

# Introduction to Sequential Recommender Systems

2020. 10. 23

Data Mining & Quality Analytics Lab.

발표자 : 김서연

# 발표자 소개



- 김서연 (Seo-Yeon Kim)
  - 고려대학교 산업경영공학부 재학 중
  - Data Mining & Quality Analytics Lab
  - 석사과정 (2020.03 ~)
- 관심 연구 분야
  - Machine Learning Algorithms
  - Recommender System
- E-mail : joanne2kim@korea.ac.kr

# Contents

1. Introduction to Recommender System
2. Deep Learning based Recommender System
  - Neural Collaborative Filtering
3. Sequential Recommender System
  - Feature-level Deeper Self-Attention Network for Sequential Recommendation

# Introduction to Recommender System

추천시스템이란?

## 추천 시스템

(Recommender System)



NETFLIX



# Introduction to Recommender System

추천시스템이란?

2020년 9월 25일  
**Stochastic Gradient Descent**  
Data Mining & Quality Analytics Lab  
발표자: 배진수

Stochastic Gradient Descent  
조회수 397회 · 2020. 9. 25.

김성범[교수 / 산업경영공학과]  
구독자 4.81천명

다음 동영상 자동재생

- [핵심 머신러닝] 의사결정나무 모델 1 (모질개요, 예측나무)  
김성범[교수 / 산업경영공학과]  
조회수 6.1천회 · 1년 전
- 코딩 1시간만에 배우기 - 파이썬 (ft. 실리온벨리 엔지니어)  
Teccebol Wonie  
조회수 43만회 · 9개월 전
- 코딩 1시간만에 배우기  
초단기 속성 파이썬 1:31:12
- Time Series Regression - Part 1  
김성범[교수 / 산업경영공학과]  
조회수 529회 · 2주 전
- Stochastic Gradient Descent, Clearly Explained!!!  
StatQuest with Josh Starmer  
조회수 13만회 · 1년 전
- Stochastic Gradient Descent...  
...Clearly Explained!!!
- [핵심 확률/통계] 분산, 공분산, 상관관계  
김성범[교수 / 산업경영공학과]  
조회수 828회 · 3개월 전
- 전용종어 없이 듣는 딥러닝의 모든 것 (Feat. 인공지능망, 퍼셉...  
코딩하는거니  
조회수 5.4만회 · 9개월 전
- 경사하강법 (Gradient Descent)의 기본 개념을 쉽게 알려드립니다...  
테디노트  
조회수 1천회 · 4개월 전
- 경사하강법을 수학으로 쉽게 이해하기
- GPT-3 퍼러다임을 바꿀 미친 성능의 인공지능 등장 및 활용 사...  
조코딩 JoCoding  
조회수 9만회 · 1개월 전

# Introduction to Recommender System

추천시스템이란?

2020년 9월 25일

Stochastic Gradient Descent  
Data Mining & Quality Analytics Lab

**사용자에 대하여 이해하도록 도와주는 정보**

ex) 시청한 동영상, 구독 정보, 등

다음 동영상

[핵심 머신러닝] 의사결정나무 모델 1 (모질개요, 예측나무)  
김성범[교수 / 산업경영공학과]  
조회수 6.1천회 · 1년 전

코딩 1시간만에 배우기 - 파이썬 (ft. 실리콘밸리 엔지니어)  
Tecbol Wonie  
조회수 43만회 · 9개월 전

코딩 1시간만에 배우기  
초단기 속성 파이썬 1:31:12

Time Series Regression - Part 1  
김성범[교수 / 산업경영공학과]

**사용자가 관심을 보일 다른 아이템**

ex) 유사한 다른 동영상

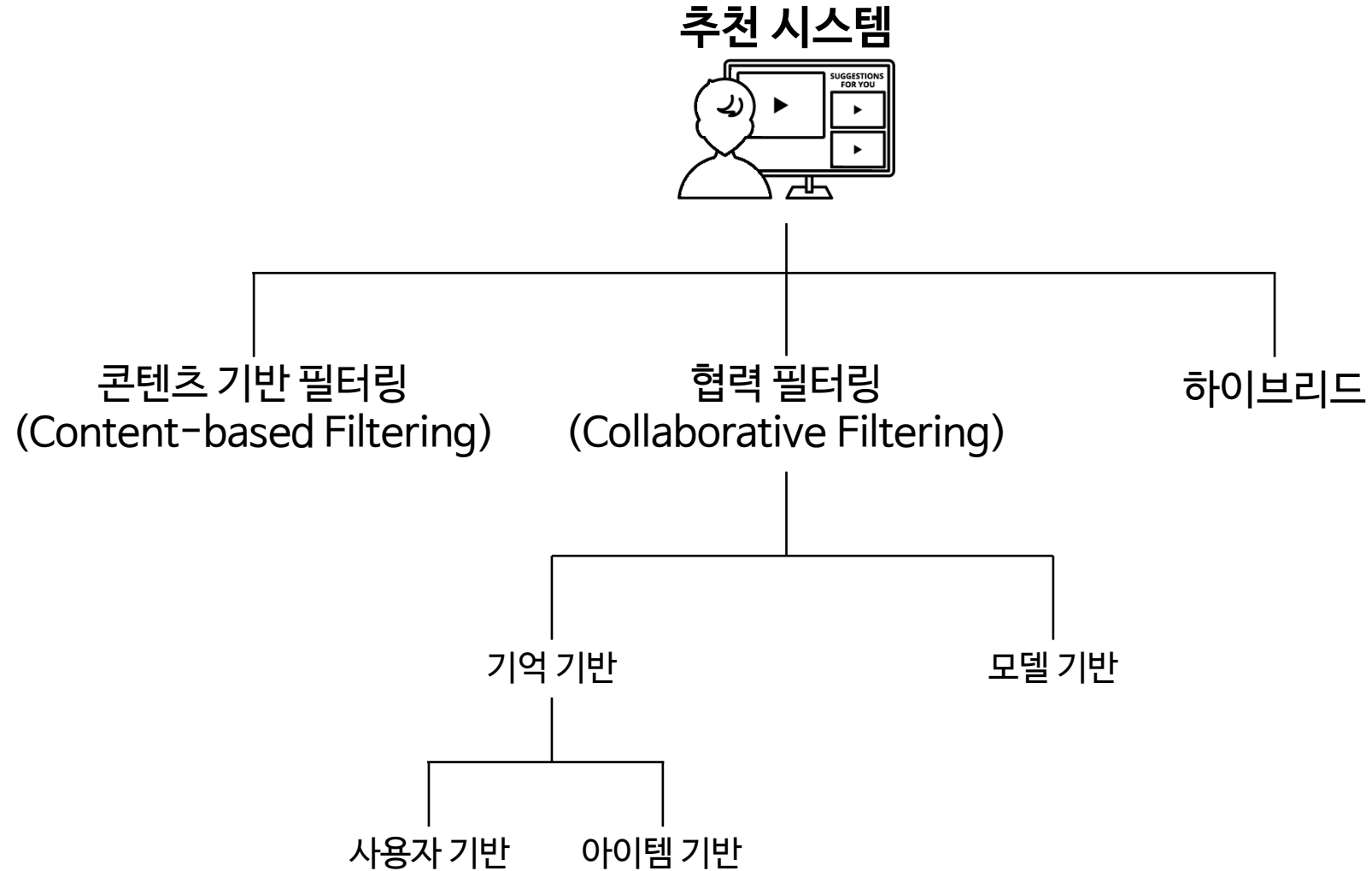
답리닝 (Feat. 인공지능)  
조회수 5.4만회 · 9개월 전

경사하강법을 수학으로 쉽게 이해하기  
테디노트  
조회수 1천회 · 4개월 전

GPT-3 퍼러다임을 바꿀 미친 성능의 인공지능 등장 및 활용 사...  
조코딩 JoCoding  
조회수 9만회 · 1개월 전

# Introduction to Recommender System

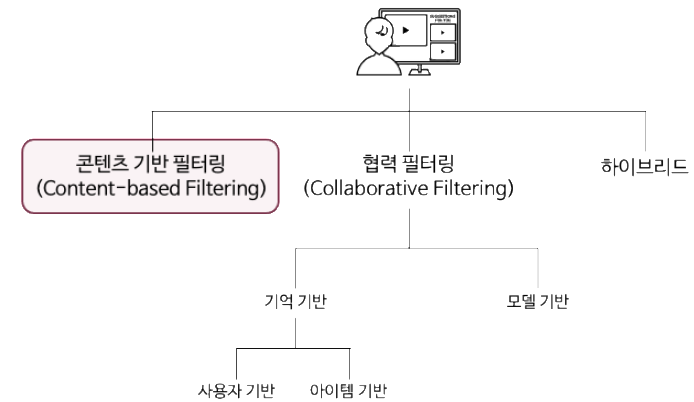
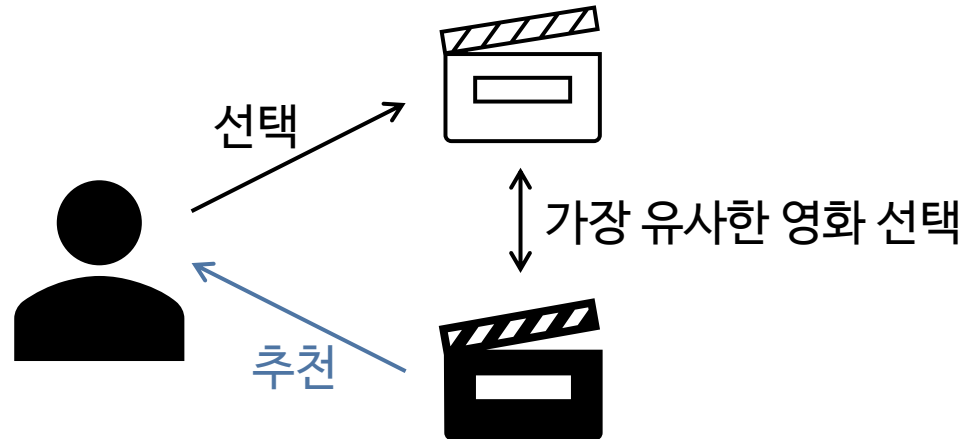
추천시스템이란?



# Introduction to Recommender System

## 콘텐츠 기반 필터링 (Content-based Filtering)

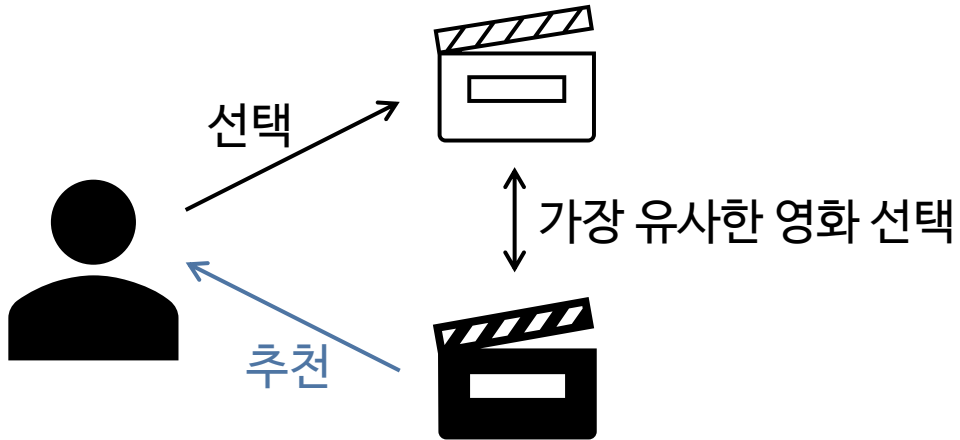
- **콘텐츠** 기반으로 분석하여 추천해주는 방식
- 사용자가 관심분야에 대해서 직접 입력한 정보나, 구매 내역, 평점 등을 기반으로 선호 아이템 파악
- 선호하는 아이템과 가장 유사한 다른 아이템 추천





# Introduction to Recommender System

## 콘텐츠 기반 필터링 (Content-based Filtering)

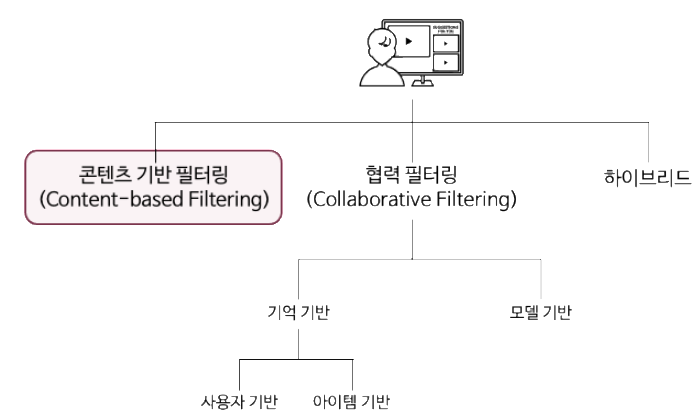


장점 :

- 아이템의 속성(콘텐츠)을 기반으로 추천하기 때문에 이전에 선택 이력이 없는 새로운 아이템도 추천 가능

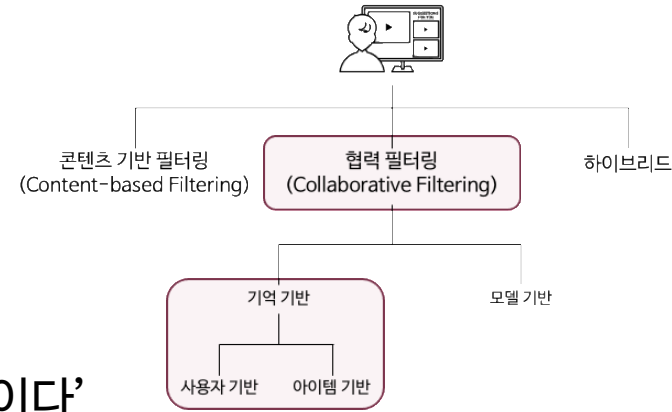
단점 :

- 해당 고객에 대한 데이터가 부족한 경우 추천 성능 보장이 어려움
- 과도한 특수화 (Over Specialization)
  - 이전에 구매 및 선택한 아이템과 비슷한 제품만 추천하는 경향



# Introduction to Recommender System

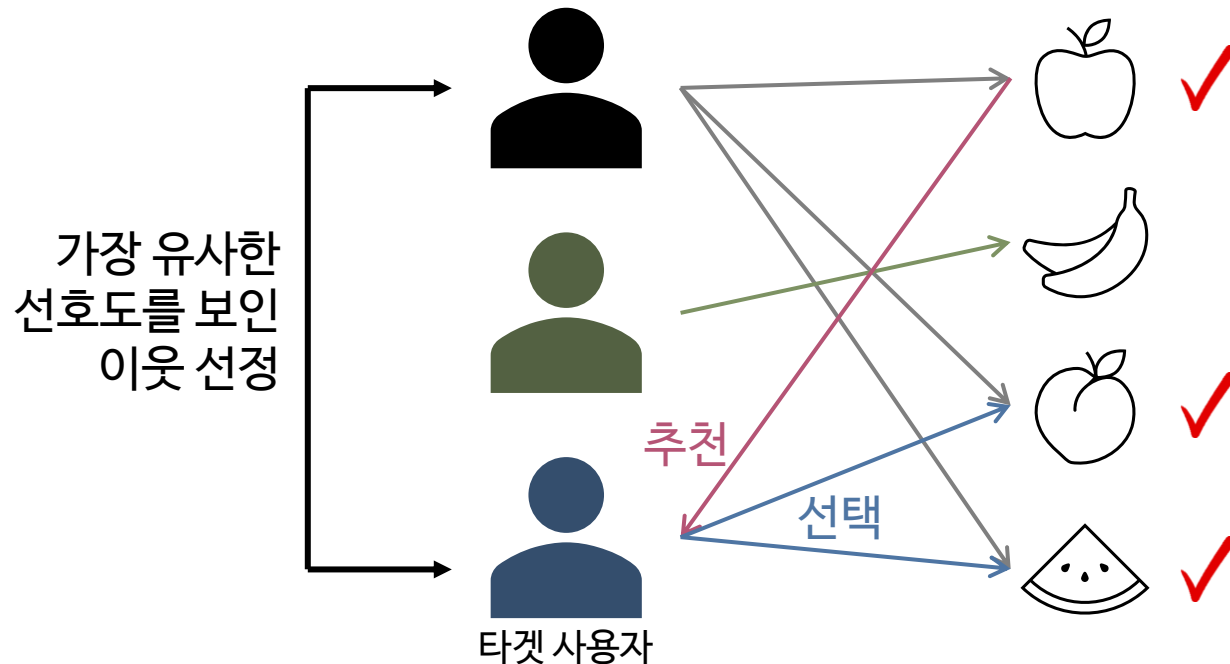
## 협력 필터링 (Collaborative Filtering) - 기억 기반



- ‘특정 아이템에 대하여 선호도가 유사한 고객들은

다른 아이템에 대해서도 비슷한 선호도를 보일 것이다’

- 사용자 기반 협력 필터링 : 사용자 간의 유사도를 측정하여 유사도 높은 이웃이 선택한 아이템 중에서 추천



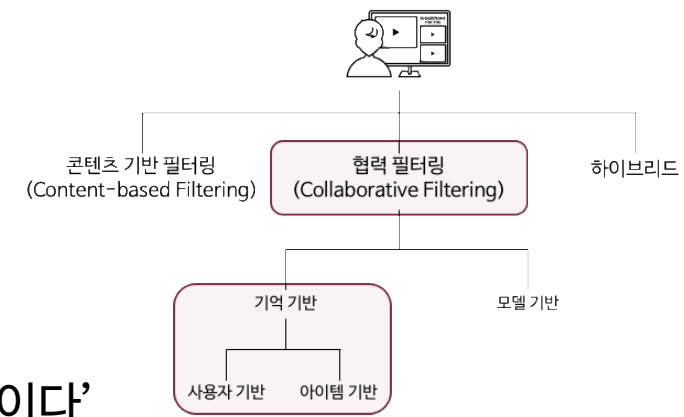
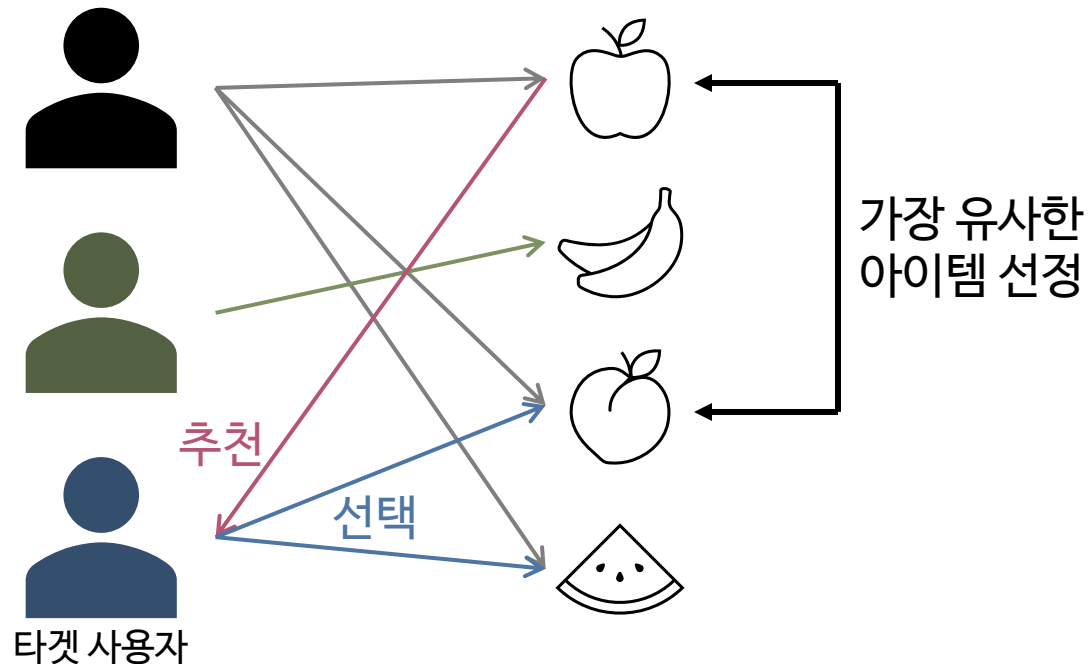
# Introduction to Recommender System

## 협력 필터링 (Collaborative Filtering) - 기억 기반

- ‘특정 아이템에 대하여 선호도가 유사한 고객들은

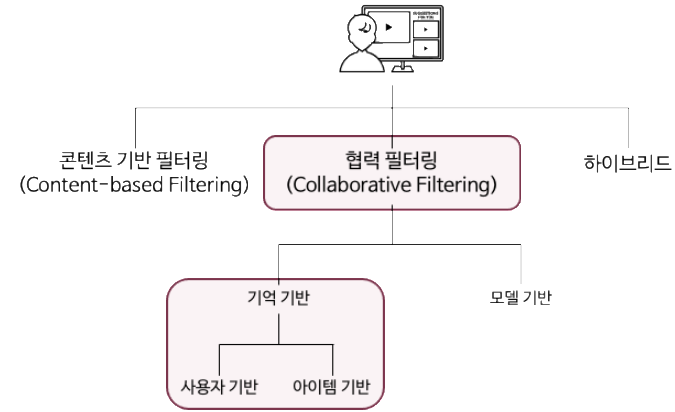
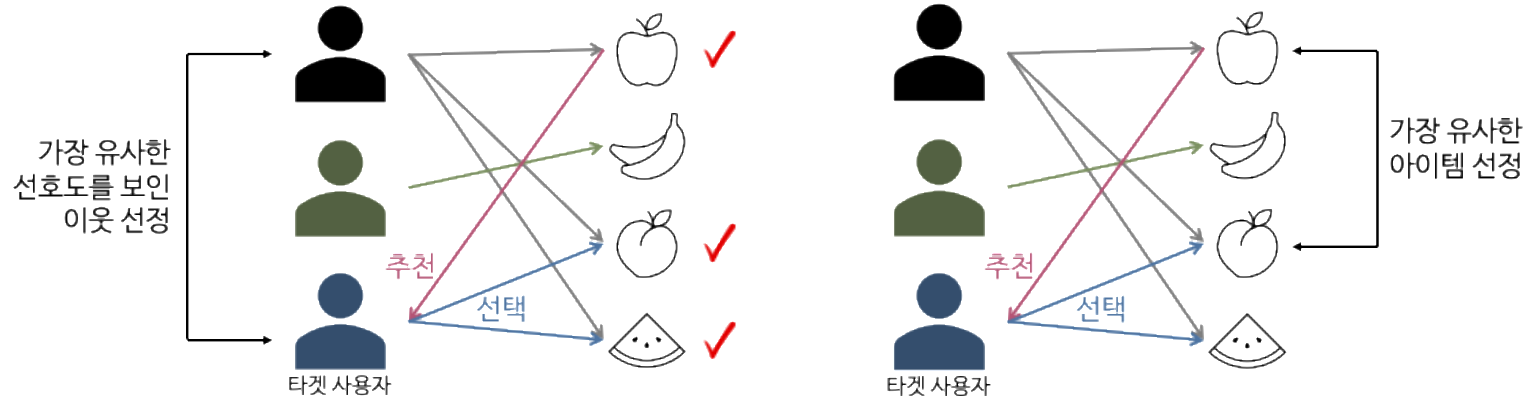
다른 아이템에 대해서도 비슷한 선호도를 보일 것이다’

- **아이템 기반 협력 필터링** : 아이템 간 유사도를 측정하여 유사도 높은 아이템을 추천



# Introduction to Recommender System

## 협력 필터링 (Collaborative Filtering) - 기억 기반

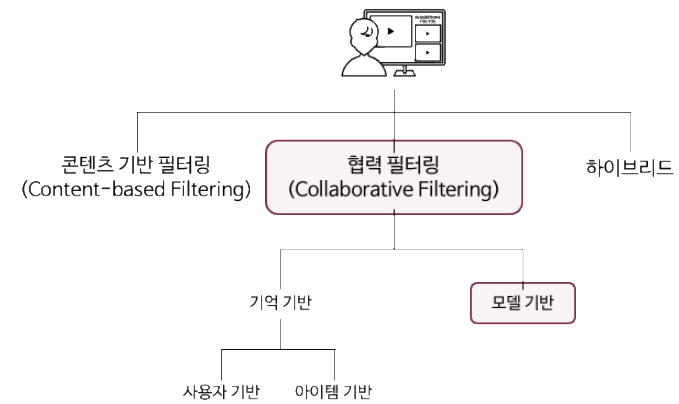
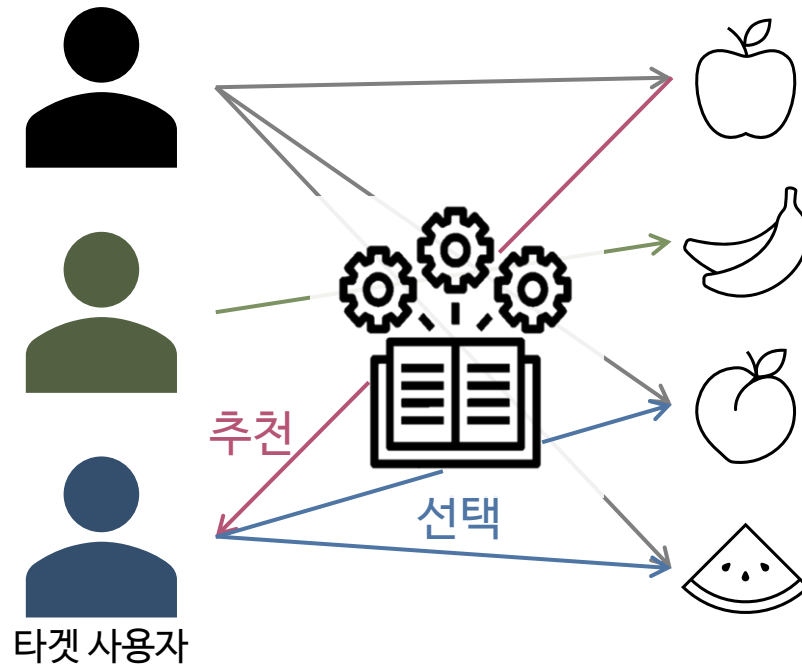


- Cold Start Problem : 한 사용자에게 대한 충분한 데이터가 부족한 경우 선호도 예측 불가능
- First Rater : 새로운 아이템이 등장하여 평점 점수가 부족한 경우 추천 불가능
- Grey Sheep Problem : 일관성이 없는 의견을 가진 사용자들의 데이터는 추천에 혼란을 줌
- Shilling Attack : 악의적으로 평가 점수를 긍정/부정으로 입력하는 경우 추천에 방해가 됨

# Introduction to Recommender System

## 협력 필터링 (Collaborative Filtering) - 모델 기반

- 데이터에 내재 되어있는 복잡한 패턴을 발견하도록 다양한 모델을 활용한 기법
- 실제 데이터에 적용했을 때 성능이 우수함

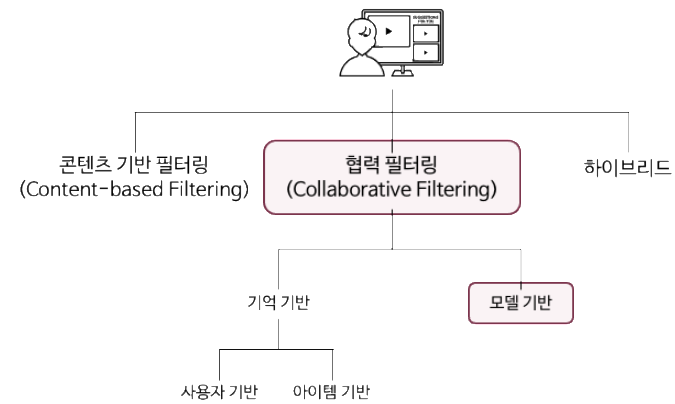


# Introduction to Recommender System

## 협력 필터링 (Collaborative Filtering) - 모델 기반

- Matrix Factorization

- ‘사용자와 아이템 사이에는 사용자의 행동과 평점에 영향을 끼치는 잠재된 특성이 있을 것이다’
- 크기가 크며 복잡한 데이터로도 쉽고 빠르게 분석 진행 가능



사용자-아이템 평점 행렬

1		3		4	
	5		4	2	
2		1			5
$m$ 2	3		4	1	
3	1	2	3		4
1		4	5	2	3
	1		2	4	3

$n$

사용자의 잠재 정보 행렬


$d$

=

×

아이템의 잠재 정보 행렬


$n$



# Deep Learning based Recommender System

딥러닝을 활용한 추천시스템의 등장

- 데이터 내에 존재하는 여러 추상적이며 복잡한 내용을 학습하여 핵심을 알아내는 **딥러닝**의 발전
- 추천시스템의 관점에서, **사용자의 복잡한 정보를 학습**하여 핵심을 알아낼 수 있다면?



머신러닝



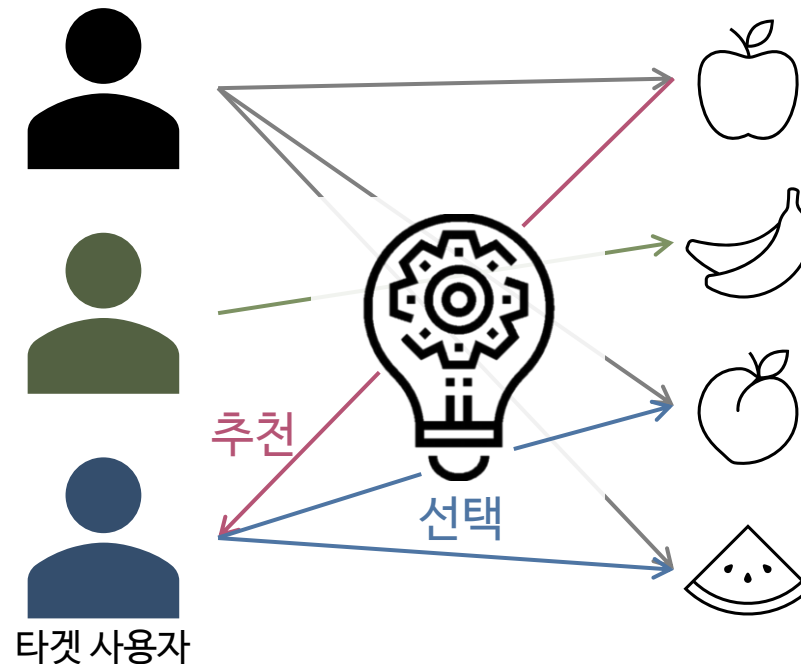
딥러닝



# Deep Learning based Recommender System

딥러닝을 활용한 추천시스템의 등장

- 현실의 데이터가 더 많은 사용자, 더 많은 아이템에 대한 정보를 담게 되면서 기존의 기본적인 방식보다 데이터를 잘 이해하고, 이에 더 좋은 추천을 해주는 알고리즘의 필요성 대두
- 기존 데이터 뿐만 아니라 이미지, 텍스트 등 다양한 데이터도 함께 활용



# Deep Learning based Recommender System

## Neural Collaborative Filtering

- International World Wide Web Conference에서 2017년 발표된 논문
- 2020년 10월 14일 기준 1597회 인용

### 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.

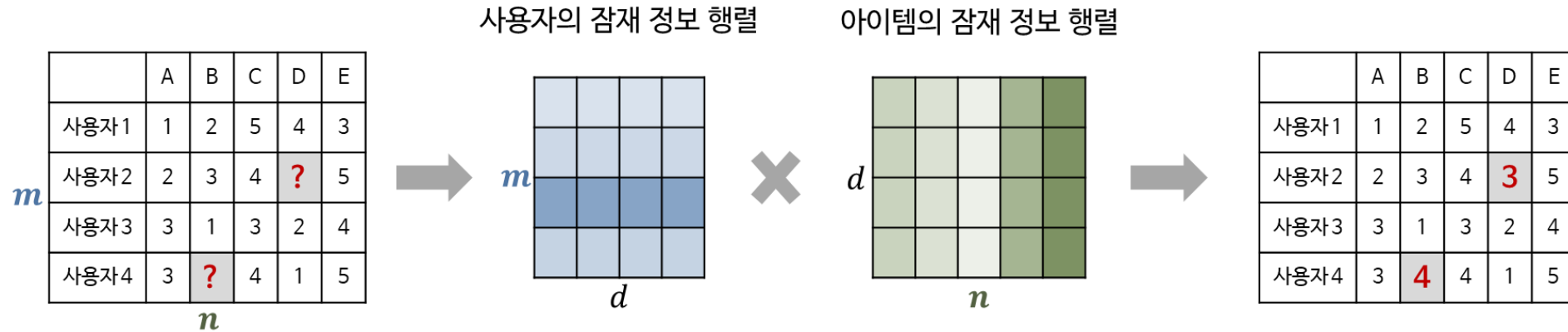
#### 1. INTRODUCTION

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<sup>1</sup>. While it seems to be just a trivial tweak for the inner product operator [14], it points to the positive effect of designing a

# Neural Collaborative Filtering

## Matrix Factorization의 단점



\*예시 : 4명의 사용자가 5개의 아이템에 대한 평점을 매긴 경우

- 사용자와 아이템에 대한 잠재 정보 행렬을 만든다는 것은 특정 2차원 공간으로 투영시켜 표현하는 것을 의미함
- 하지만, 새로운 사용자가 등장했을 때 2차원 상에서 이를 적합하게 표현하는 것의 어려움 존재

# Neural Collaborative Filtering

## Matrix Factorization의 단점

- Jaccard Coefficient를 활용하여 유사도를 계산함

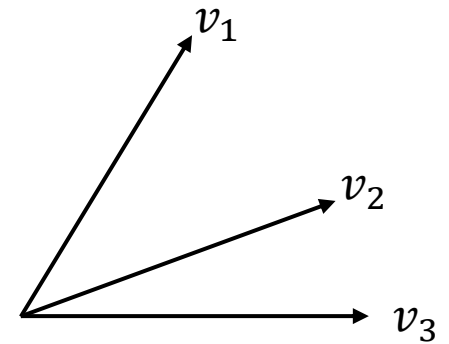
- $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$ ,  $\rightarrow s_{ij}$  : 사용자  $i$ 와 사용자  $j$ 사이의 유사도,  $v_i$  : 사용자  $i$ 의 벡터

	아이템 1	아이템 2	아이템 3	아이템 4	아이템 5
사용자 1	1	1	1	0	1
사용자 2	0	1	1	0	0
사용자 3	0	1	1	1	0

$$s_{12} = \frac{2}{4} = 0.5$$

$$s_{13} = \frac{2}{5} = 0.4$$

$$s_{23} = \frac{2}{3} = 0.666$$



\* 예시 : 아이템 구매 여부를 0과 1로 표현한 데이터

# Neural Collaborative Filtering

## Matrix Factorization의 단점

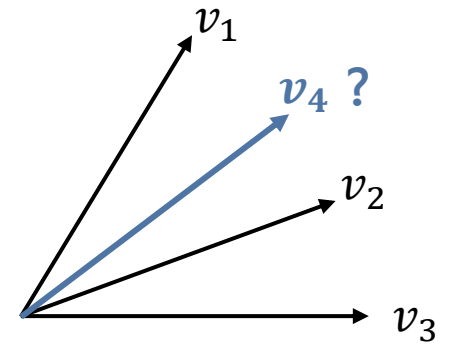
- Jaccard Coefficient ( $\frac{|A \cap B|}{|A \cup B|}$ )를 활용하여 유사도를 계산함
- 새로운 사용자가 등장하였을 때, 다른 사용자들과의 유사도 정도를 2차원으로 표현하는데 한계가 있음

	아이템 1	아이템 2	아이템 3	아이템 4	아이템 5
사용자 1	1	1	1	0	1
사용자 2	0	1	1	0	0
사용자 3	0	1	1	1	0
사용자 4	1	0	1	1	1

$$s_{41} = \frac{3}{5} = 0.6$$

$$s_{42} = \frac{1}{5} = 0.2$$

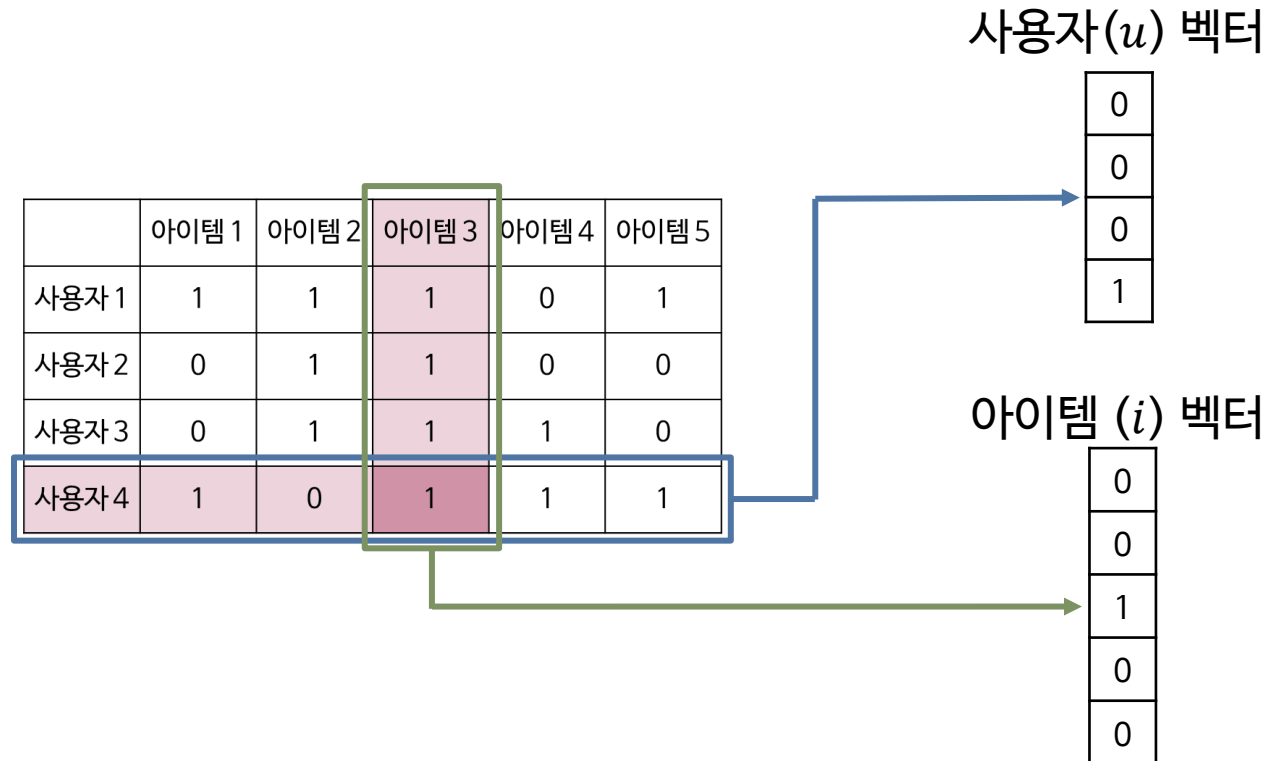
$$s_{43} = \frac{2}{5} = 0.4$$



\* 예시 : 아이템 구매 여부를 0과 1로 표현한 데이터

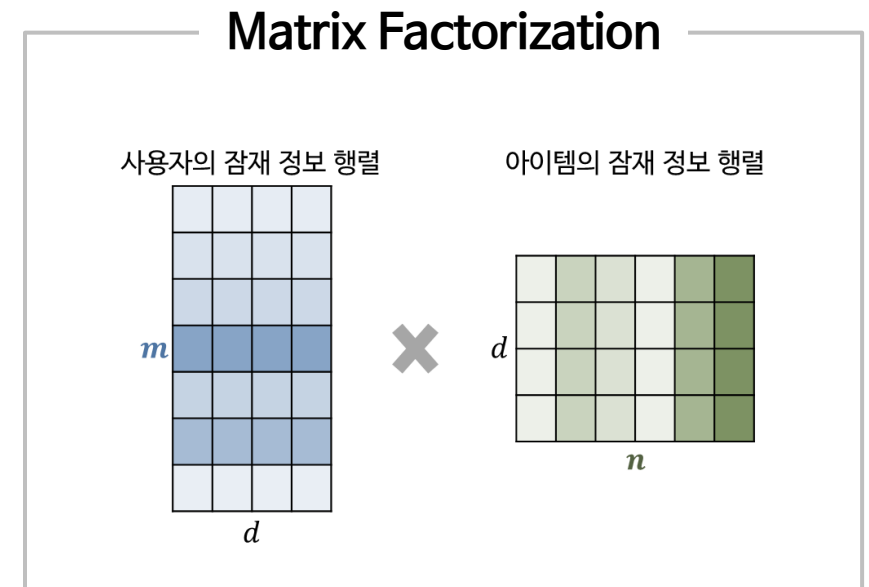
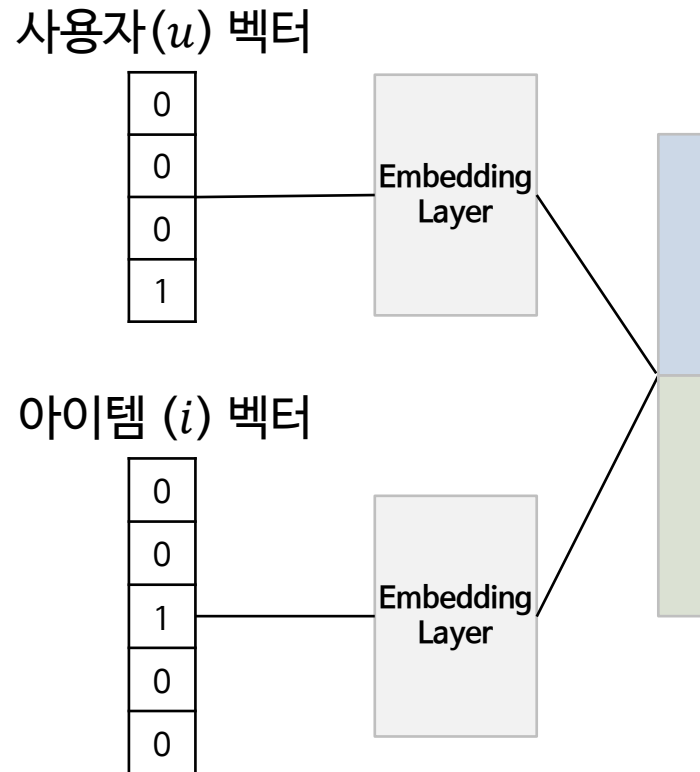
# Neural Collaborative Filtering

- 입력 데이터 : 사용자와 아이템에 대한 인풋 벡터
- 협력 필터링을 기반으로 하는 모델이기 때문에, 사용자와 아이템에 대한 정보만 인풋으로 활용



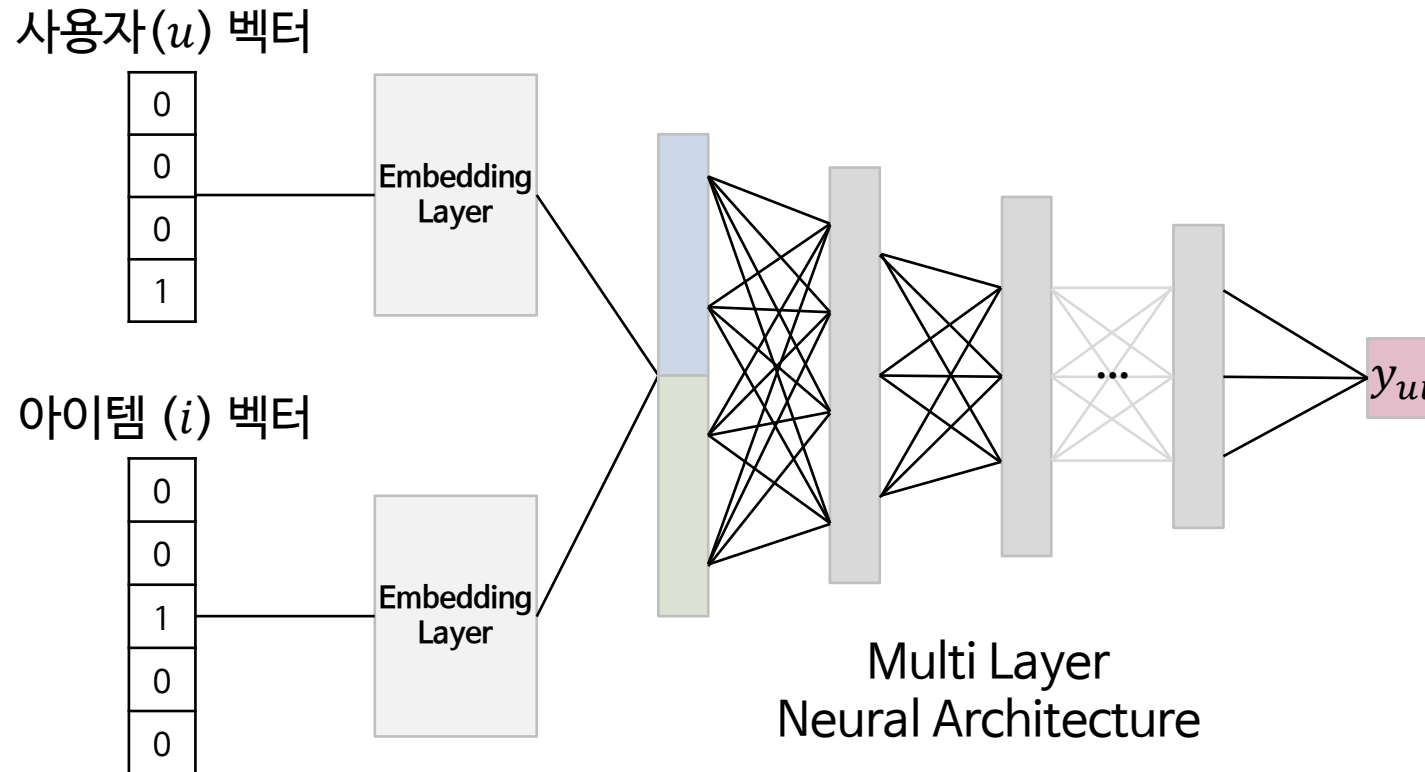
# Neural Collaborative Filtering

- 입력 데이터 : 사용자와 아이템에 대한 인풋 벡터
- 협력 필터링을 기반으로 하는 모델이기 때문에, 사용자와 아이템에 대한 정보만 인풋으로 활용



# Neural Collaborative Filtering

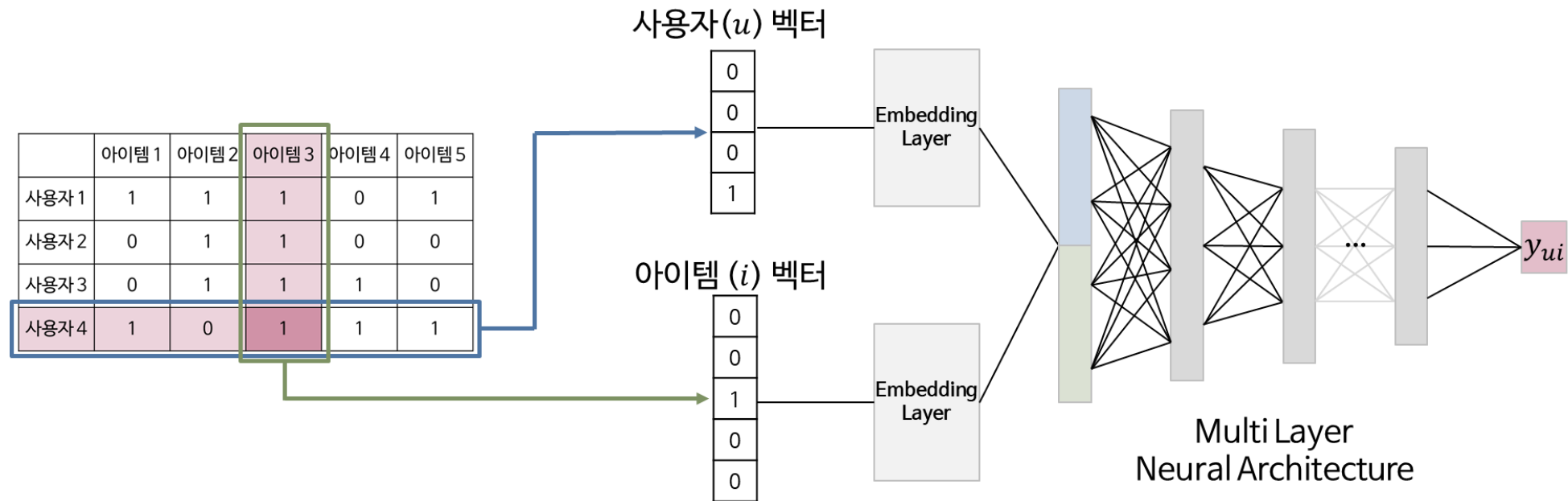
- 입력 데이터 : 사용자 4와 아이템 3에 대한 인풋 벡터
- 출력 데이터 : 사용자 4가 아이템 3을 선택한 여부 (확률값 출력)





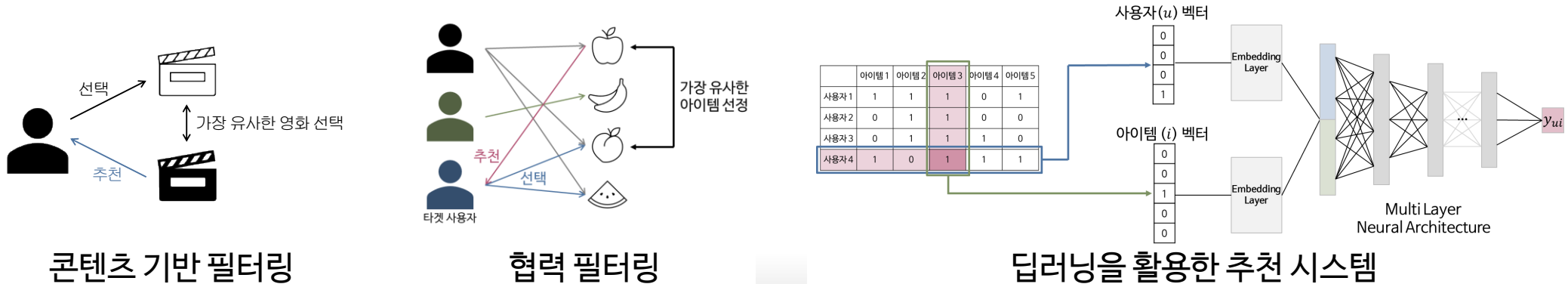
# Neural Collaborative Filtering

- 딥러닝을 추천시스템에 활용한 대표적인 사례
  - Multi Layer Perceptron + Collaborative Filtering
- 간단한 모델의 아이디어이지만, 이전 기본적인 모델을 활용하던 방법과 비교하였을 때 높은 성능을 보이고 있음



# Sequential Recommender System

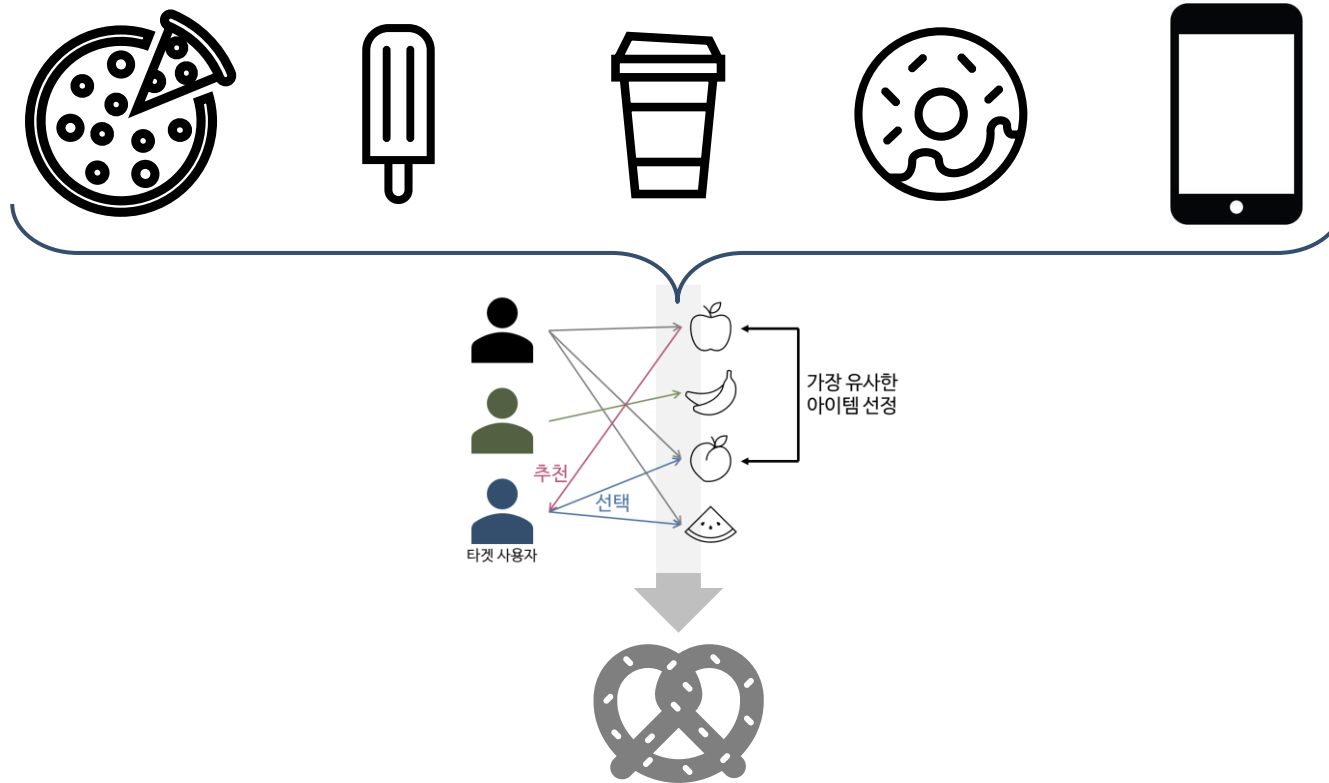
- 사용자의 과거 아이템 선택의 정보가 동일하게 중요하다는 기본 가정에서 출발한 추천 시스템 알고리즘



실제로 사용자가 선택을 할 때에는  
과거 구매 정보가 동일하게 중요할까?

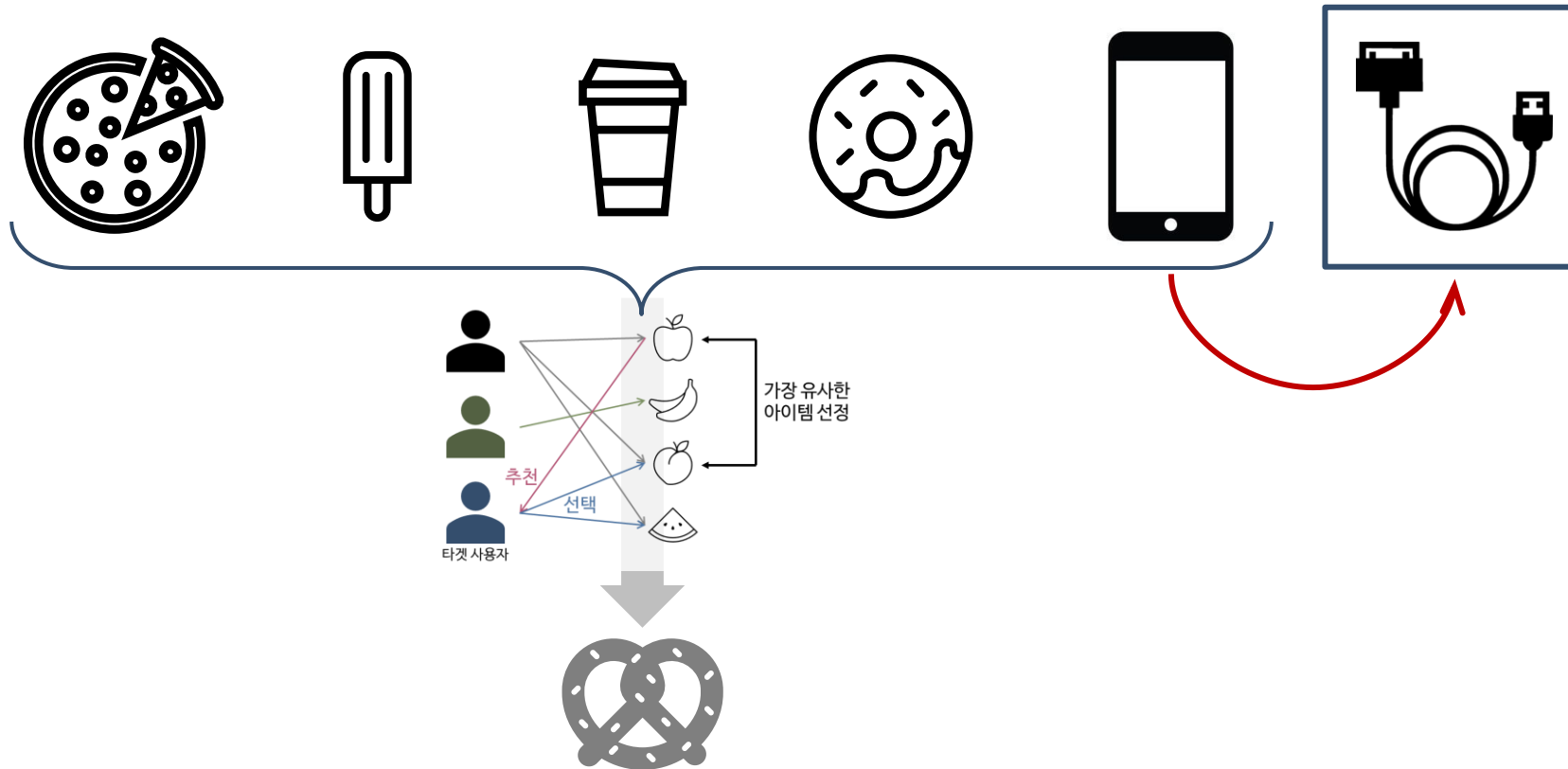
# Sequential Recommender System

- 아래의 아이템들을 선택한 사용자가 추천 받을 아이템은 무엇일까?



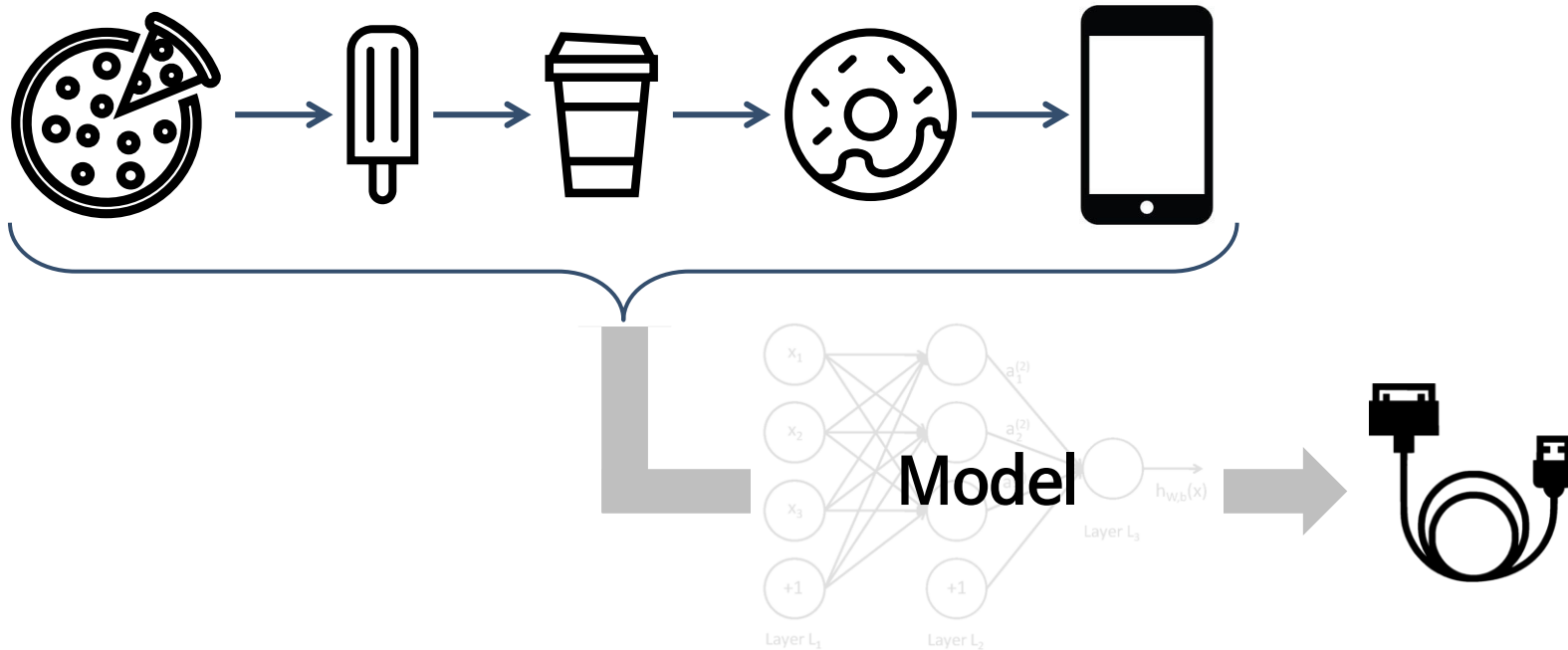
# Sequential Recommender System

- 주로 선택하던 아이템과 다른 아이템이 등장할 경우 이에 대한 충분한 설명이 부족
- 협력 필터링 등 보편적으로 사용되는 추천시스템 모델에서는 사용자가 주로 선택하던 아이템을 추천하게 됨



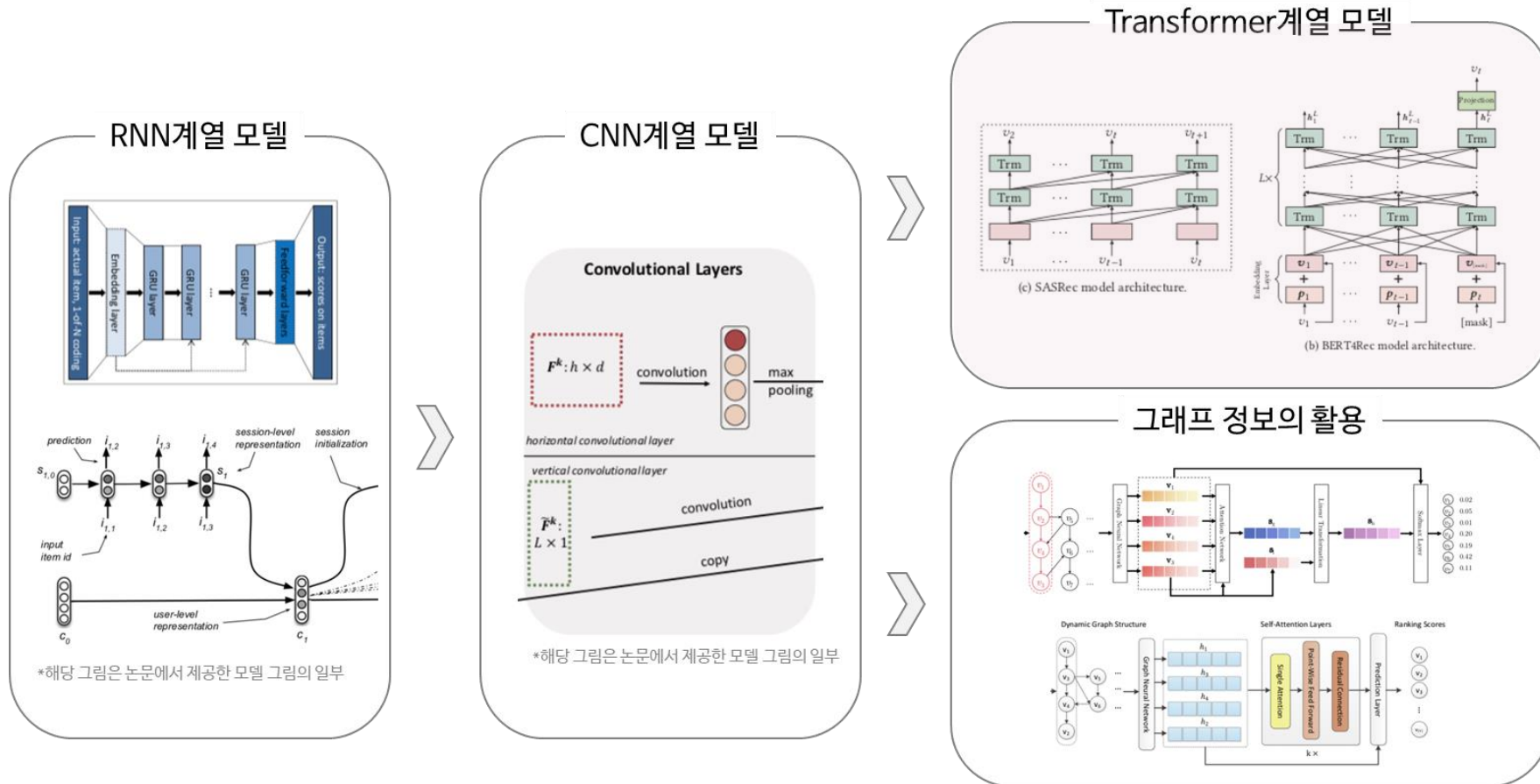
# Sequential Recommender System

- 사용자의 선호도, 관심은 끊임없이 변화하고 발전한다는 아이디어에서 시작
- 조금 더 실제 사용자의 관심사를 잘 반영하는, 변화의 패턴까지도 잡아낼 수 있는 모델을 만들어보자!
- 과거 행동에서 유의한 **순차적인 패턴**을 찾고, **최근 아이템에 더 집중**하여 이를 기반으로 가장 선호할 다음 아이템을 추천



# Sequential Recommender System

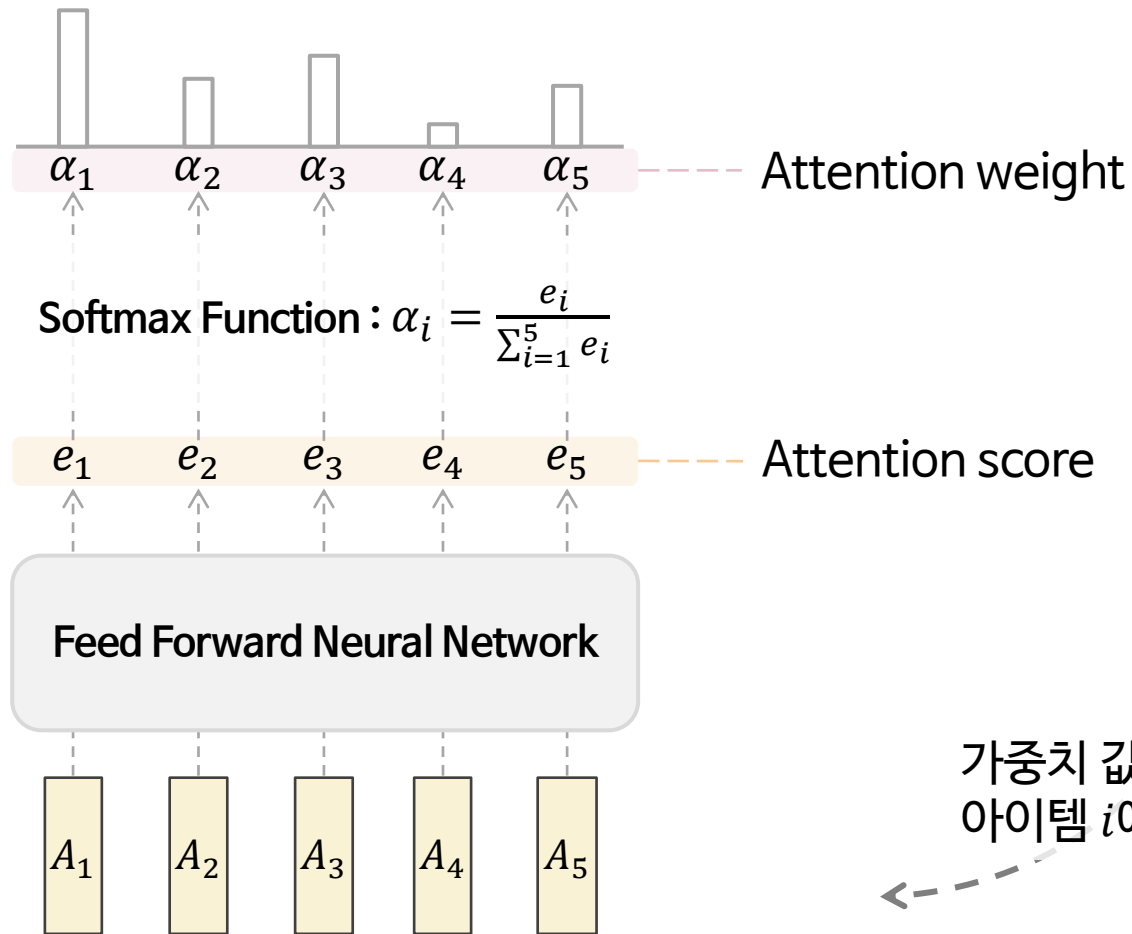
- 다양한 딥러닝 모델들을 활용한 추천시스템 방법론들이 제안됨



# Sequential Recommender System

## Vanilla Attention

- 입력된 데이터 중 어느 부분에 더 **집중**을 해야 하는지 중요도 가중합을 계산하는 방법



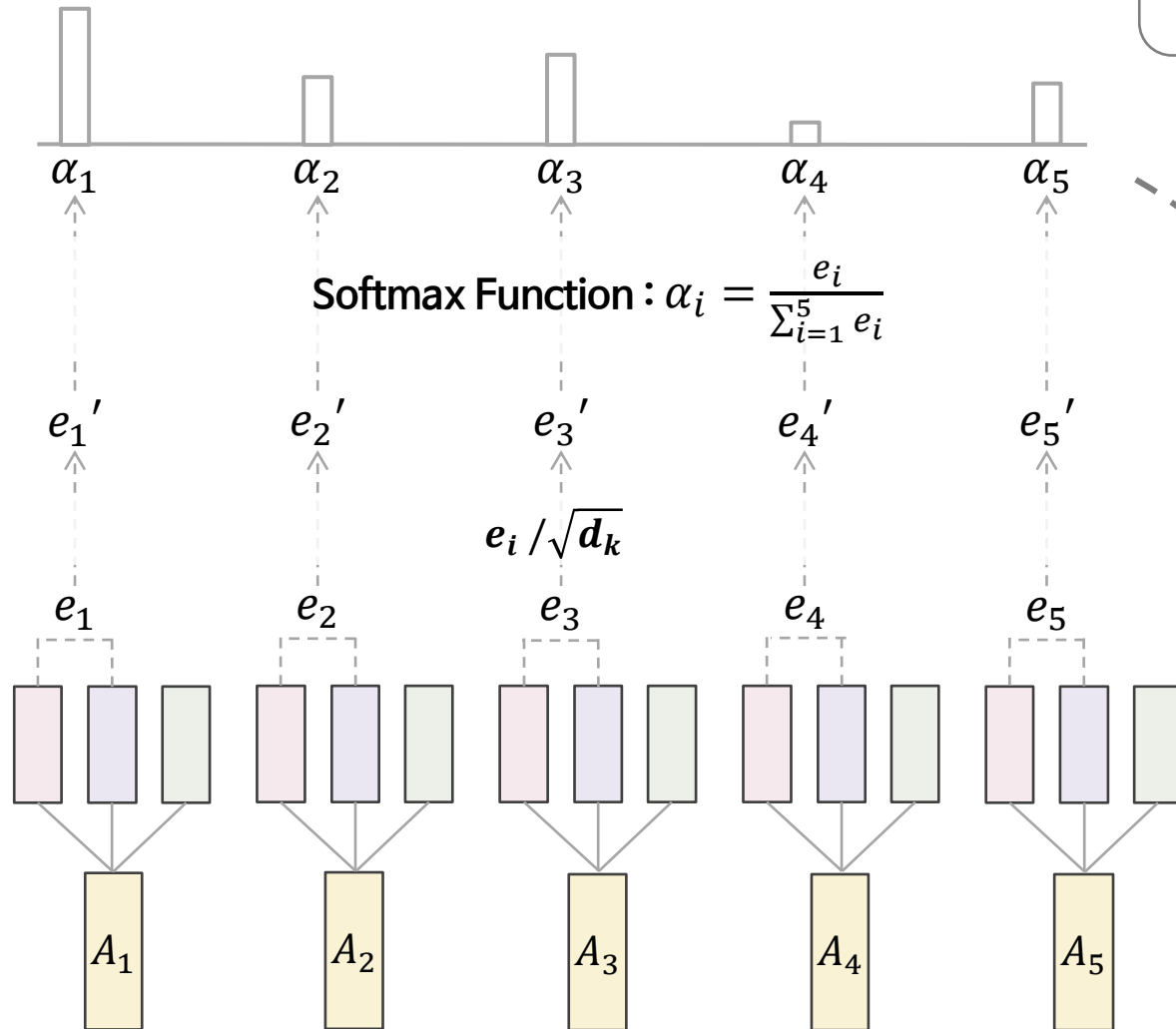
$$\alpha_i = \text{softmax}(W^f A_i + b^f)$$
$$f_i = \alpha_i A_i$$

가중치 값과의 연산을 통하여 얻은  
아이템  $i$ 에 대한 **feature representation**

# Sequential Recommender System

## Self-Attention

- Query
- Key
- Value



$$\text{Softmax Function : } \alpha_i = \frac{e_i}{\sum_{i=1}^5 e_i}$$

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V$$

Scaled dot-product Attention을 활용하여 얻는 아이템  $i$ 에 대한 feature representation

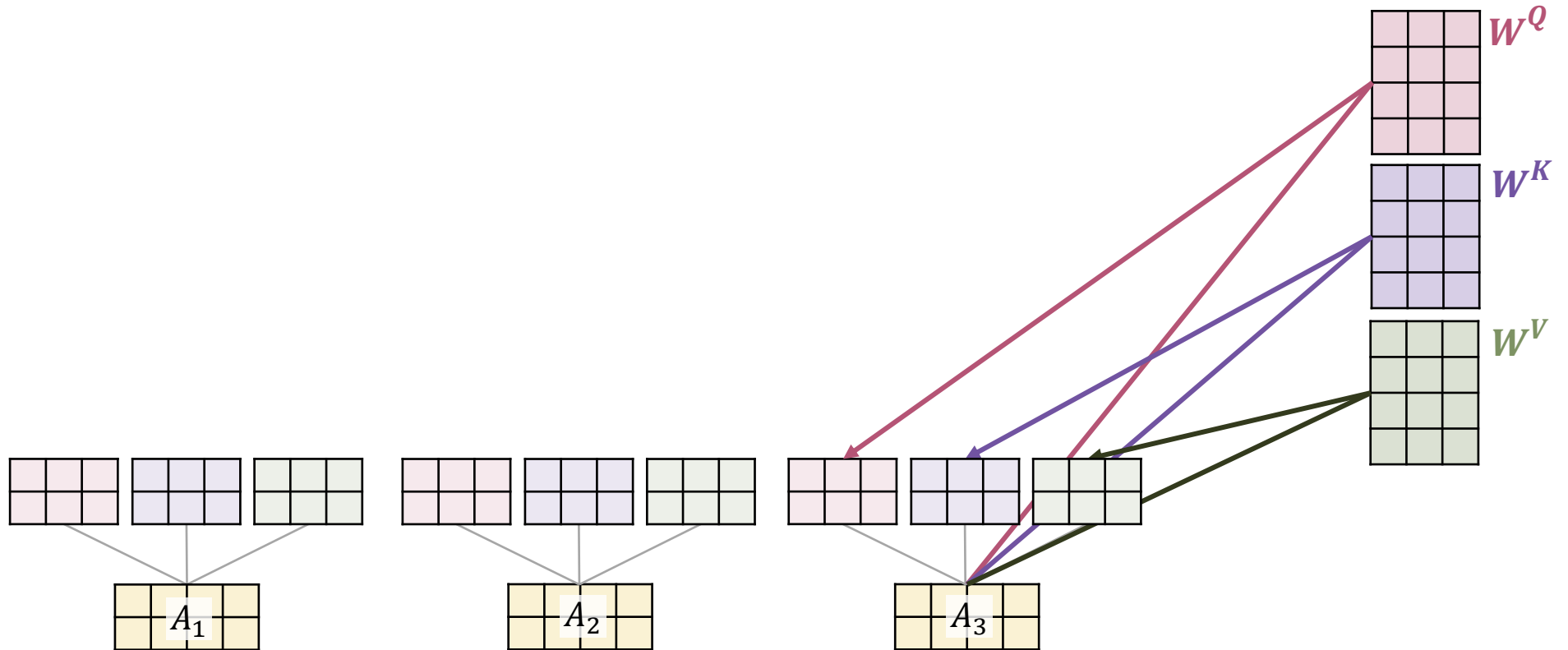


# Sequential Recommender System

Self-Attention : Matrix Calculation

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- Query
- Key
- Value

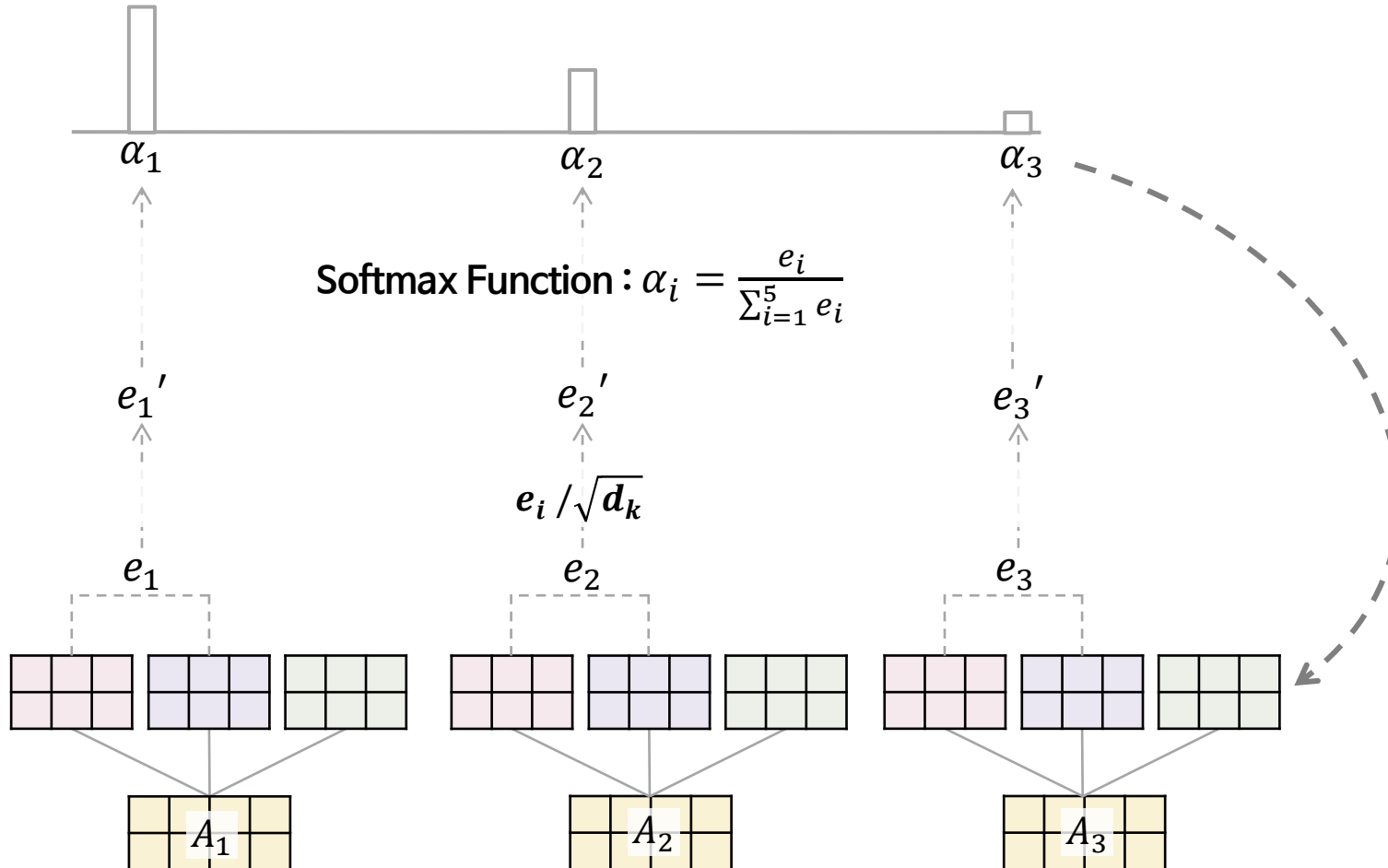


# Sequential Recommender System

Self-Attention : Matrix Calculation

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- Query
- Key
- Value

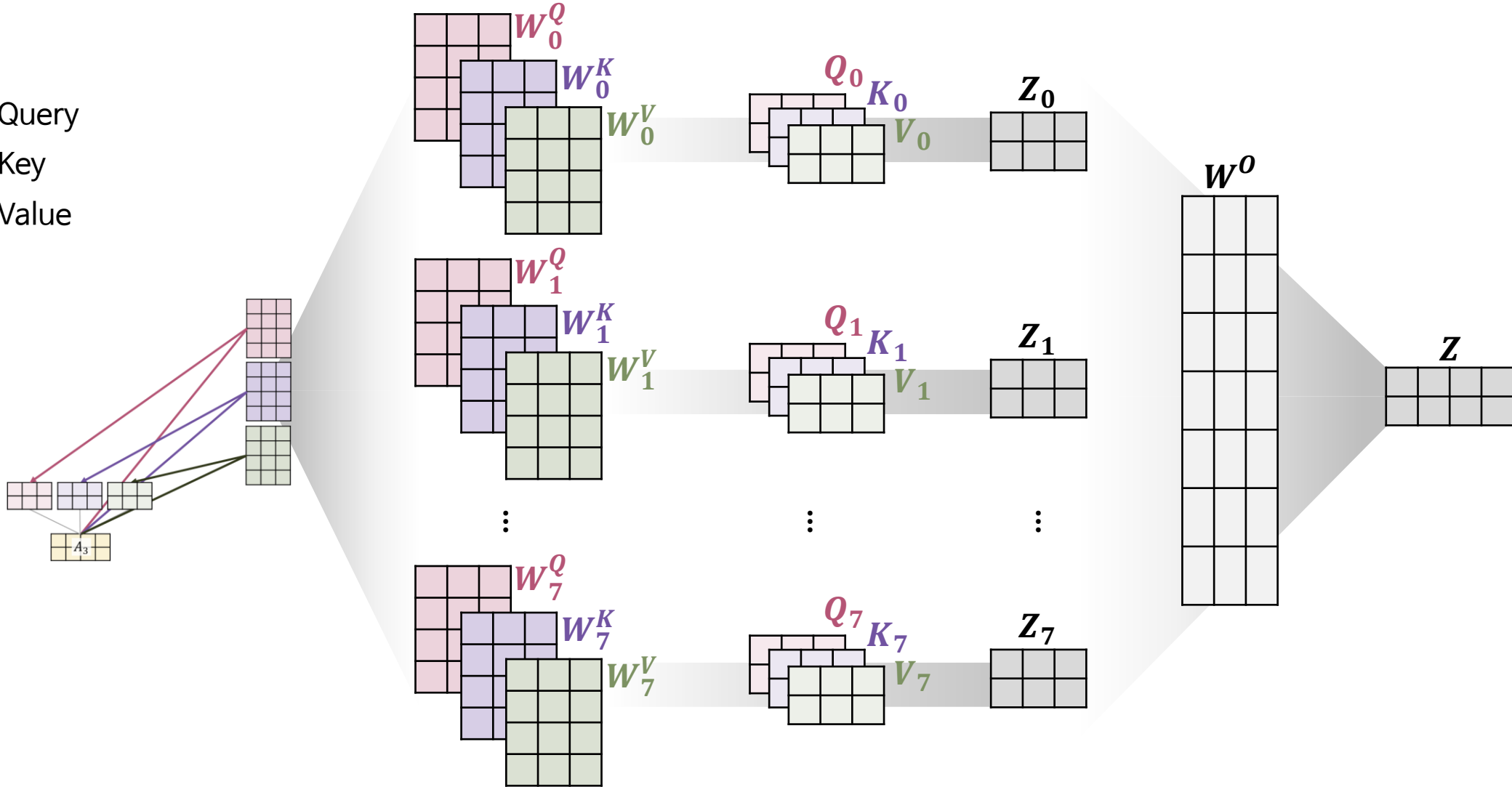


# Sequential Recommender System

## Multi-Head Attention

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- Query
- Key
- Value



# Sequential Recommender System

## Feature-level Deeper Self-Attention Network for Sequential Recommendation

- International Joint Conference on Artificial Intelligence 에서 2019년 발표된 논문
- 2020년 10월 20일 기준 17회 인용

### Feature-level Deeper Self-Attention Network for Sequential Recommendation

Tingting Zhang<sup>1,2</sup>, Pengpeng Zhao<sup>1,2\*</sup>, Yanchi Liu<sup>3</sup>, Victor S. Sheng<sup>4</sup>,  
Jiajie Xu<sup>1</sup>, Deqing Wang<sup>5</sup>, Guanfeng Liu<sup>6</sup> and Xiaofang Zhou<sup>7,2</sup>

<sup>1</sup>Institute of AI, School of Computer Science and Technology, Soochow University, China

<sup>2</sup>Zhejiang Lab, China

<sup>3</sup>Rutgers University, New Jersey, USA

<sup>4</sup>University of Central Arkansas, Conway, USA

<sup>5</sup>School of Computer, Beihang University, Beijing, China

<sup>6</sup>Department of Computing, Macquarie University, Sydney, Australia

<sup>7</sup>The University of Queensland, Brisbane, Australia

#### Abstract

Sequential recommendation, which aims to recommend next item that the user will likely interact in a near future, has become essential in various Internet applications. Existing methods usually consider the transition patterns between features of items, but ignore the transition patterns between features of items. We argue that only the item-level sequences cannot reveal the full sequential patterns, while explicit and implicit feature-level sequences can help extract the full sequential patterns. In this paper, we propose a novel method named Feature-level Deeper Self-Attention Network (FDSA) for sequential recommendation. Specifically, FDSA first integrates various heterogeneous features of items into feature sequences with different weights through a vanilla attention mechanism. After that, FDSA applies separated self-attention blocks on item-level sequences and feature-level sequences, respectively, to model item transition patterns and feature transition patterns. Then, we integrate the outputs of these two blocks to a fully-connected layer for next item recommendation. Finally, comprehensive experimental results demonstrate that considering the transition relationships between features can significantly improve the performance of sequential recommendation.

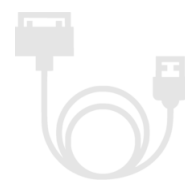
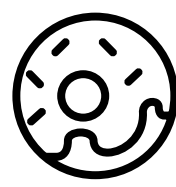
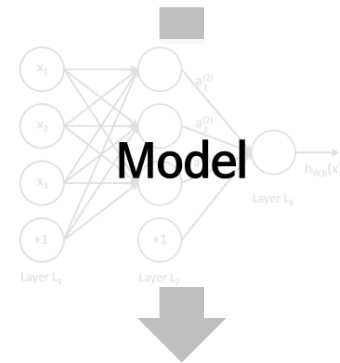
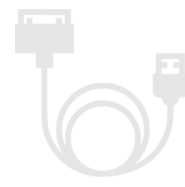
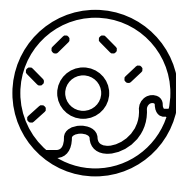
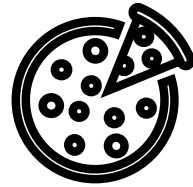
capturing useful sequential patterns from user historical behaviors.

Increasing research interests have been put in sequential recommendation with various models proposed. For modeling sequential patterns, the classic Factorizing Personalized Markov Chain (FPMC) model has been introduced to factorize the user-specific transition matrix by considering the Markov Chains [Rendle *et al.*, 2010]. However, the Markov assumption has difficulty in constructing a more effective relationship among factors [Huang *et al.*, 2018]. With the success of deep learning, Recurrent Neural Network (RNN) methods have been widely adopted in sequential recommendation [Hidasi *et al.*, 2016; Zhao *et al.*, 2019]. These RNN methods usually employ the last hidden state of RNN as the user representation, which is used to predict the next action. Despite the success, these RNN models are difficult to preserve long-range dependencies even using the advanced memory cell structures like Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) [Chung *et al.*, 2014]. Besides, RNN-based methods need to learn to pass relevant information forward step by step, which makes RNN hard to parallelize [Al-Rfou *et al.*, 2019]. Recently, self-attention networks (SANs) have shown promising empirical results in various NLP tasks, such as machine translation [Vaswani *et al.*, 2017], natural language inference [Shen *et al.*, 2018], and question answering [Li *et al.*, 2019]. One strong point of self-attention networks is the strength of capturing long-range dependencies by calculating attention weights between

# Sequential Recommender System

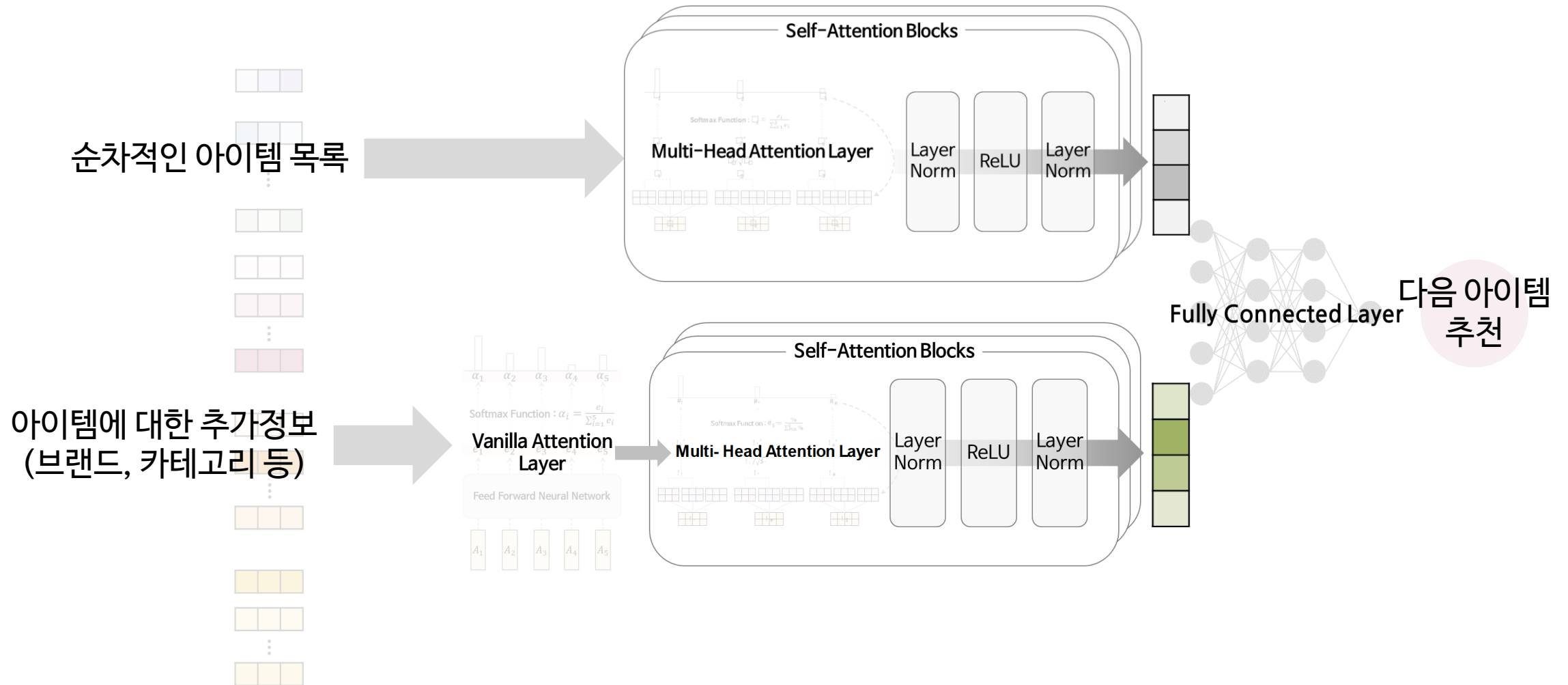
모델 인풋과 아웃풋 형태

- 인풋 : 한 사용자의 순차적인 아이템 목록 중 앞 L개 아이템
- 아웃풋 : 한 칸 미룬 후의 L개 아이템

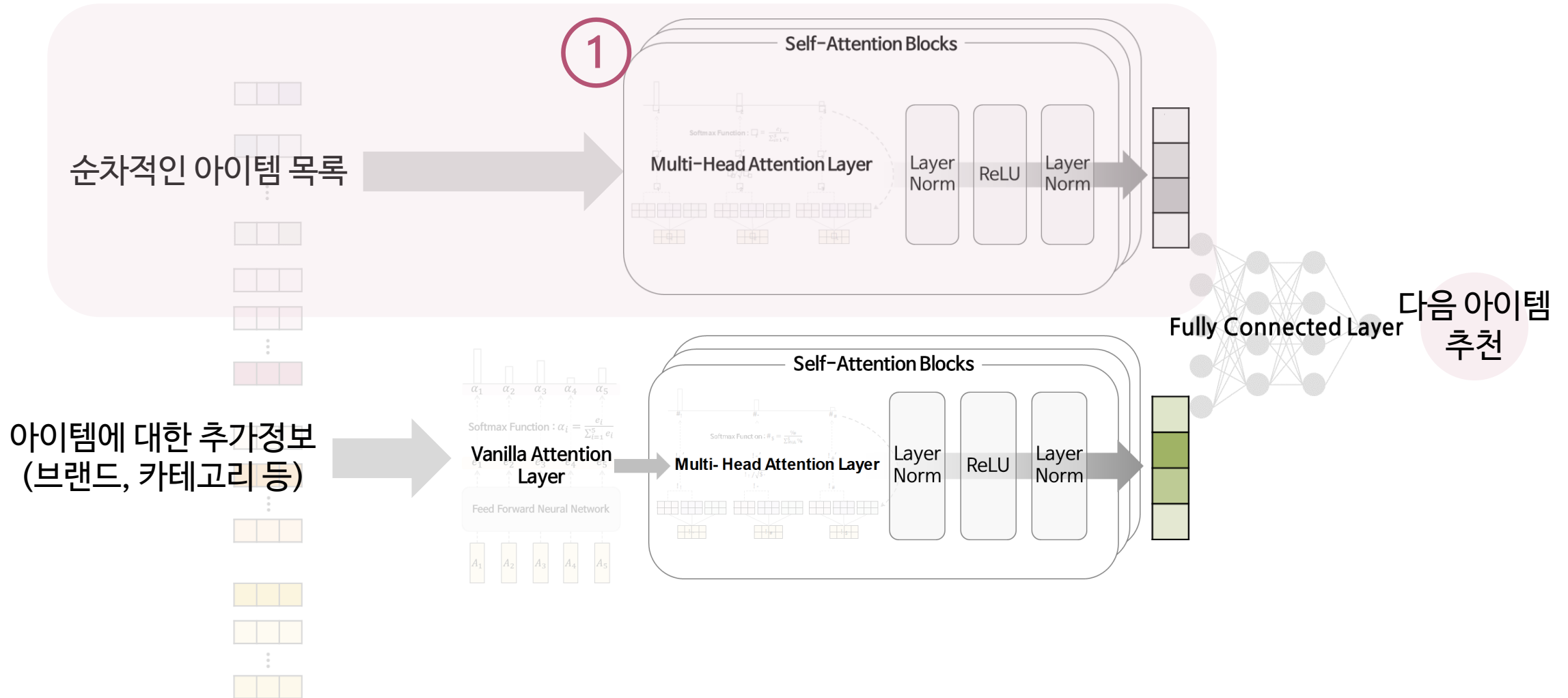


# Feature-level Deeper Self-Attention Network for Sequential Recommendation

모델의 전반적인 구조



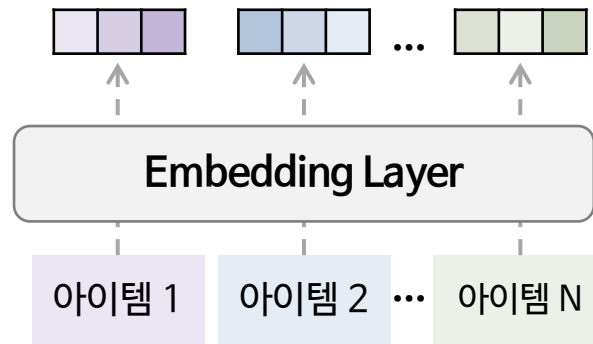
# Feature-level Deeper Self-Attention Network for Sequential Recommendation



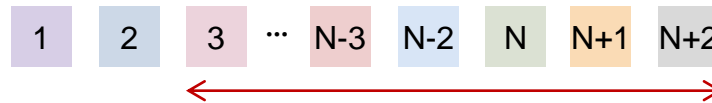
# Feature-level Deeper Self-Attention Network for Sequential Recommendation

모델의 구조 - 순차적인 아이템 정보 요약

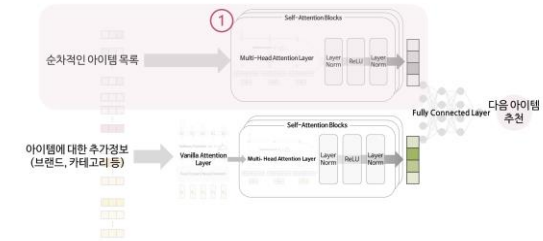
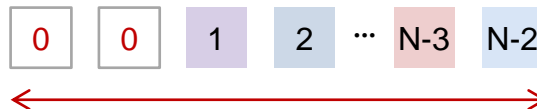
- 인풋으로 사용하기 위하여 순차적인 아이템 목록을 각각 위치정보를 반영한 벡터로 변환



- 전체 사용자의 아이템 개수가 N보다 많을 때



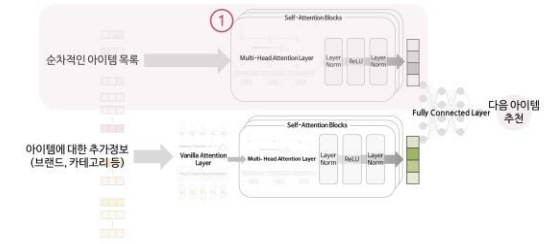
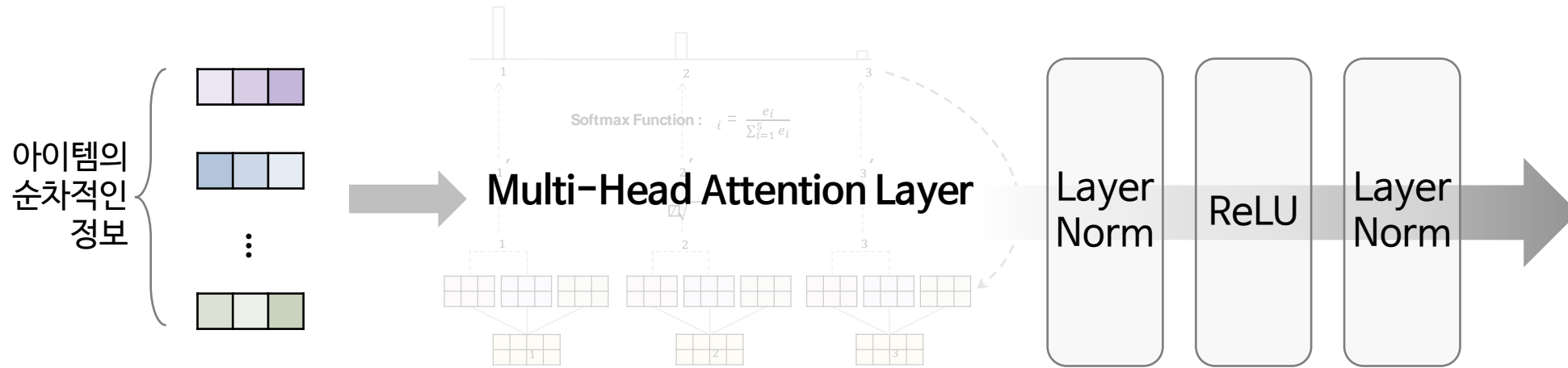
- 전체 사용자의 아이템 개수가 N보다 적을 때





# Feature-level Deeper Self-Attention Network for Sequential Recommendation

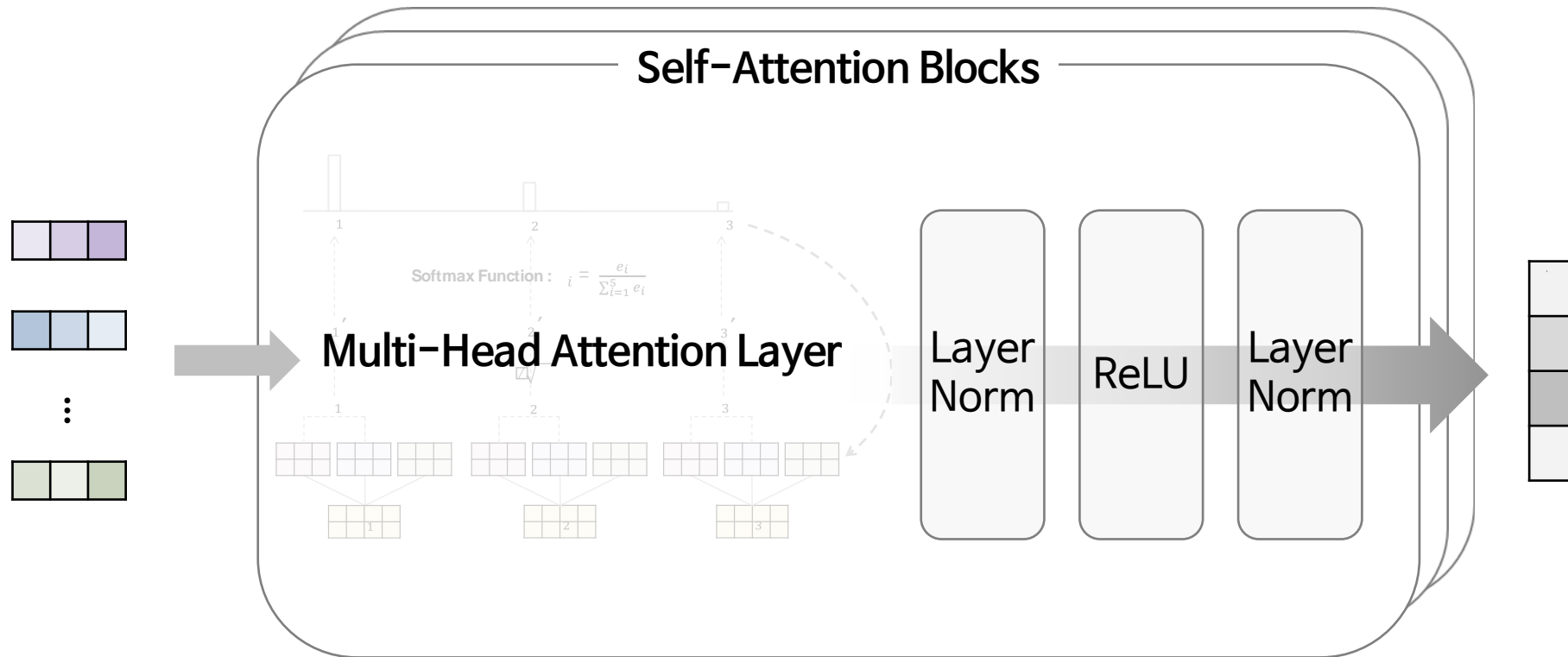
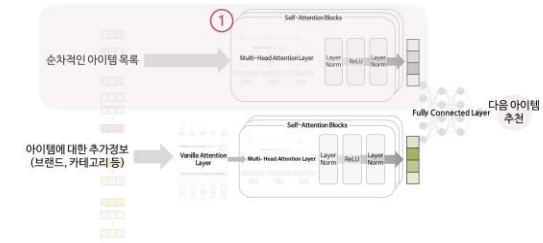
모델의 구조 - 순차적인 아이템 정보 요약



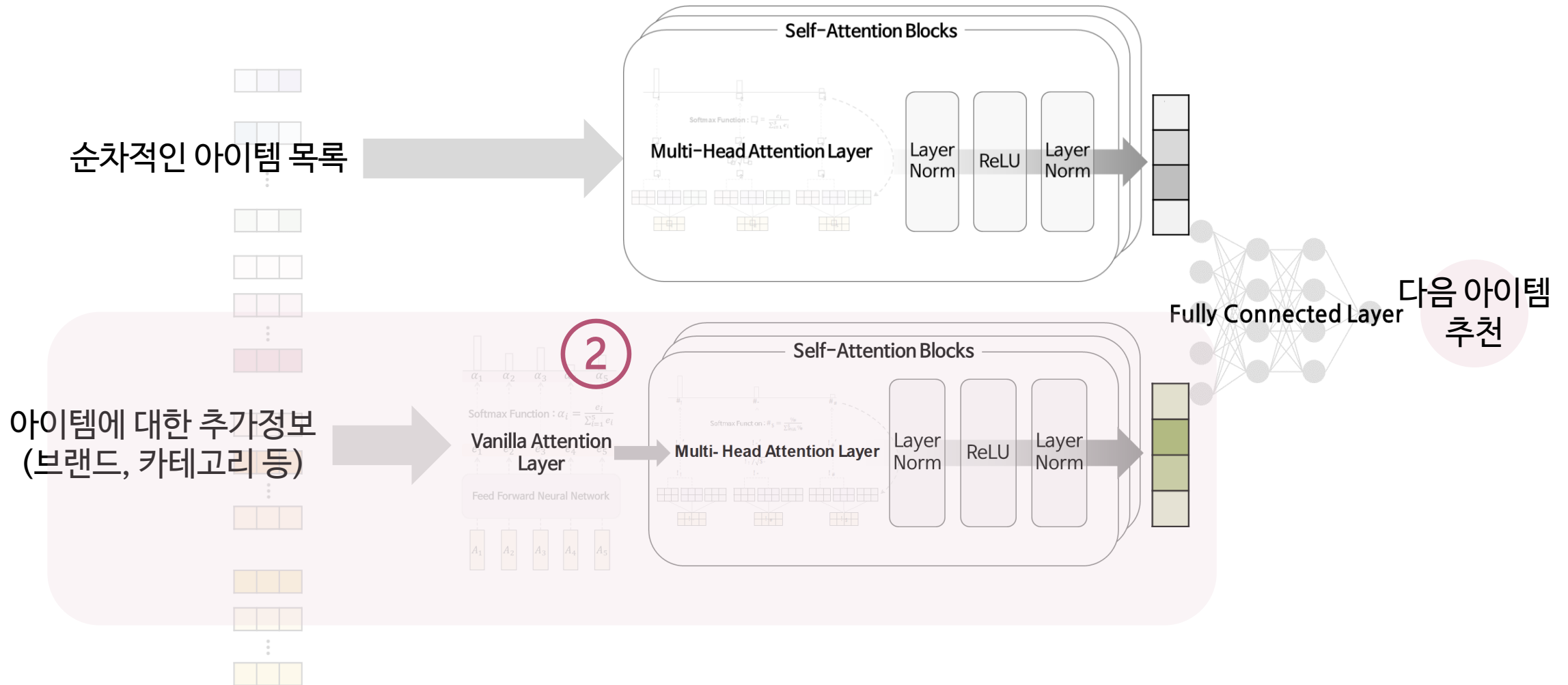
# Feature-level Deeper Self-Attention Network for Sequential Recommendation

모델의 구조 - 순차적인 아이템 정보 요약

- 더 복잡한 특성들을 파악하기 위하여 여러 Self-Attention Block을 결합하여 사용



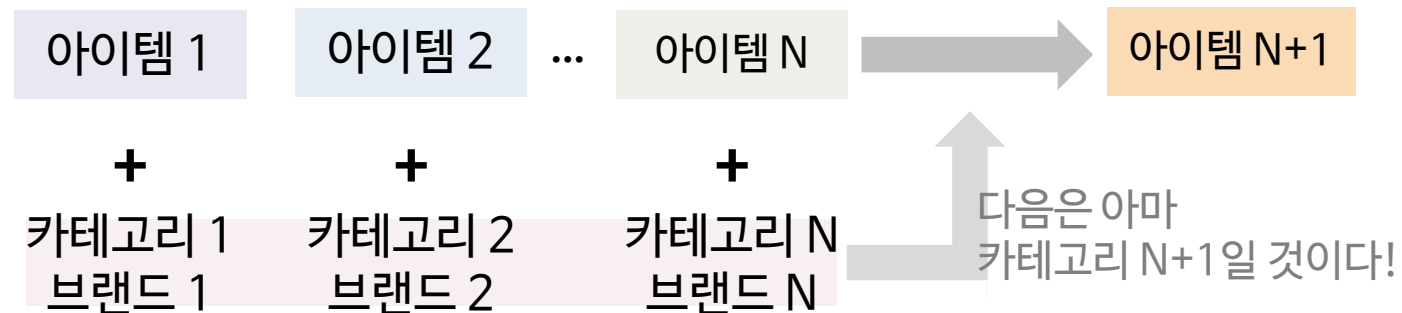
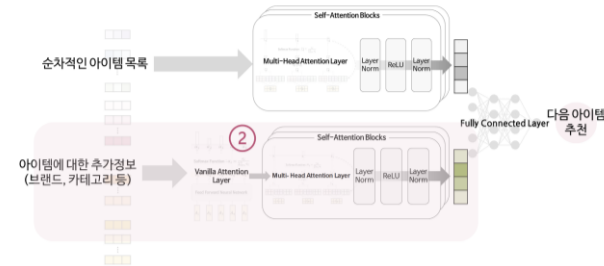
# Feature-level Deeper Self-Attention Network for Sequential Recommendation



# Feature-level Deeper Self-Attention Network for Sequential Recommendation

모델의 구조 - 추가적인 정보의 요약

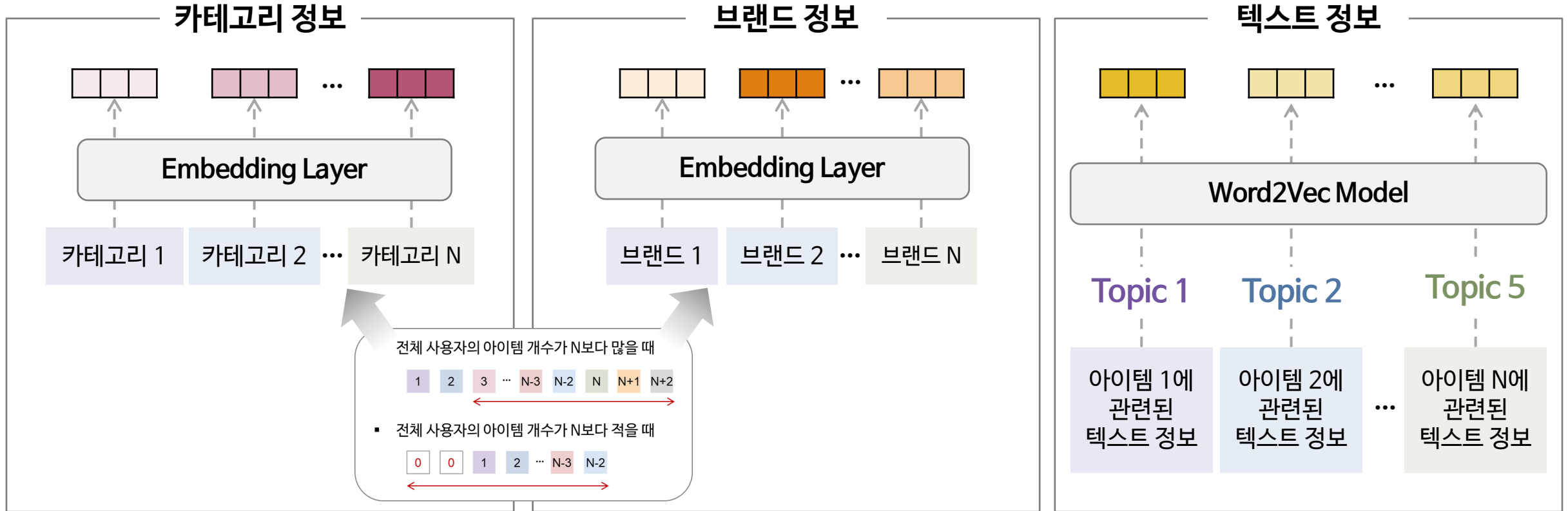
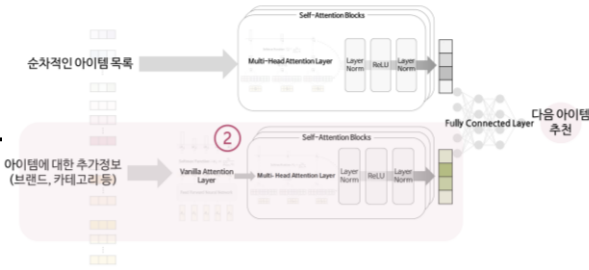
- 기존의 모델에서는 주로 순차적인 아이템 정보만을 사용하여 추천
- 모델에 추가적인 정보를 제공함으로써 추천 성능을 높일 수 있는 구조 제안
- 사용자가 선택한 아이템의 부가적인 정보 (브랜드, 카테고리 등)을 추가적으로 요약하여 활용



# Feature-level Deeper Self-Attention Network for Sequential Recommendation

모델의 구조 - 추가적인 정보의 요약

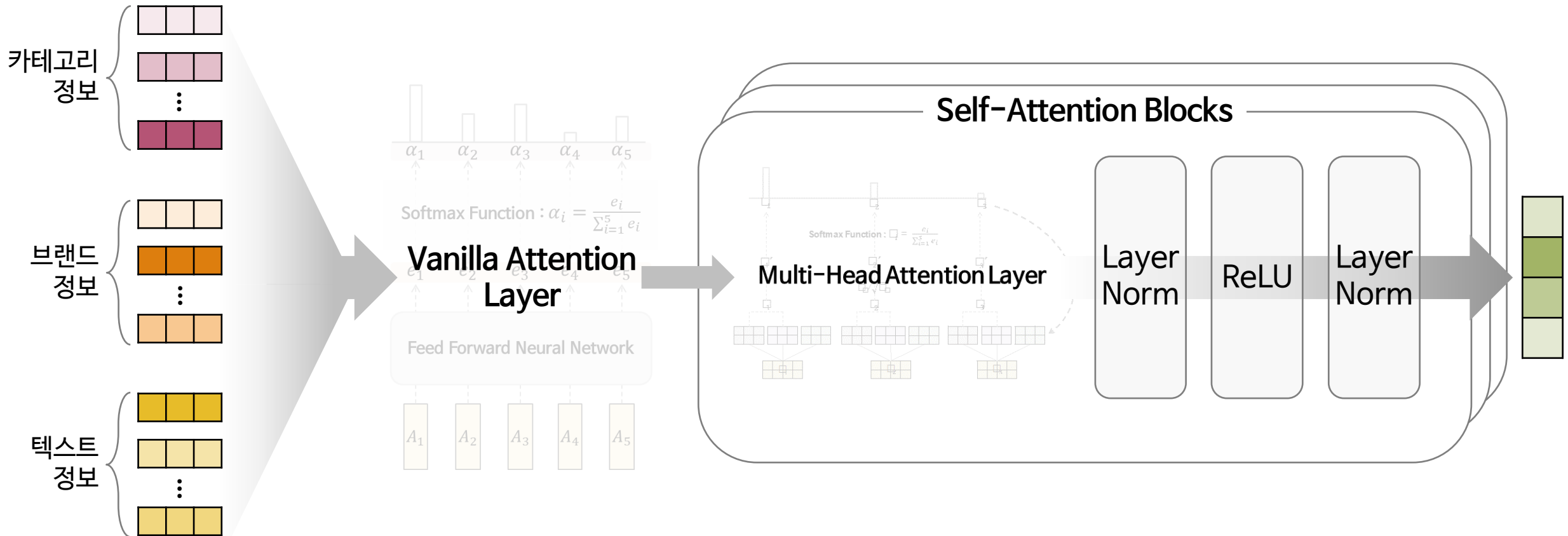
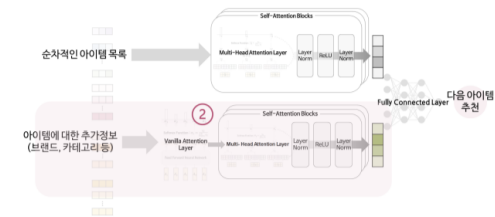
- 카테고리 정보, 브랜드 정보, 선정 아이템에 대한 텍스트 정보 등을 한번에 결합하는 방법 제안
- 인풋으로 사용하기 위하여 모든 정보를 Dense Vector Representation의 형태로 변환



# Feature-level Deeper Self-Attention Network for Sequential Recommendation

모델의 구조 - 추가적인 정보의 요약

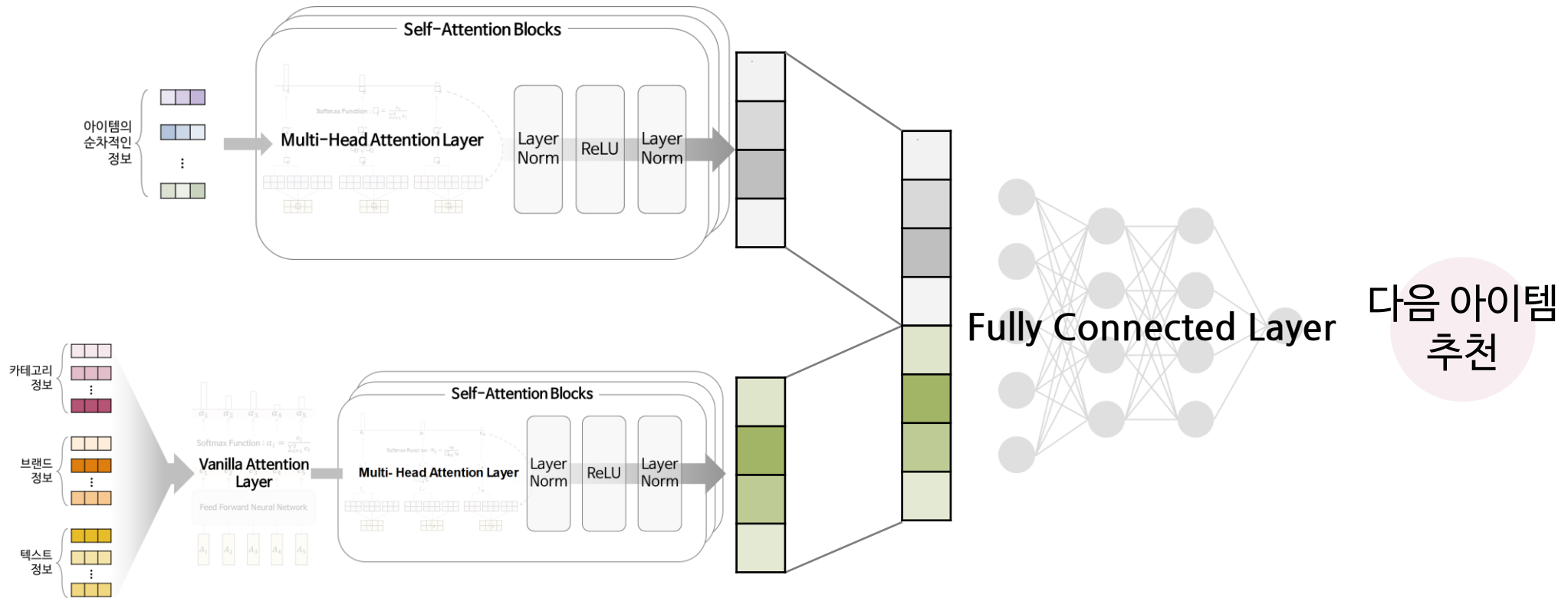
- 어느 정보가 사용자의 선택에 영향을 미쳤는지 알아보기 위한 Vanilla Attention Layer 활용
- 순차적인 정보는 보존한 채 각 아이템 정보 간 관계를 파악할 수 있는 Self-Attention Blocks 활용



# Feature-level Deeper Self-Attention Network for Sequential Recommendation

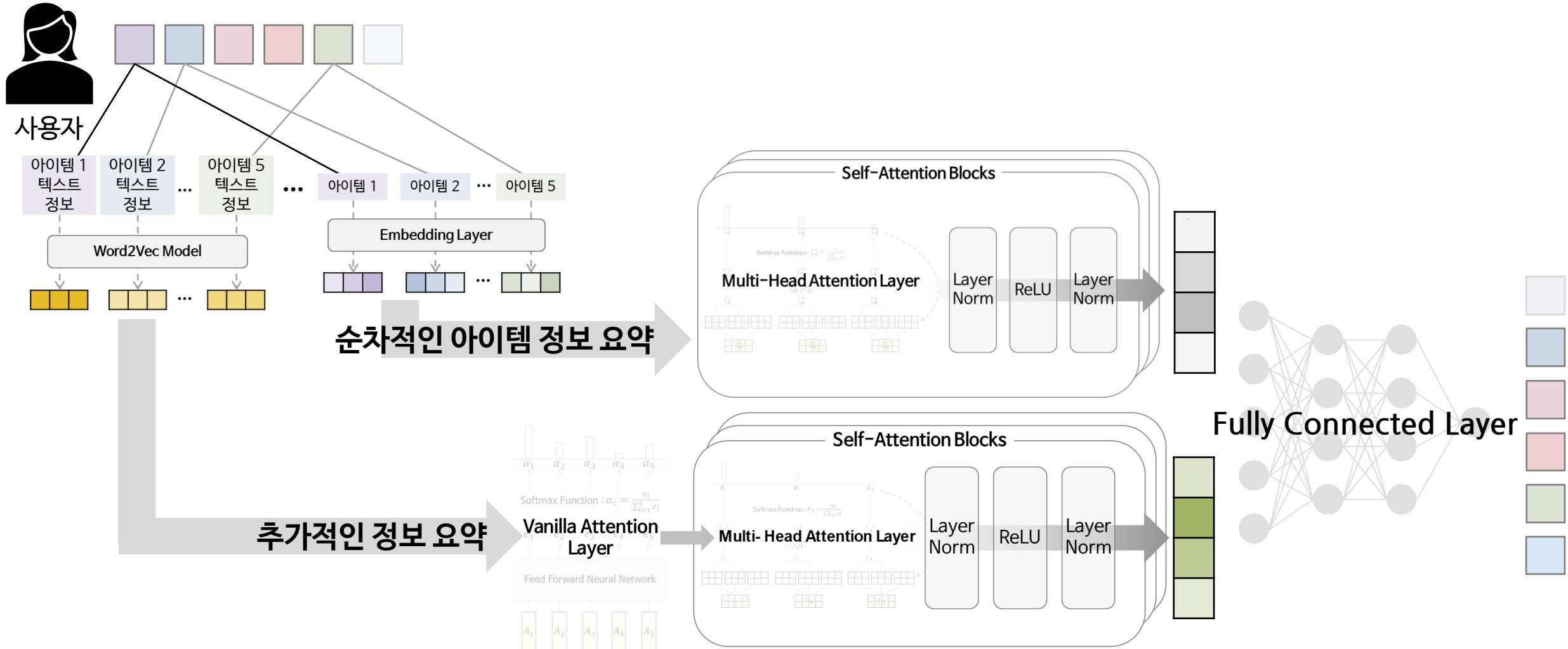
모델의 구조 - 결합 및 최종 결과값 도출

- 최종 결과값으로 인풋으로 들어온 다음 아이템을 예측



# Feature-level Deeper Self-Attention Network for Sequential Recommendation

모델의 전반적인 구조



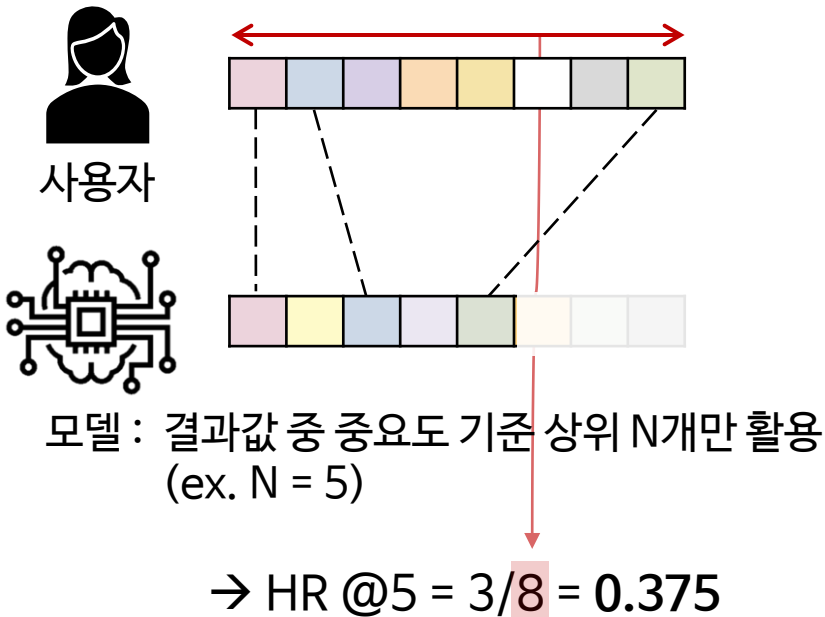


# Feature-level Deeper Self-Attention Network for Sequential Recommendation

## 평가지표

### Hit @N (Hit Ratio)

- 추천의 정확도를 나타내는 평가지표
- 사용자의 실제값 중 모델이 맞게 추천한 아이템의 수

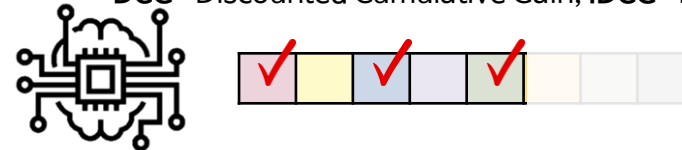


### NDCG @N (Normalized Discounted Cumulative Gain)

- 추천된 순위에 가중치를 주어 계산하는 평가지표
- 가장 이상적인 순위와 실제 추천된 순위에 대한 비교 진행

$$DCG = \sum_{i=1}^n \frac{relevance_i}{\log_2(i+1)}, \quad NDCG = \frac{DCG}{iDCG}$$

DCG : Discounted Cumulative Gain, iDCG : Ideal Discounted Cumulative Gain



순위	정답 여부	평점 (Relevance)
1	O	1
2	X	1
3	O	1
4	X	1
5	O	1

$$DCG = \frac{1}{\log_2(1+1)} + \frac{1}{\log_2(3+1)} + \frac{1}{\log_2(5+1)}$$

$$= 1.8869$$

$$iDCG = \frac{1}{\log_2(1+1)} + \frac{1}{\log_2(2+1)} + \frac{1}{\log_2(3+1)}$$

$$= 2.1309$$

→ NDCG @5 = 1.8869/2.1309 = **0.8855**

# Feature-level Deeper Self-Attention Network for Sequential Recommendation

## 모델을 활용한 실험 결과

- 실험 결과, 이전에 제안된 다양한 순차적 추천시스템 알고리즘들보다 더 높은 성능을 보이고 있음을 확인

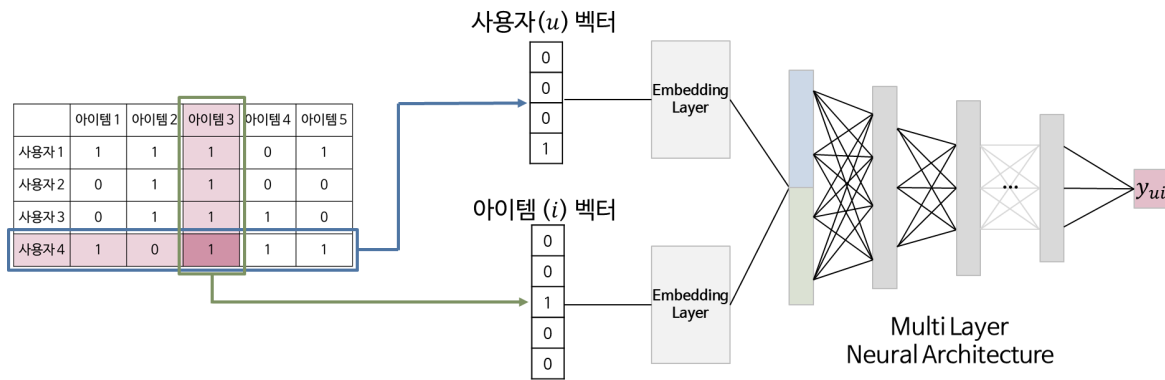
Dataset	Method	@5		@10	
		Hit	NDCG	Hit	NDCG
Tmall	PopRec	0.1532	0.0988	0.2397	0.1267
	BPR	0.1749	0.1129	0.2647	0.1418
	FPMC	0.2731	0.2034	0.3680	0.2339
	TransRec	0.2652	0.1854	0.3773	0.2214
	GRU4Rec	0.1674	0.1217	0.2446	0.1465
	CSAN	0.3481	0.2440	0.4787	0.2863
	SASRec	0.3572	0.2531	0.4840	0.2940
	SASRec+	0.3427	0.2415	0.4714	0.2829
	SASRec++	0.3550	0.2534	0.4785	0.2932
	CFSA	0.3836	0.2724	0.5152	0.3149
	<b>FDSA</b>	<b>0.3940</b>	<b>0.2820</b>	<b>0.5197</b>	<b>0.3226</b>
	Toys and Games	PopRec	0.1952	0.1287	0.3058
BPR		0.2096	0.1394	0.3219	0.1756
FPMC		0.2983	0.2261	0.3833	0.2535
TransRec		0.3135	0.2255	0.4206	0.2600
GRU4Rec		0.2039	0.1359	0.3118	0.1705
CSAN		0.2327	0.1601	0.3404	0.1947
SASRec		0.3292	0.2334	0.4441	0.2705
SASRec+		0.3367	0.2410	0.4510	0.2776
SASRec++		0.3394	0.2428	0.4544	0.2799
CFSA		0.3391	0.2411	0.4538	0.2782
<b>FDSA</b>		<b>0.3571</b>	<b>0.2572</b>	<b>0.4738</b>	<b>0.2949</b>

Table 2: Experimental results of FDSA and baselines. The best performance of each column (the larger is the better) is in bold.

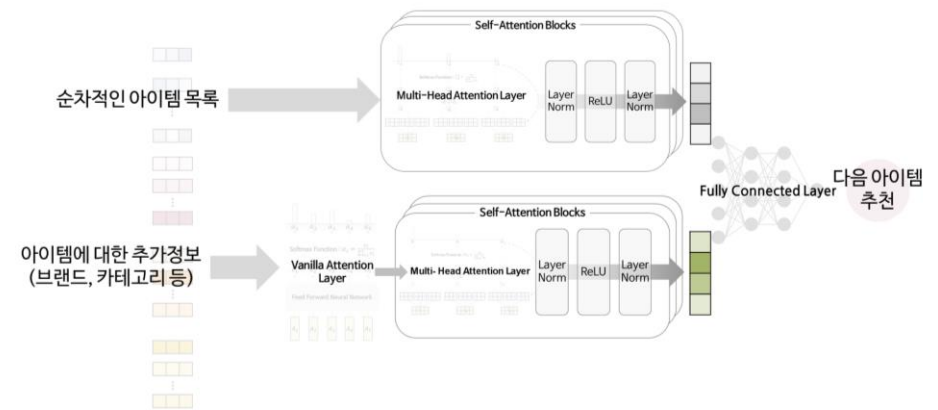
# Introduction to Sequential Recommender Systems

## 결론

- 기본적인 추천 시스템 알고리즘부터 순차적인 정보를 고려하는 추천 시스템 알고리즘까지 다양한 방향으로 발전
- 다양한 머신러닝 및 딥러닝 알고리즘들이 발전되면서 해당 모델을 추천시스템에 적용시키는 사례가 다양
- Feature-level Deeper Self-Attention Network for Sequential Recommendation
  - 사용자가 선택한 아이템의 순차적 정보와 그 아이템에 대한 세부적인 정보까지 반영한 방법론 제안
  - Multi-Head Attention 등의 기법을 활용하여 각 정보들을 요약하고, 이를 활용



Neural Collaborative Filtering



Feature-level Deeper Self-Attention Network for Sequential Recommendation

감사합니다.

## 참고자료

- <https://medium.com/sfu-csmp/recommendation-systems-collaborative-filtering-using-matrix-factorization-simplified-2118f4ef2cd3>
- <https://leehyejin91.github.io/post-ncf/>
- Son, Jieun, et al. “Review and Analysis of Recommender Systems.” *Journal of Korean Institute of Industrial Engineers*, vol. 41, no. 2, Korean Institute of Industrial Engineers, Apr. 2015, pp. 185 - 208. *Crossref*, doi:10.7232/jkiie.2015.41.2.185.
- He, X. et al. “Neural Collaborative Filtering.” *Proceedings of the 26th International Conference on World Wide Web* (2017): n. pag.
- Fang, Hui et al. “Deep Learning for Sequential Recommendation: Algorithms, Influential Factors, and Evaluations.” *arXiv: Information Retrieval* (2019): n. pag.
- Zhang, Tingting et al. “Feature-level Deeper Self-Attention Network for Sequential Recommendation.” *IJCAI* (2019).