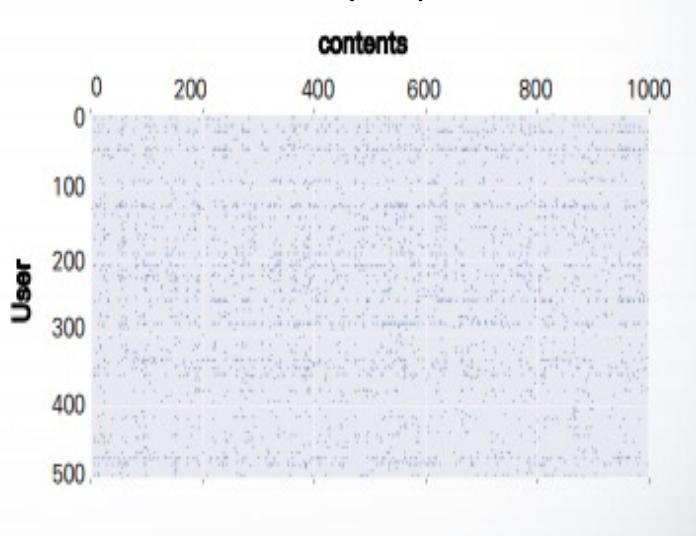
컨텐츠 ID	벡터값 (20차원)			
	1	2	3	...
13207	1.62	0.24	-1.17	...
12328	1.19	-0.36	0.28	...
40222	0.65	-0.79	0.36	...
....

컨텐츠 벡터 공간을 Cosine similarity로
거리를 측정하여
컨텐츠 간의 유사도를 기준으로 추천

Unit 04 | Autoencoder-based Model

Matrix completion을 활용한 사례

사용자의 컨텐츠 시청 정보(1 / 0)



사용자마다 컨텐츠 시청 확률 (출력)									
8.5 (1)	3.4 (4)	-∞	6.2 (3)	7.5 (2)	1.5 (5)	-∞	...	2.2 (6)	-∞

사용자마다 컨텐츠 시청 확률이
높은 순서대로 컨텐츠를 추천

Unit 04 | Autoencoder-based Model

이 밖에도 다양한 방식으로 활용이 가능!
(사용자의 평점, 영화의 다양한 feature 등)

Unit 04 | Autoencoder-based Model

추천 시스템 모델들은 각 모델마다 장단점이 존재!
따라서 상황에 맞게 모델을 활용하는 것이 좋다!
즉, 모델을 1개만 활용하면 안된다!

참고자료

- <https://developers.google.com/machine-learning/recommendation?hl=ko>
- <https://tkdguq05.github.io/2021/03/07/apriori-FP/>
- https://dnndnjs.github.io/recomm/2019/08/15/recommendation_system_basics/
- <https://leehyejin91.github.io/post-ncf/>
- https://www.youtube.com/watch?v=1fhNX_JFaow
- <https://www.youtube.com/watch?v=h6vePourB7E&list=LL&index=1>
- <https://yjo.tistory.com/18>
- Zhang, S., Yao, L., Sun, A., & Tay, Y. (2017). Deep Learning based Recommender System: A Survey and New Perspectives. *ACM*.
- He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. (2017). Neural Collaborative Filtering. *WWW*.
- Cheng, H., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., … Shah, H. (2016). Wide & Deep Learning for Recommender Systems. *Google Inc.*
- Covington, P., Adams, J., & Sargin, E. (2016). Deep Neural Networks for YouTube Recommendations. *RecSys*.
- Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D. (2016). Session-based Recommendations with Recurrent Neural Networks. *ICLR*.
- Li, J., Ren, P., Chen, Z., Ren, Z., Lain, T., Ma, J. (2017). Neural Attentive Session-based Recommendation. *ACM*.

Q & A

들어주셔서 감사합니다.