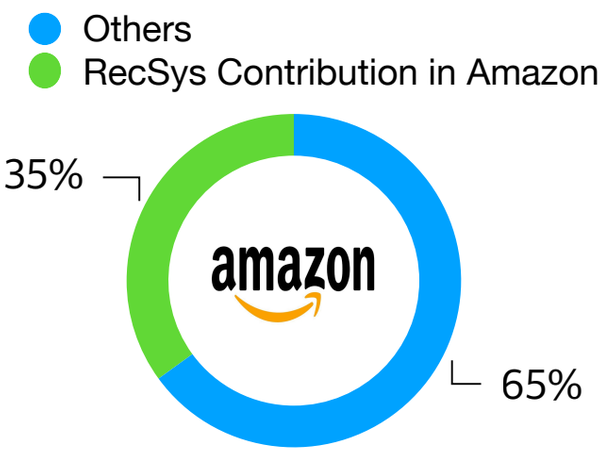
# Machine Learning for Personalization & RecSys:

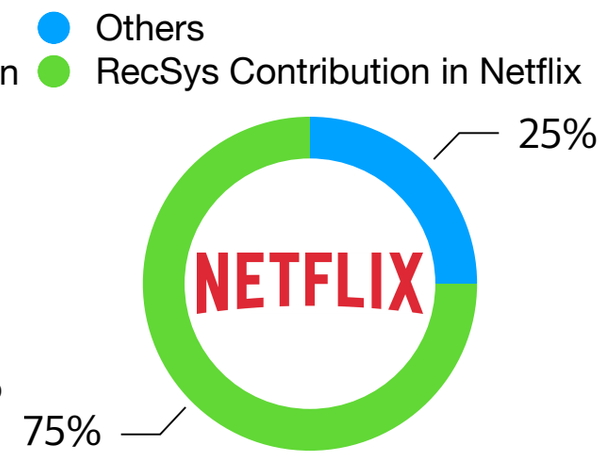
## Focusing on Collaborative Filtering

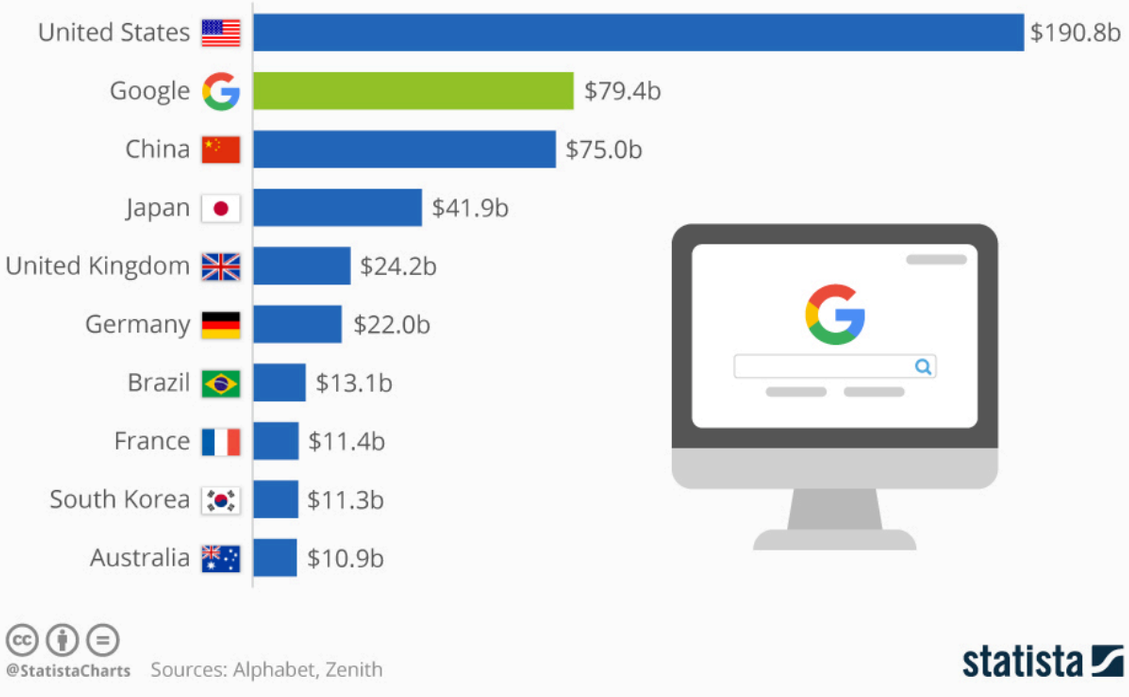
SK텔레콤 Data Biz. Platform 개발팀
이원성

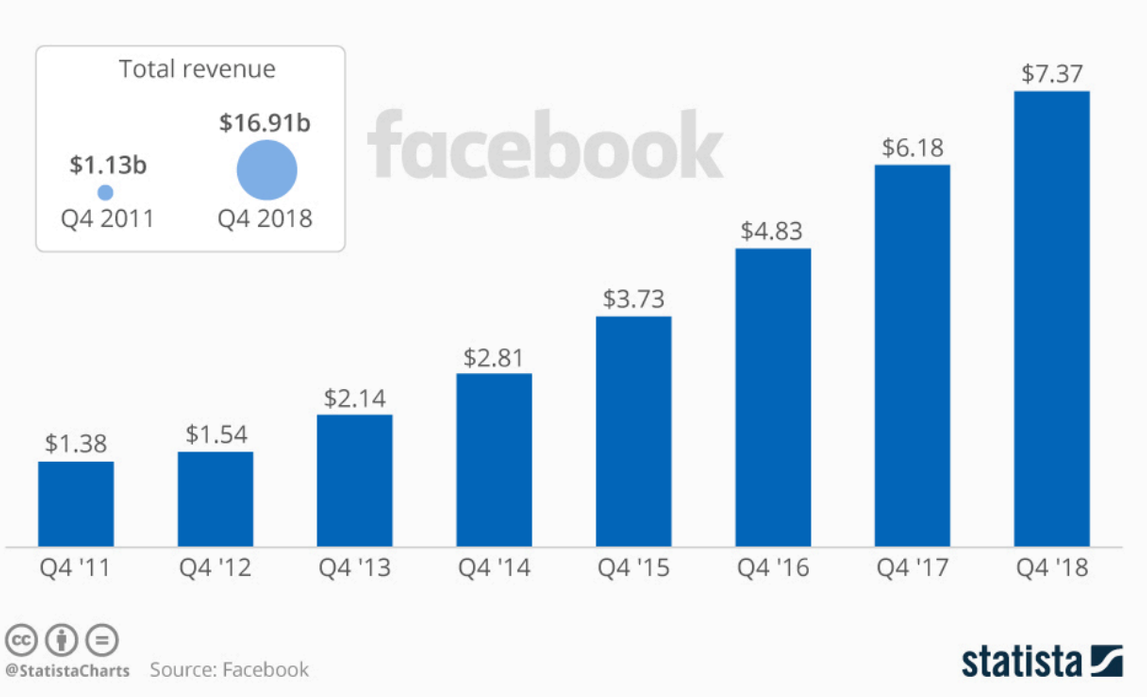
# 추천시스템, 개인화, 그리고 FANG

추천 시스템과 개인화 기술은 FANG으로 대표되는 거대 IT 기업의 성장에 중요한 역할을 수행함

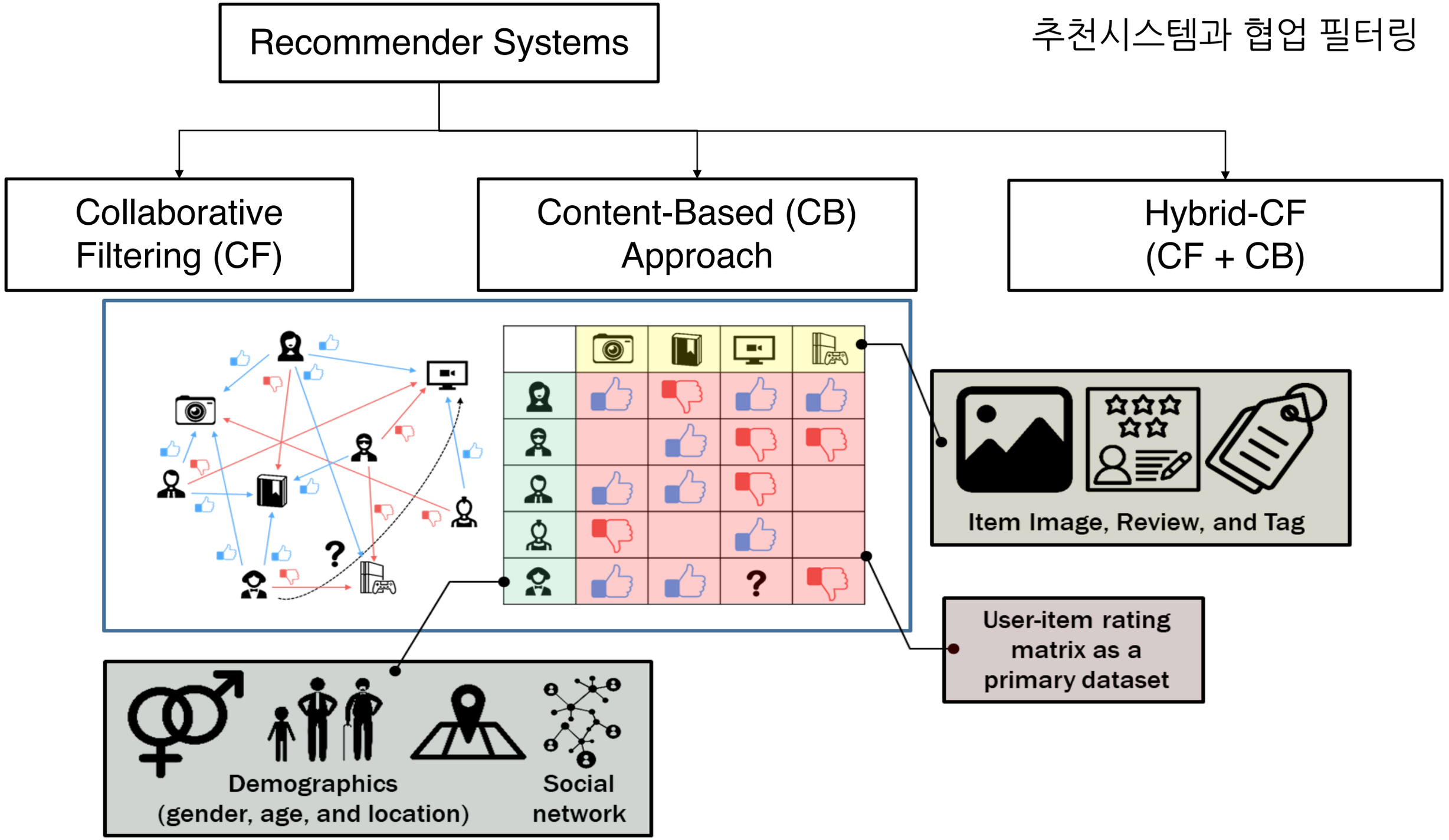



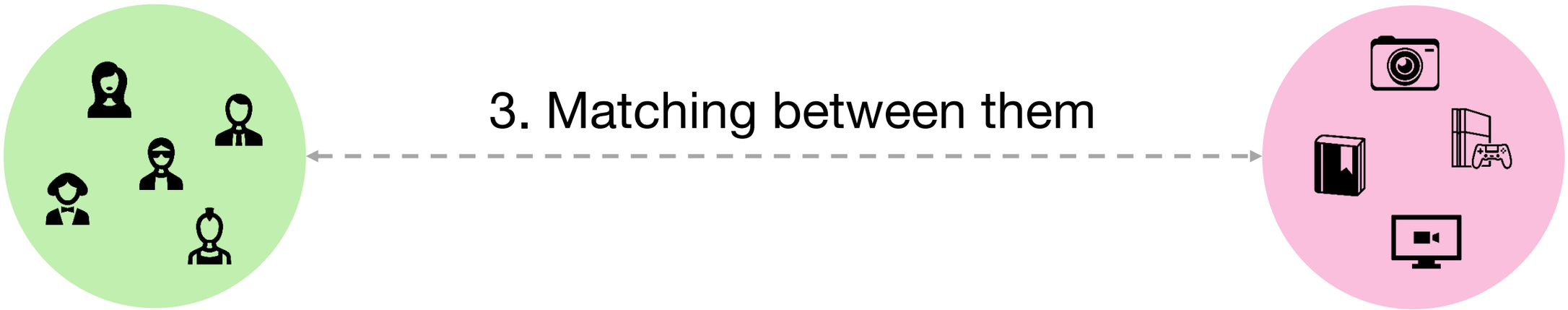
# 협업 필터링 (Collaborative Filtering) 알고리즘?

협업 필터링은 대용량의 데이터로부터 유용한 정보를 걸러내는 일련의 프로세스를 의미함



# 협업 필터링 (Collaborative Filtering) 알고리즘?

추천시스템과 개인화 관점에서, 사용자 및 아이템의 유의미한 정보를 찾아내는 일련의 방법론
더 나아가, 사용자와 아이템을 개인화되고 관련성 있는 방식으로 매칭해주는 기법



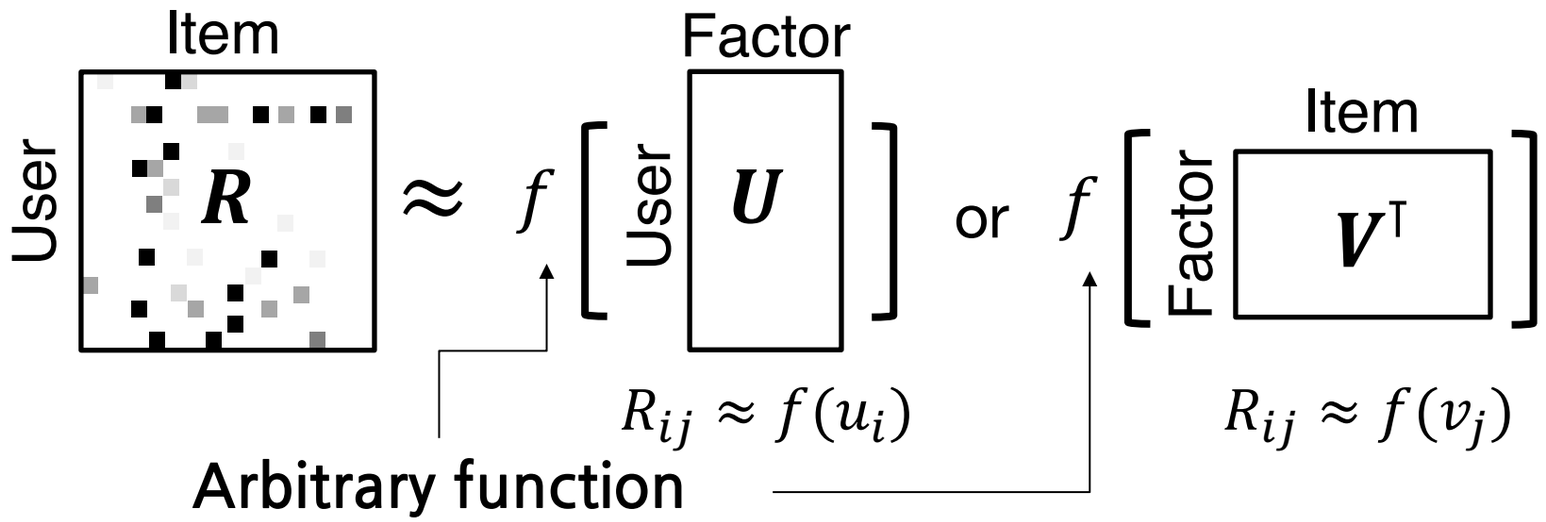

1. User representation

2. Item representation

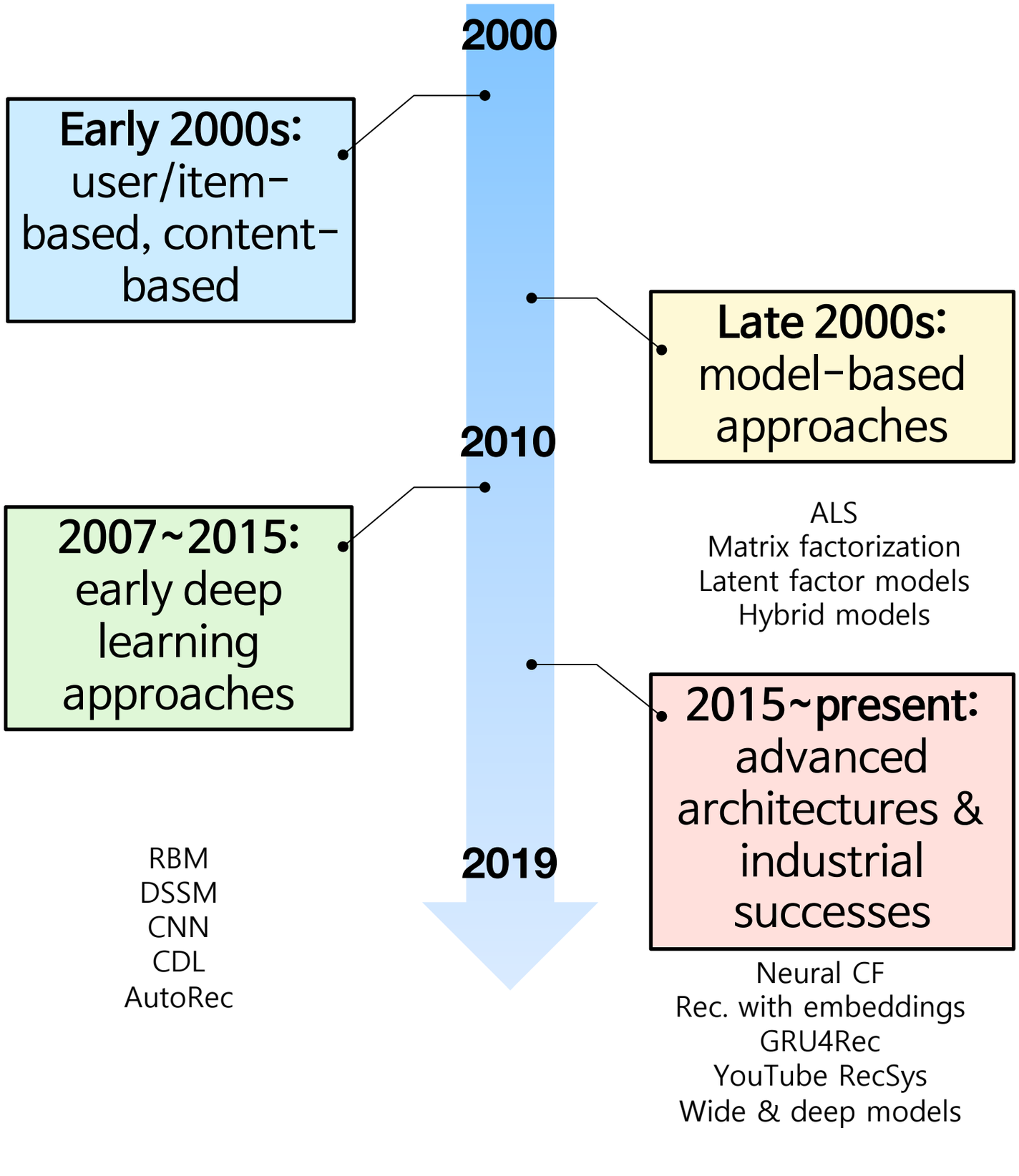
개인화 및 추천시스템을 위한 협업 필터링: 관측된 선호를 기반으로 사용자의 취향을 모델링

**목표?** 사용자의 만족을
최대화하는 추천 아이템 제공

**어떻게?** 관측된 사용자의 선호 및
다양한 컨텍스트에 기반한
사용자의 taste를 모델링



# 추천시스템을 위한 CF: (주관적인) 타임라인*




**Related Ad Tech Papers**

LR with FTRL-proximal (Google, 2013)

GBDT + LR (FB, 2014)

Field-aware FM (Criteo, 2016)

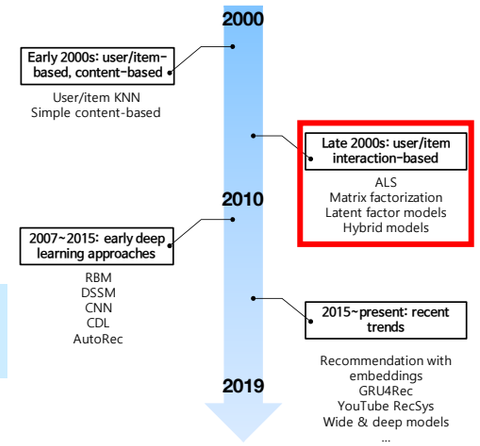
Deep & Cross Network (Google, 2017)

Deep Interest Network (Alibaba, 2018)

Deep Interest Evolution Network (Alibaba, 2019)

*http://hoondongkim.blogspot.com/2019/03/recommendation-trend.html
해당 내용은 **SKT** 김훈동 님의 블로그를 참고하였습니다.

# 사용자 혹은 아이템 기반 KNN




가장 기본적이나 꽤 유용한 방법으로 사용자 혹은 아이템의 유사도에 기반함

| | | | | |
|---|---|---|---|---|
| | 1 | -1 | 1 | 1 |
| | 0 | 1 | -1 | -1 |
| | 1 | 1 | -1 | 0 |
| | -1 | 0 | 1 | 0 |
| | 1 | 1 | ? | -1 |

1. Neighbor selection

2. Neighbor contribution

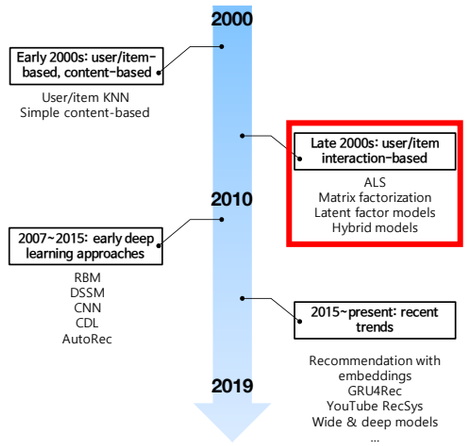
$$\hat{R}_{i,j} = \frac{\sum_{k \in N_j} R_{i,k} \times sim(j,k)}{\sum_{k \in N_j} sim(j,k)}$$

3. Item similarity

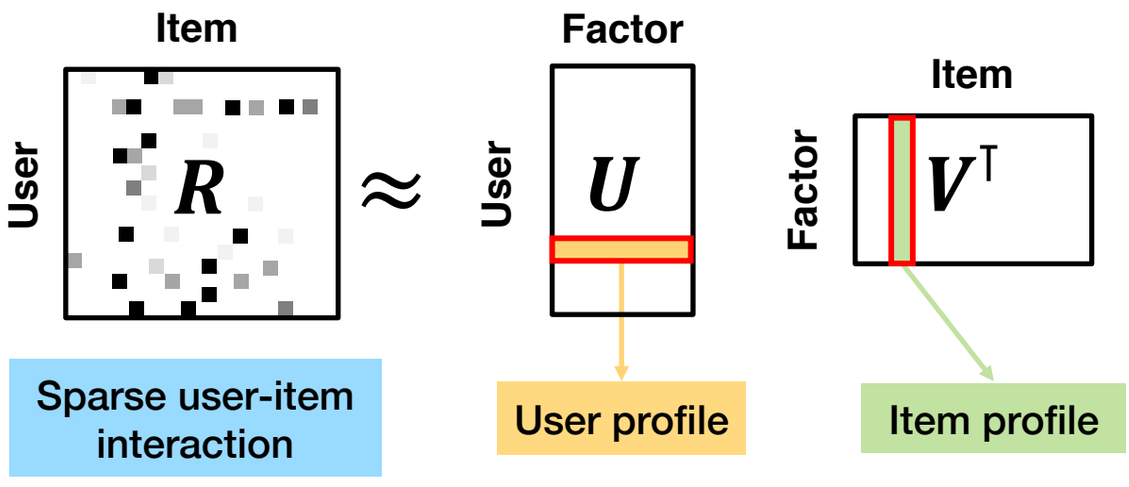
User: $i$, Item: $j$, Item Neighbor: $k$
Similarity between item $j$ and $k$: $sim(j,k)$

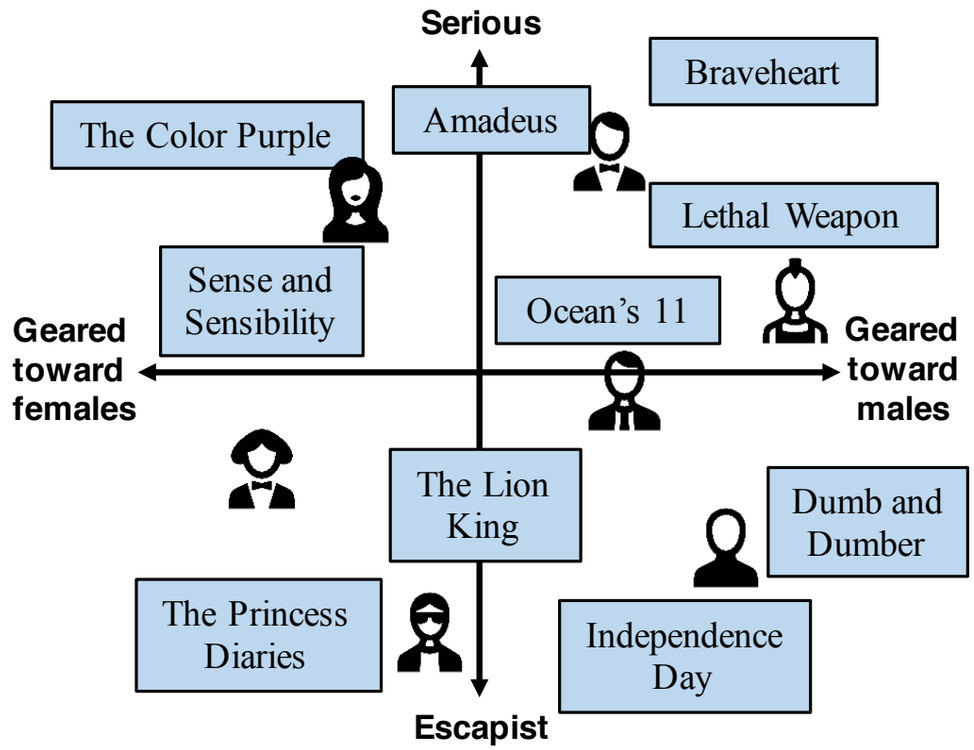
Similarity-weighted prediction: **Utilizes top *K* neighbors to compute prediction**

# Alternating Least Squares (ALS)




선호도 추정을 행렬 분해 (MF)의 관점에 접근함
Low-rank approximation for the higher-order sparse matrix




borrowed from Koren et al. with modificaiton

$$R_{ij} \sim N(u_i^\intercal v_j, c_{ij}^{-1})$$

$$u_i \sim N(0, \lambda_U^{-1}\mathbf{I}_K)$$

$$v_j \sim N(0, \lambda_V^{-1}\mathbf{I}_K)$$

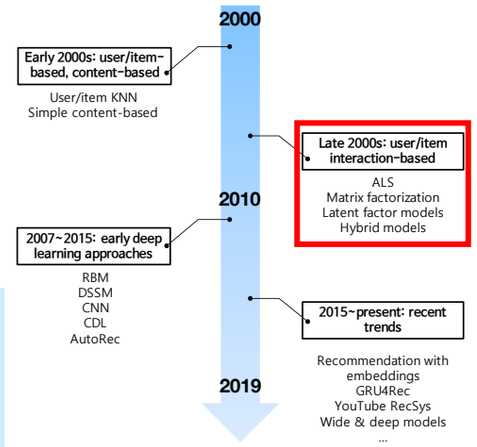
Maximizing the log-likelihood is equivalent to minimizing the objective:

$$\min_{U,V} \sum_{i,j} \left(R_{ij} - U_i V_j^\intercal\right)^2$$

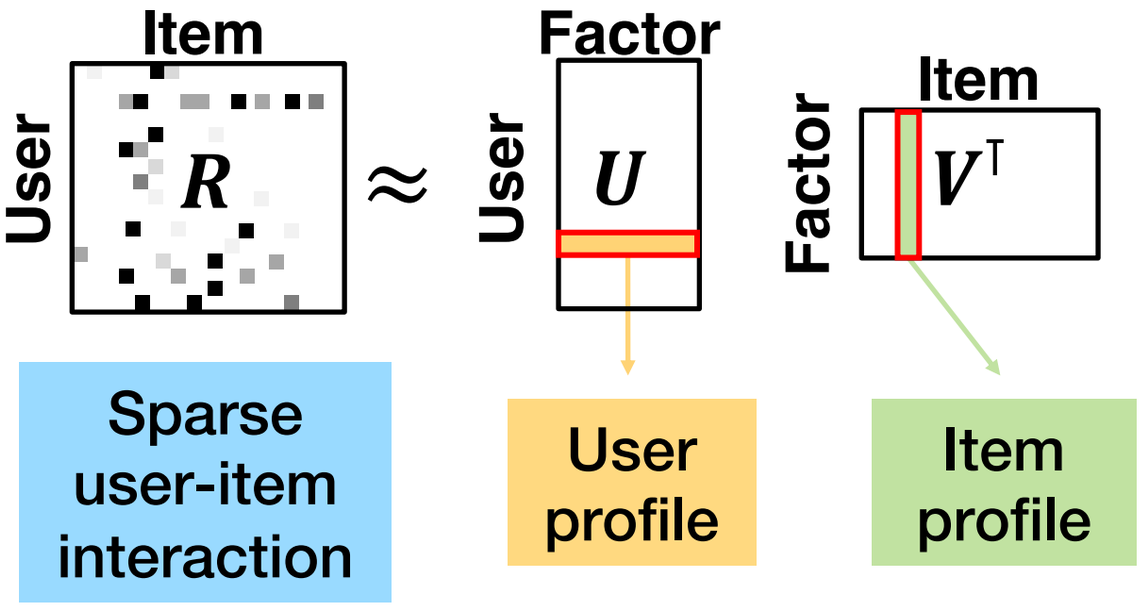
With confidence and regularization hyperparameters:

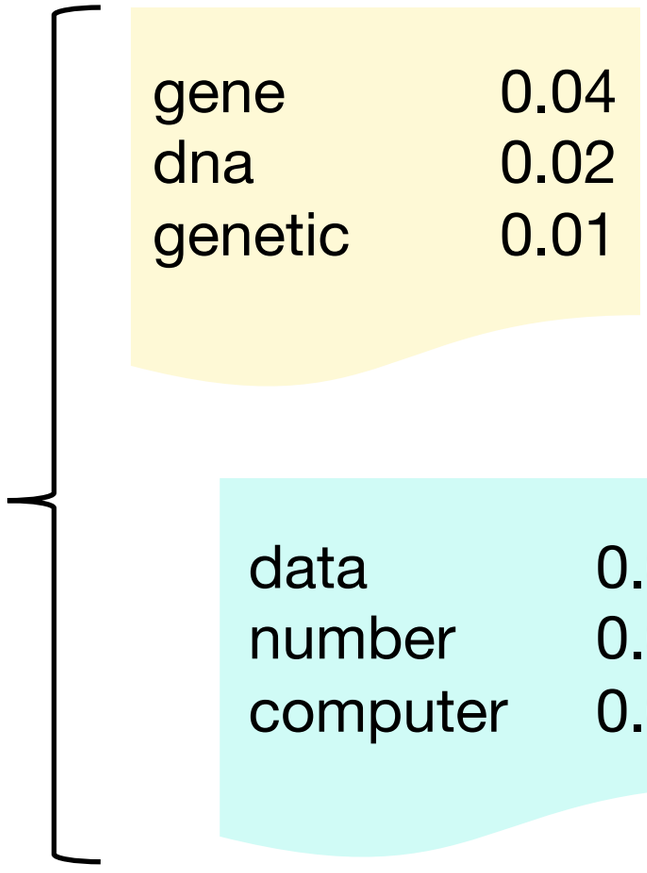
$$\min_{U,V} \sum_{i,j} c_{ij}\left(R_{ij} - U_i V_j^\intercal\right)^2 + \sum_i \lambda_U \|u_i\|^2 + \sum_j \lambda_V \|v_j\|^2$$

# MF + Content-Based Approach



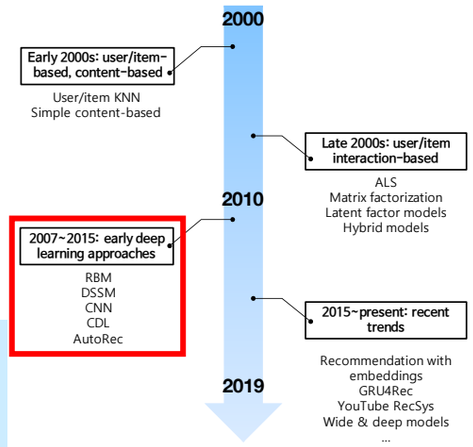

행렬 분해와 Content 모델링을 결합하는 형태의 Hybrid 방법론
e.g. Collaborative Topic Model: MF + LDA



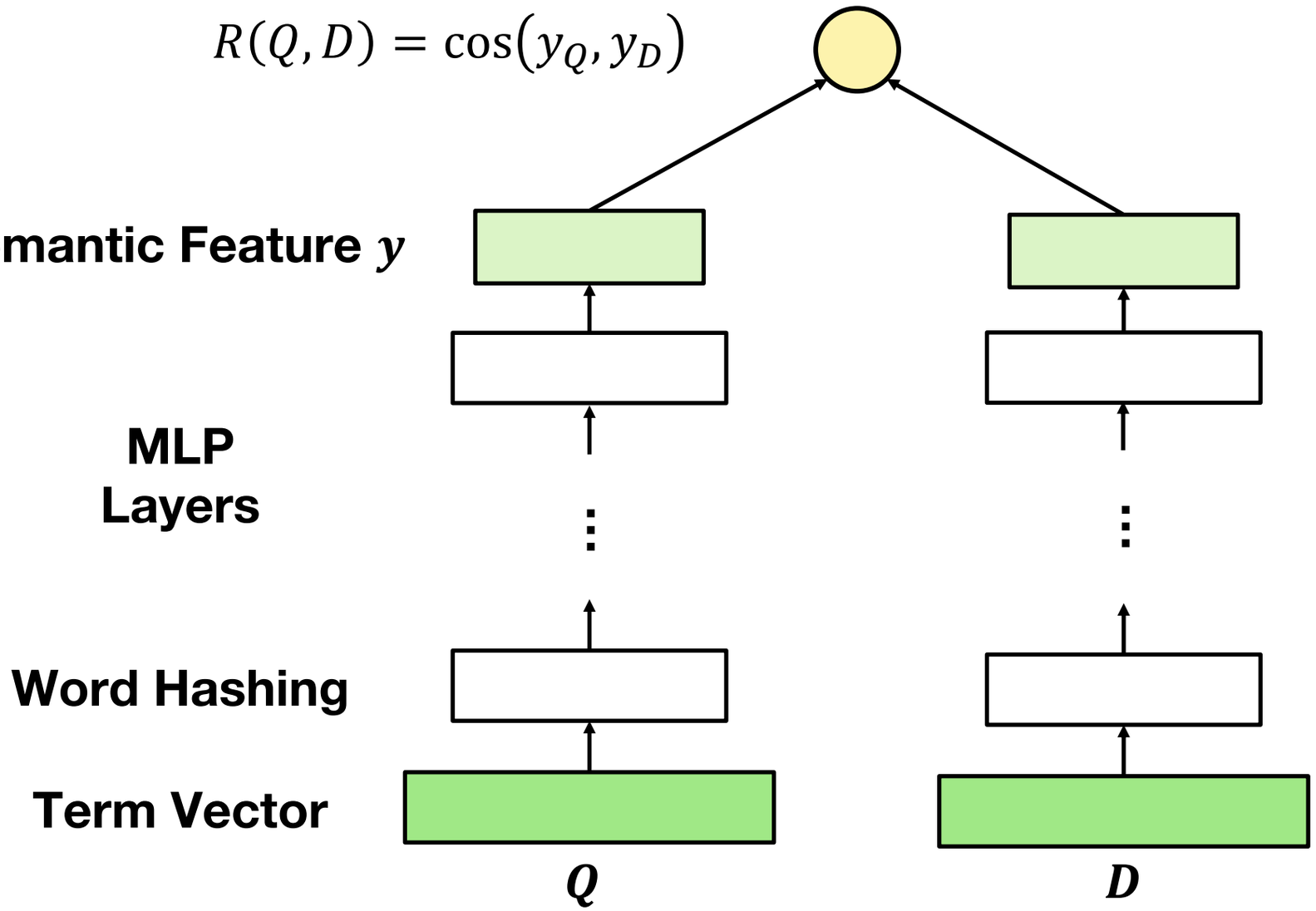

사용자의 Preference와 아이템의 Topic을 같은 공간에서 모델링함으로써 발견된 Latent Topic와 사용자 Preference와의 유사도에 기반하여 아이템을 추천함

# Compute Similarity in Semantic Space




Deep Structured Similarity Modeld (DSSM)은 본래 web search에서 쿼리와 관련 있는 문서들 사이의 semantic matching을 위해 제안됐으며, 최근 AWS에서 이를 추천을 위한 형태로 re-invented하여 제공하고 있음

$$R(Q,D) = \cos(y_Q, y_D)$$

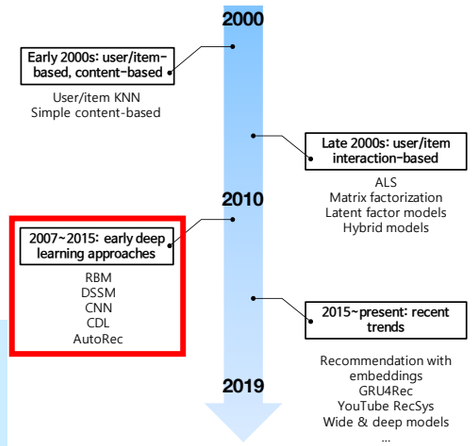



$$p(D|Q) = \frac{\exp(\gamma R(Q,D))}{\sum_{D' \in \mathbf{D}} \exp(\gamma R(Q,D'))}$$

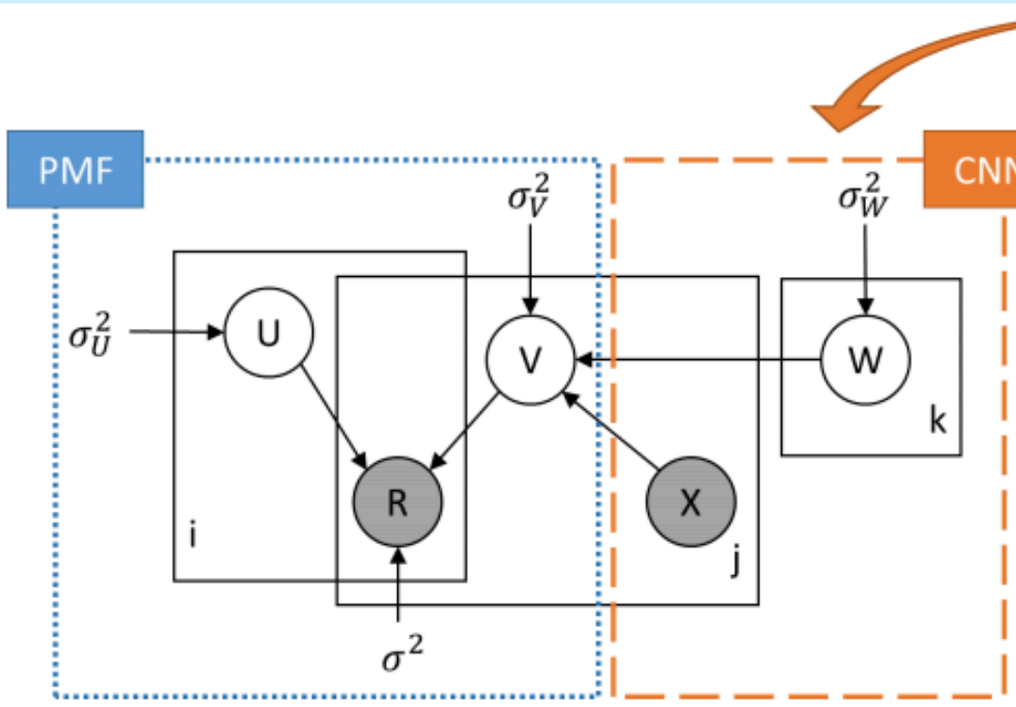
$\gamma$: smoothing factor for softmax fn.

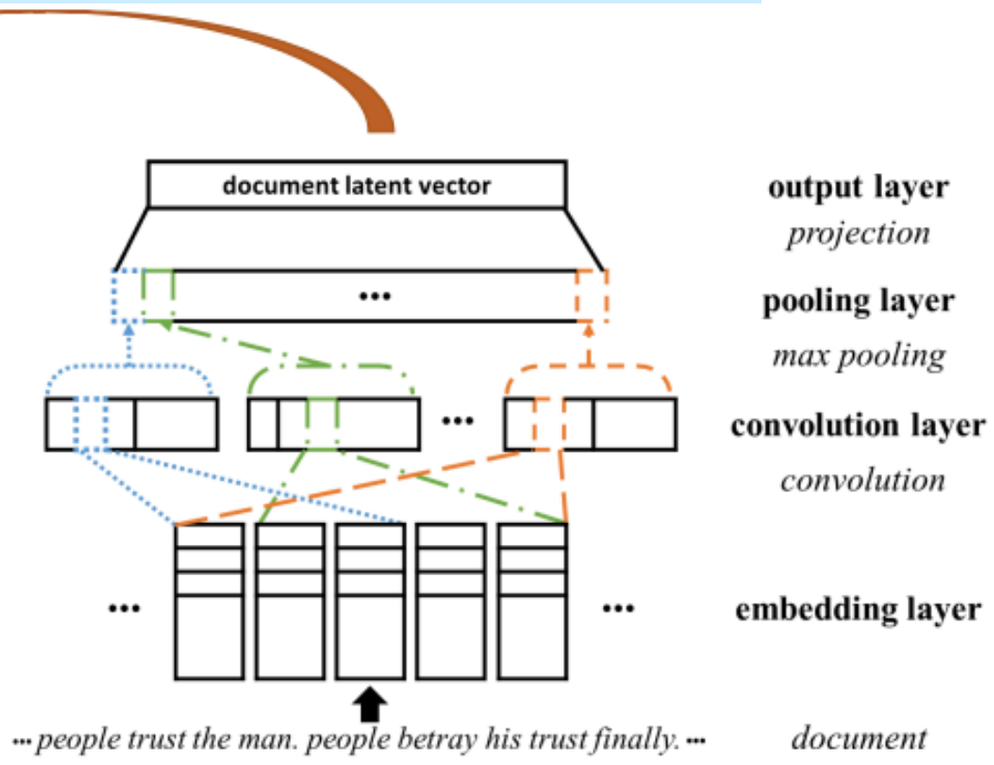
$$loss = -\log \prod_{(Q,D^+)} P(D^+|Q)$$

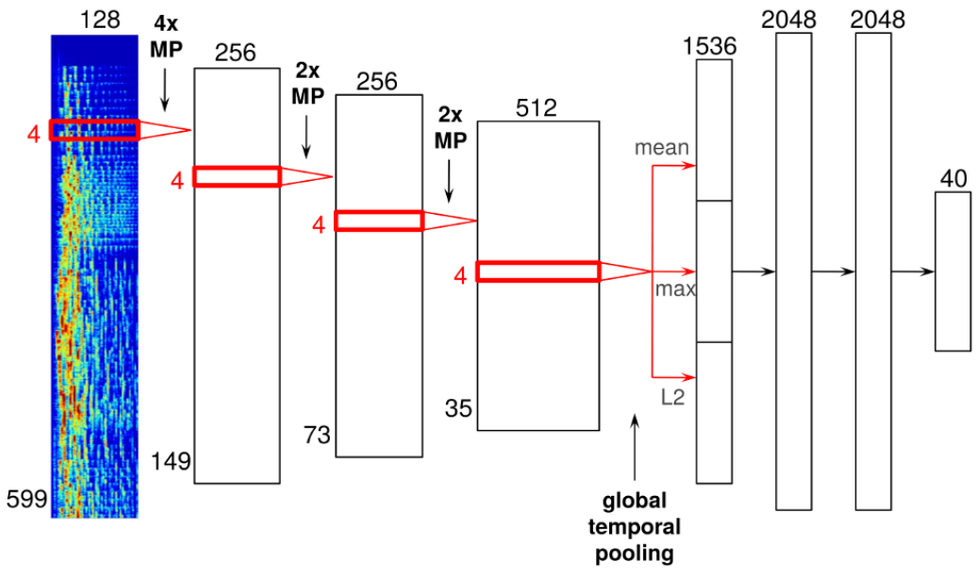
# DL for Content-Based Approaches



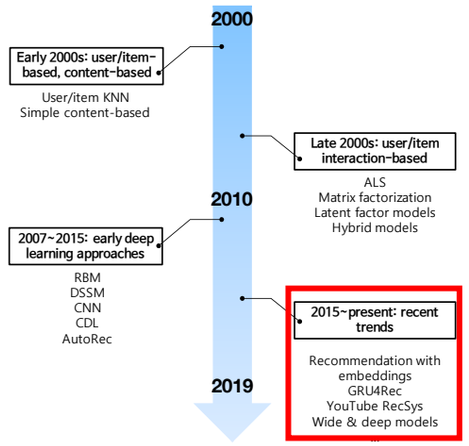

Deep Learning의 뛰어난 representation learning을 다양한 방식으로 content modeling에 적용하는 사례가 늘고 있음
CNN for music rec., SDAE or CNN for document modeling



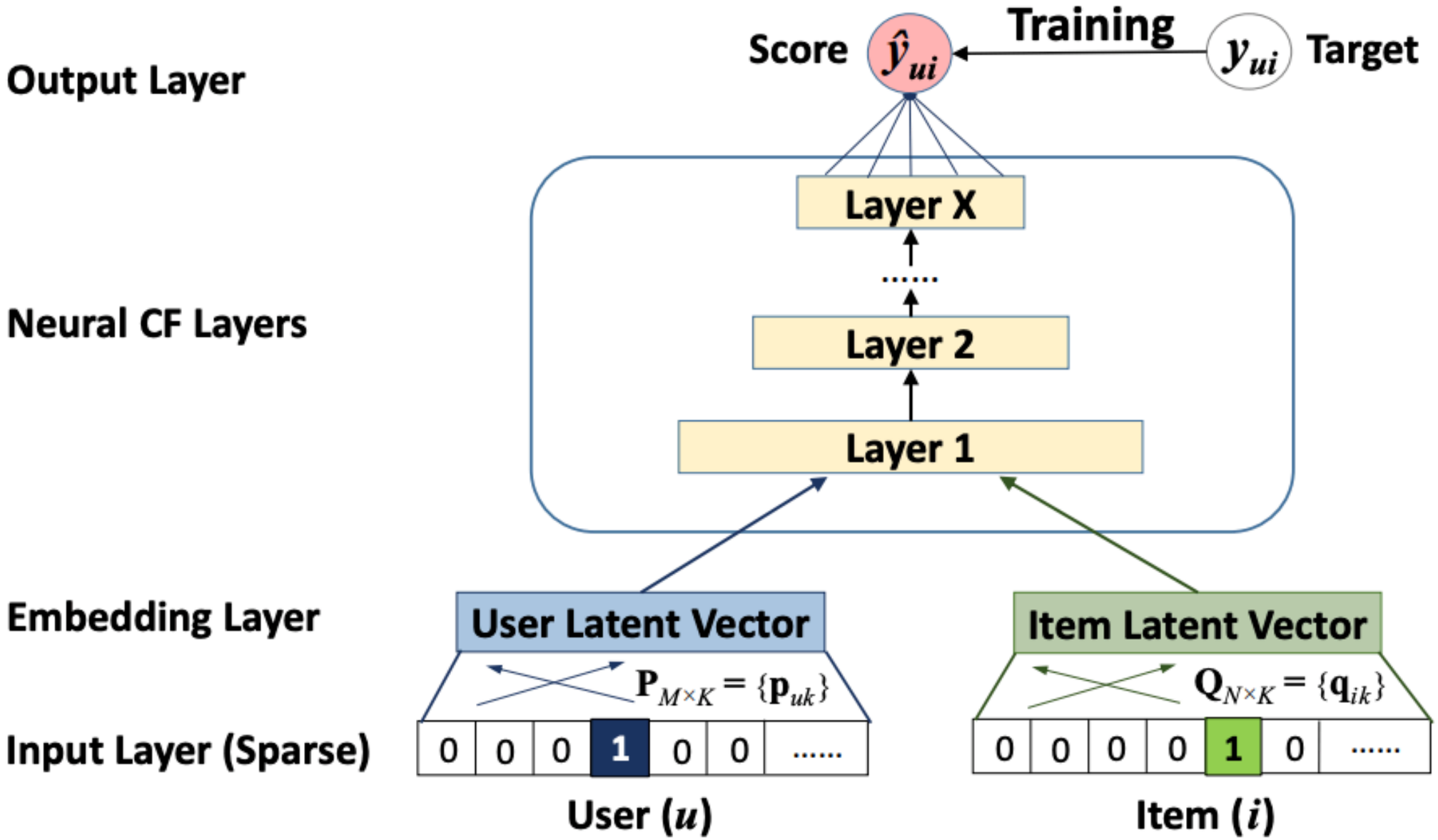



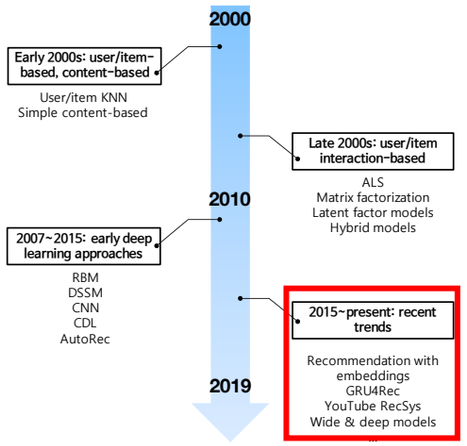
# Neural Collaborative Filtering




MF의 inner-product의 선형성으로 인한 한계를 극복하기 위해 제안됨
Element-wise product, concatenation 등을 활용함

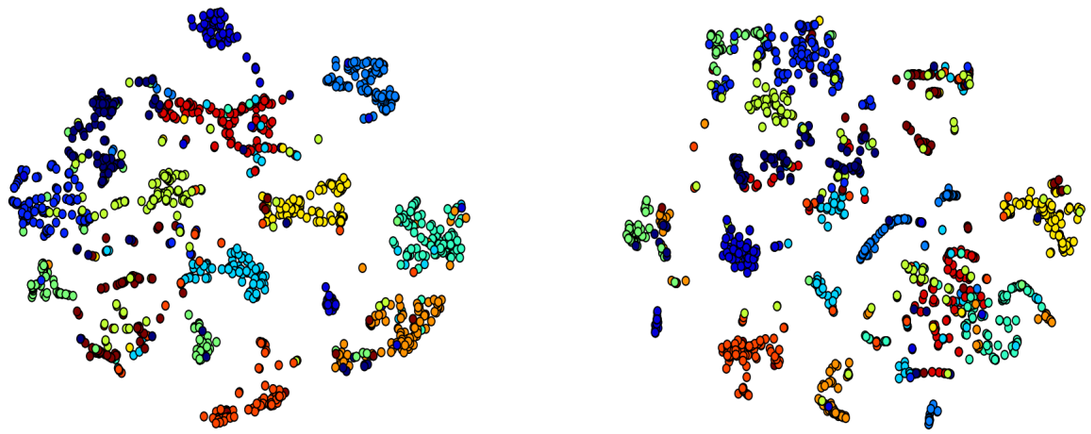


# Embedding, 그리고 추천

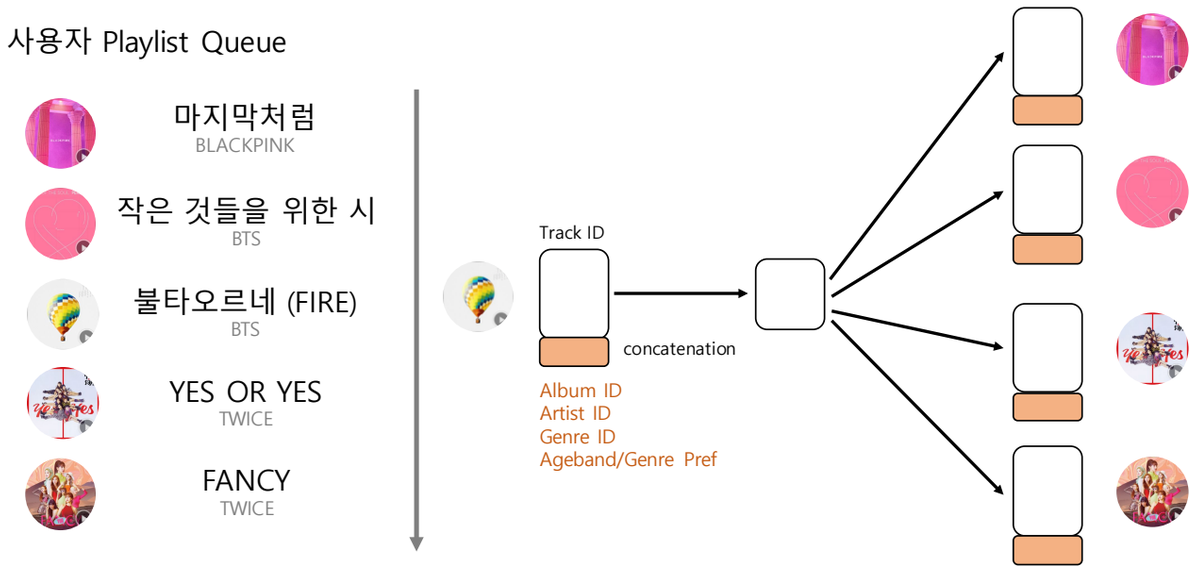



보다 풍부한 representation을 고려하기 위해 다양한 종류의 Embedding 메커니즘을 추천시스템에 적용하고 있음

- Item2Vec (Microsoft)
  - Early attempt to apply Word2Vec to recommendation
- Song2Vec for playlist (Spotify)
  - Track/Artist IDs as words
- Meta-Prod2Vec (Criteo)
  - Personalized online advertising
  - Predicts metadata (context) as well as products

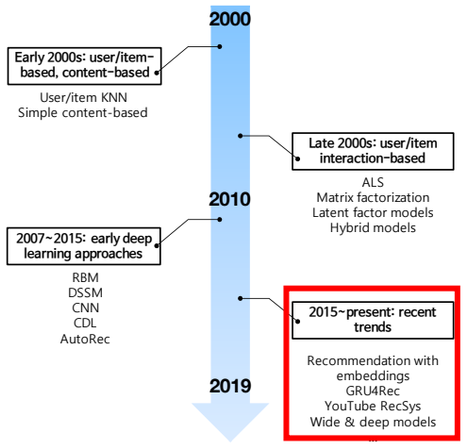


Item2Vec vs. SVD



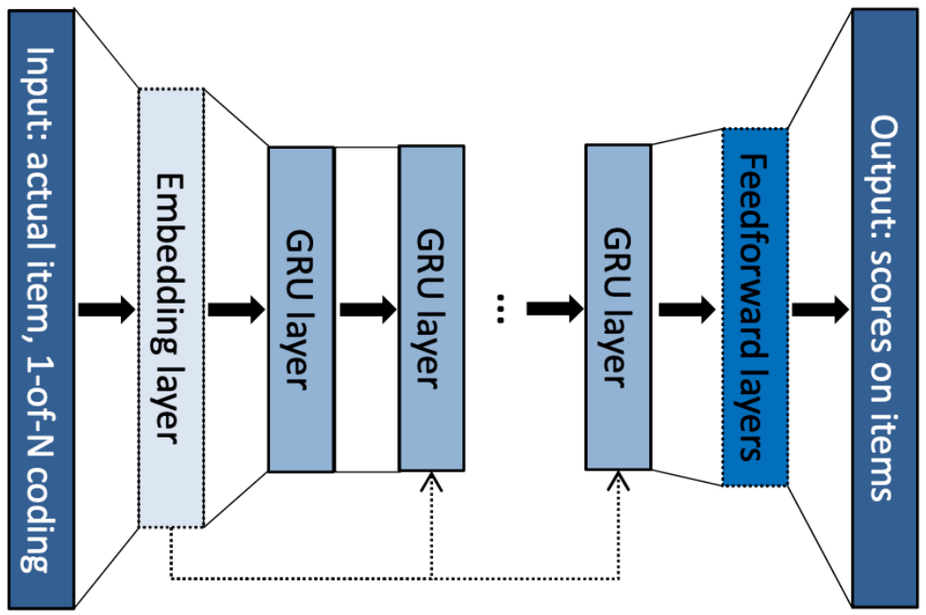

Song2Vec

# RNN for Sequential Recommendation

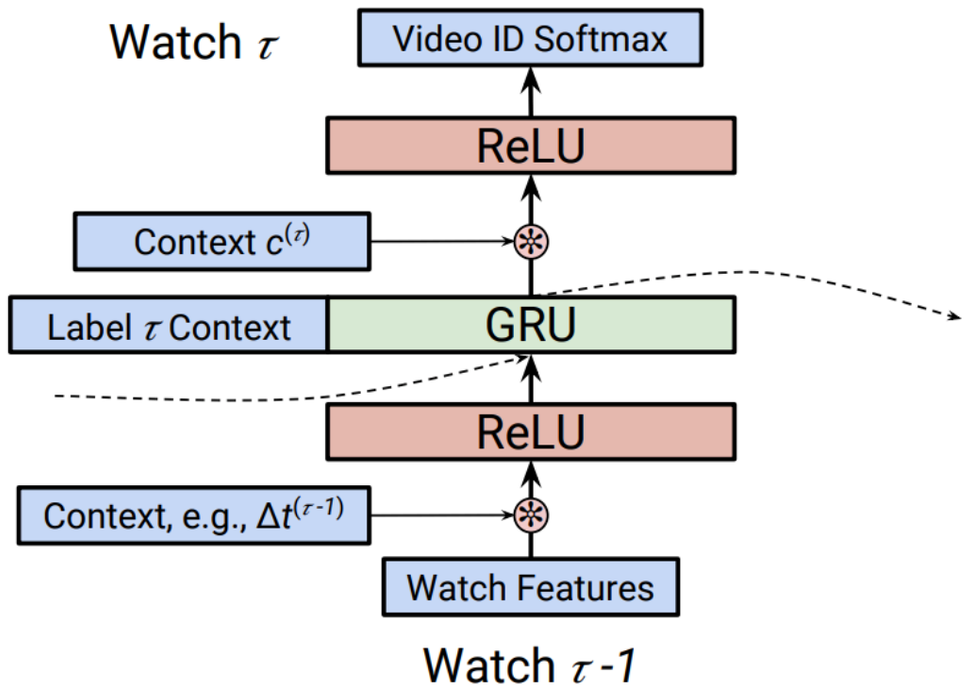



보다 현실적인 추천 및 동적 요소를 반영하기 위해 RNN을 활용한 sequential recommendation 연구도 계속되고 있음

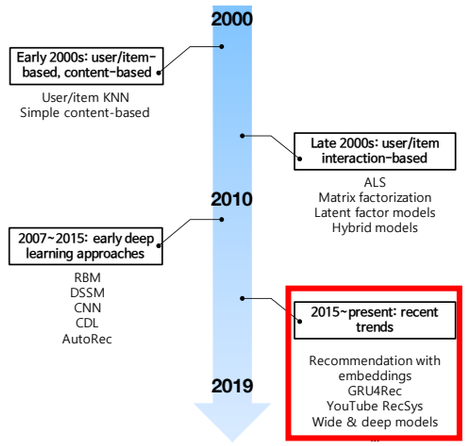
- GRU4Rec
  - One-hot encoding
  - Session-parallel mini-batch
  - w/o user identifier




- Latent-Cross (Google)
  - Element-wise product (instead of concatenation) for multiplicative relations
  - RNN-based recommender for YouTube

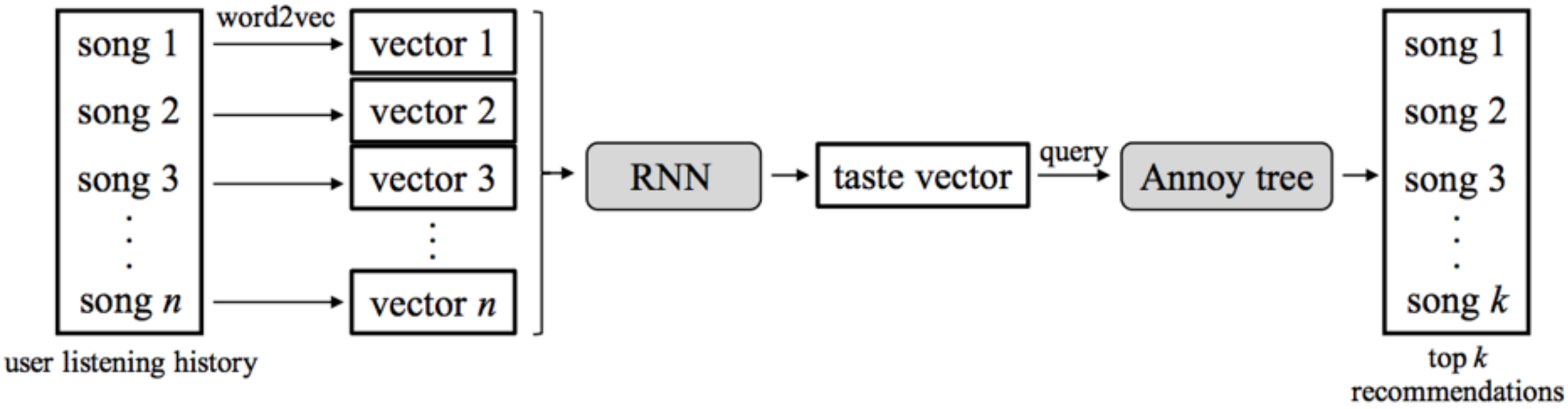


# RNN for Sequential Recommendation

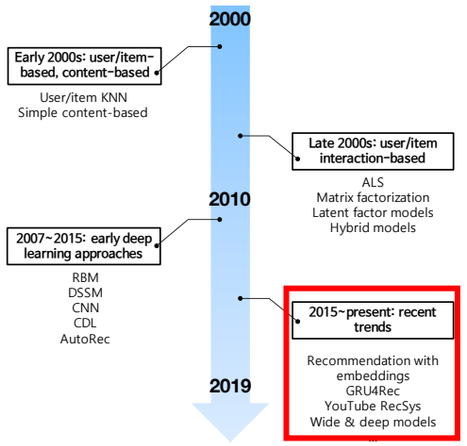



보다 현실적인 추천 및 동적 요소를 반영하기 위해 RNN을 활용한 sequential recommendation 연구도 계속되고 있음

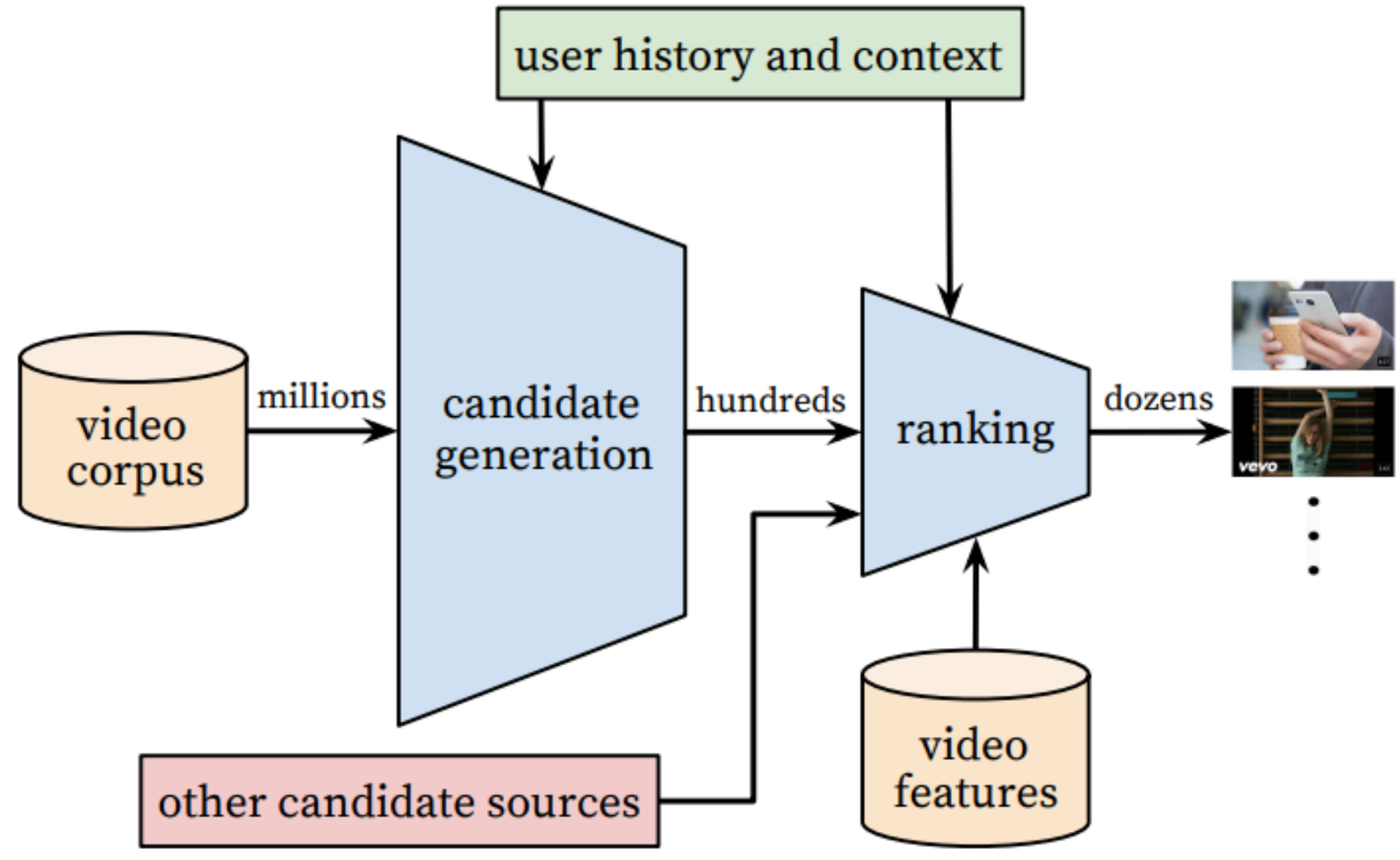
- Spotify's large-scale music discovery model
  - User representation based on song2vec
- Static CF tends to over-summarize the entire listening records.
  - Limitation of modeling temporal dynamics
- Dynamic CF based on RNN
  - Modeling a time-drifting effect of user's taste



# YouTube Recommender

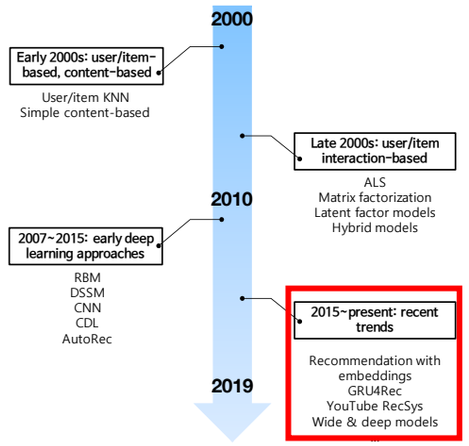



실제 industrial production-level의 성공적인 적용을 위해 two-stage 구조를 활용하는 등 다양한 고민을 한 흔적이 엿보임

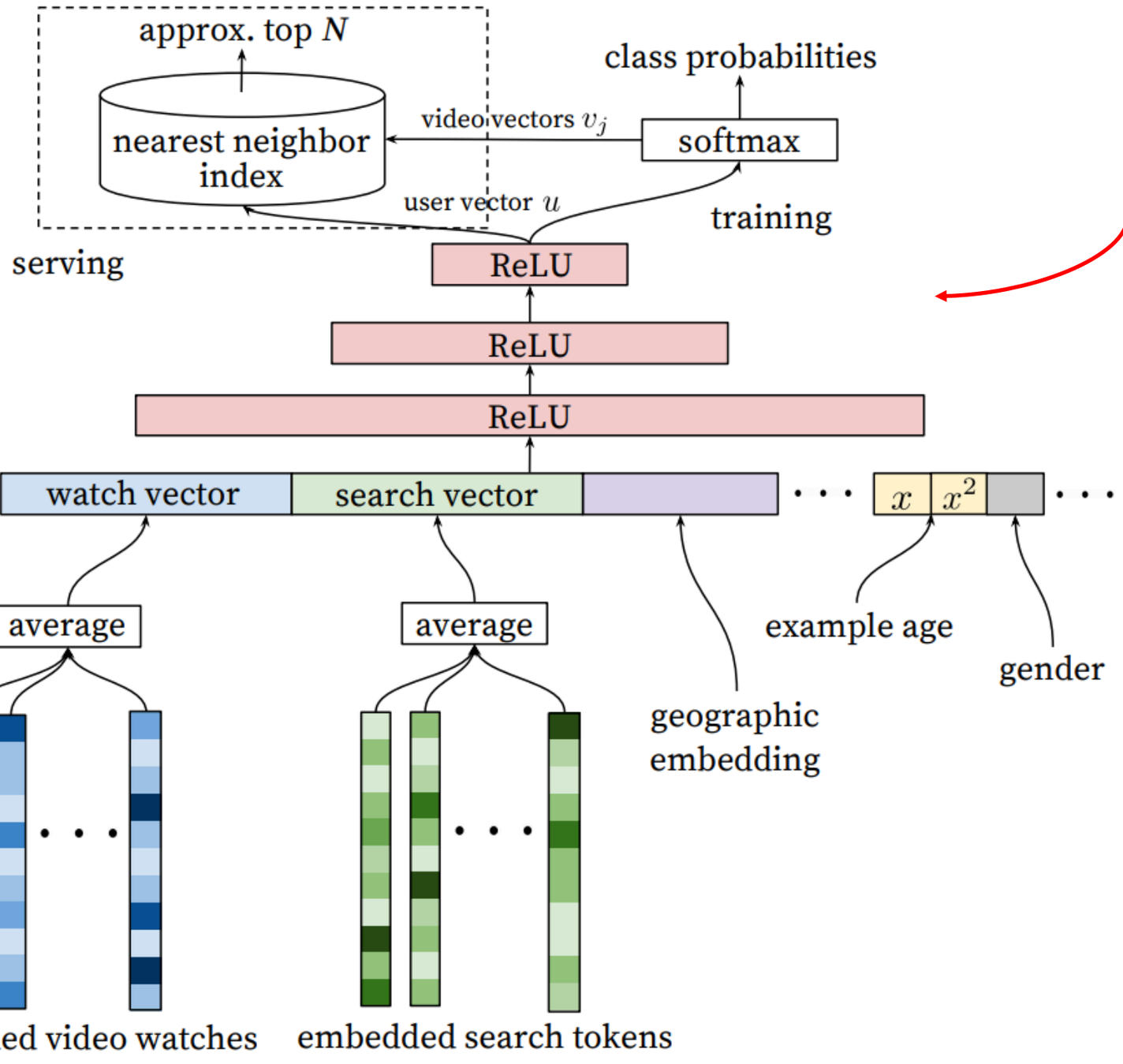


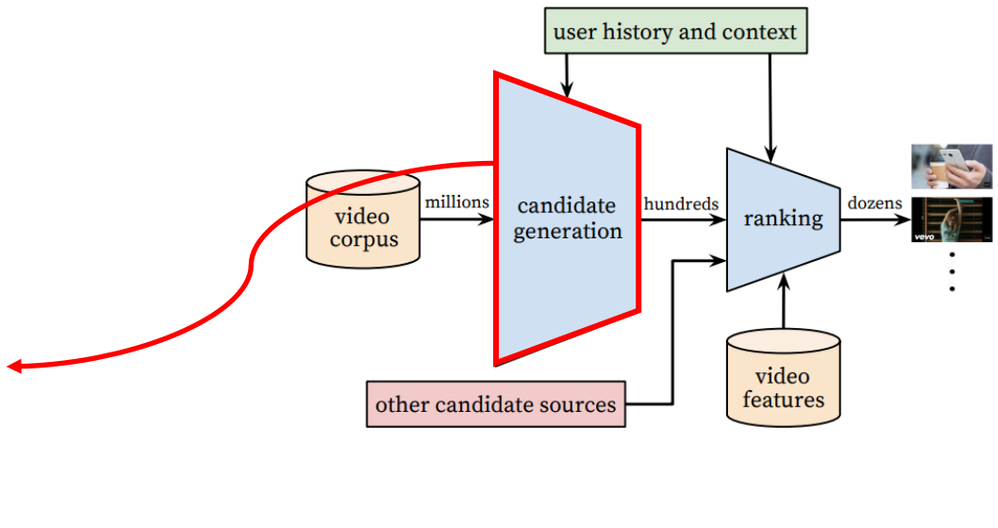

- Two-stage cascade (candidate generation and ranking)
- Use of various context

# YouTube Recommender

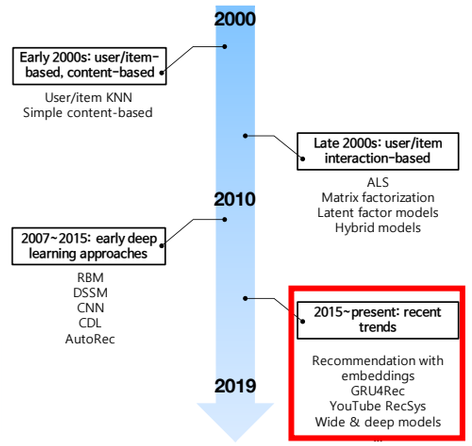



## Candidate generation network

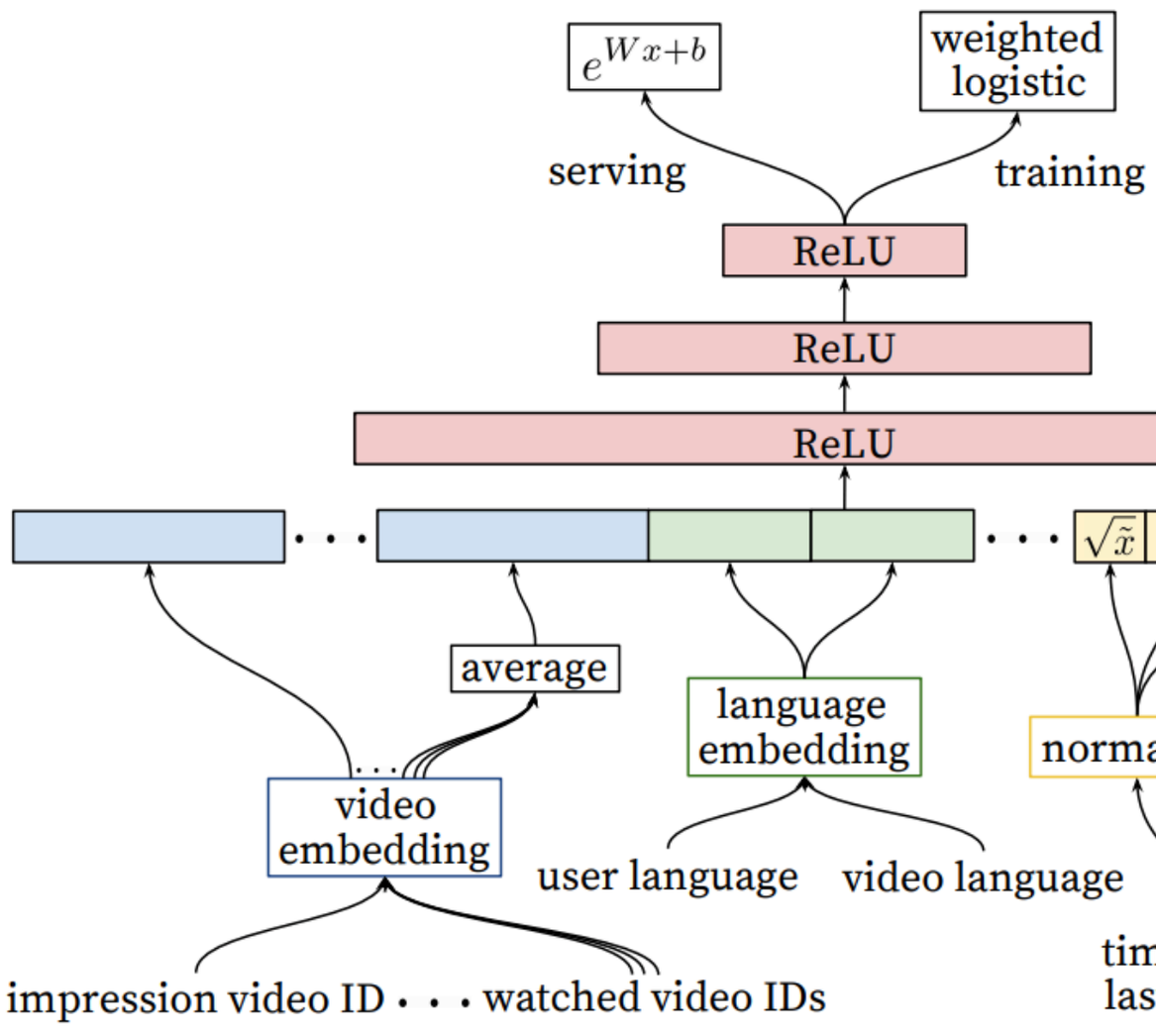


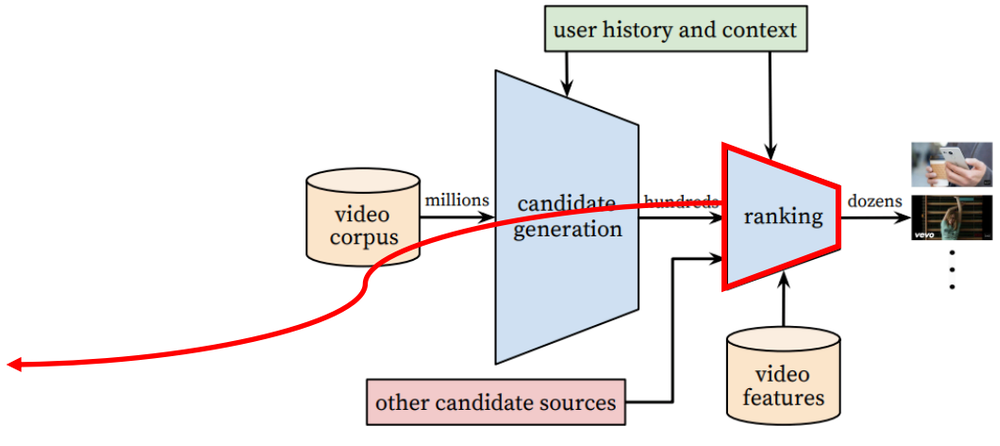



# YouTube Recommender

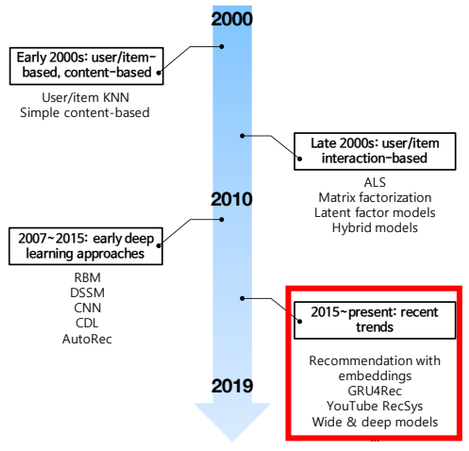



## Ranking network



# RecSys Challenge 2018

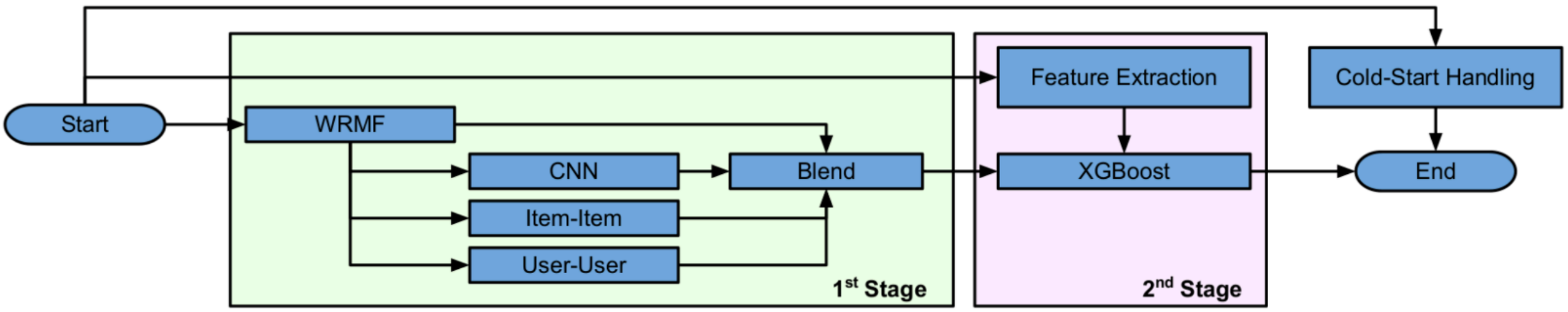



Spotify's automatic playlist continuation을 주제로 한 2018년 competition에서 Two-stage 구조 및 NN, Learning-to-rank 등이 강점을 보임

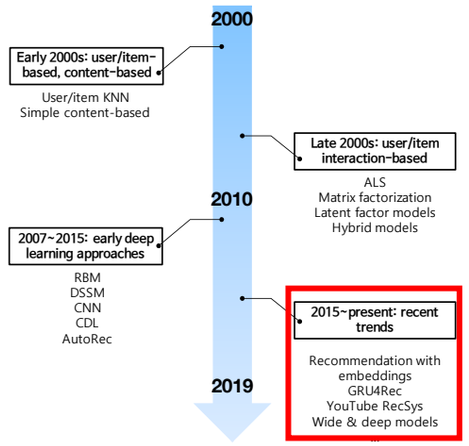
| rank | team | two stage | MF | NN | LTR |
|---|---|---|---|---|---|
| 1 | vl6 | ✓ | ✓ | ✓ | ✓ |
| 2 | hello world! | ✗ | ✗ | ✓ | ✗ |
| 3 | Avito | ✓ | ✓ | ✗ | ✓ |
| 4 | Creamy Fireflies | ✗ | ✗ | ✗ | ✗ |
| 6 | HAIR | ✓ | ✗ | ✓ | ✓ |
| 7 | KAENEN | ✗ | ✓ | ✗ | ✗ |
| 7 | BachPropagate | ✓ | ✗ | ✓ | ✓ |
| 9 | Definitive Turtles | ✗ | ✗ | ✗ | ✗ |
| 10 | IN3PD | ✓ | ✓ | ✗ | ✗ |

100만 가량의 user-created playlists
Playlist의 일부가 주어졌을 때 이를 완성하는 track을 예상하는 문제

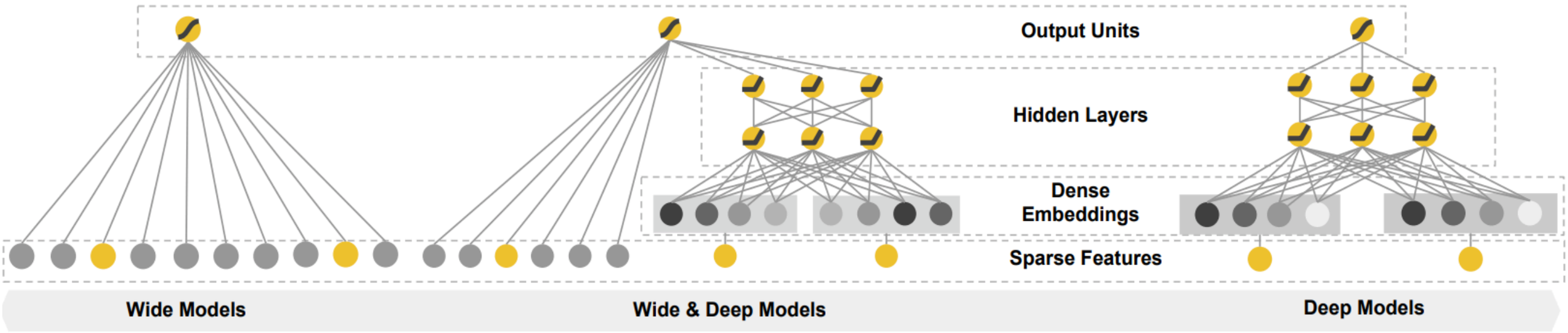
791 participants from over 20 countries &
410 teams with 1,497 submissions



# Wide & Deep Model (Google Play Store)

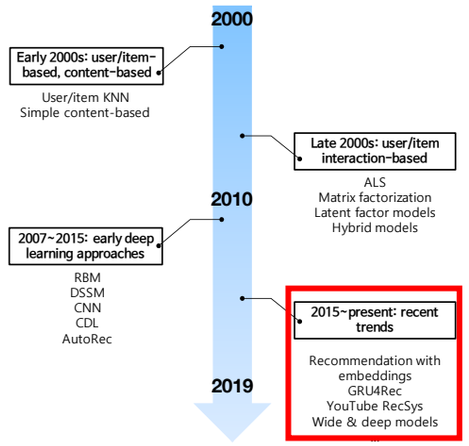



Logistic Regression (wide) + DNN (deep) 구조를 통해 memorization과 generalization의 적절한 조합을 제공할 수 있음

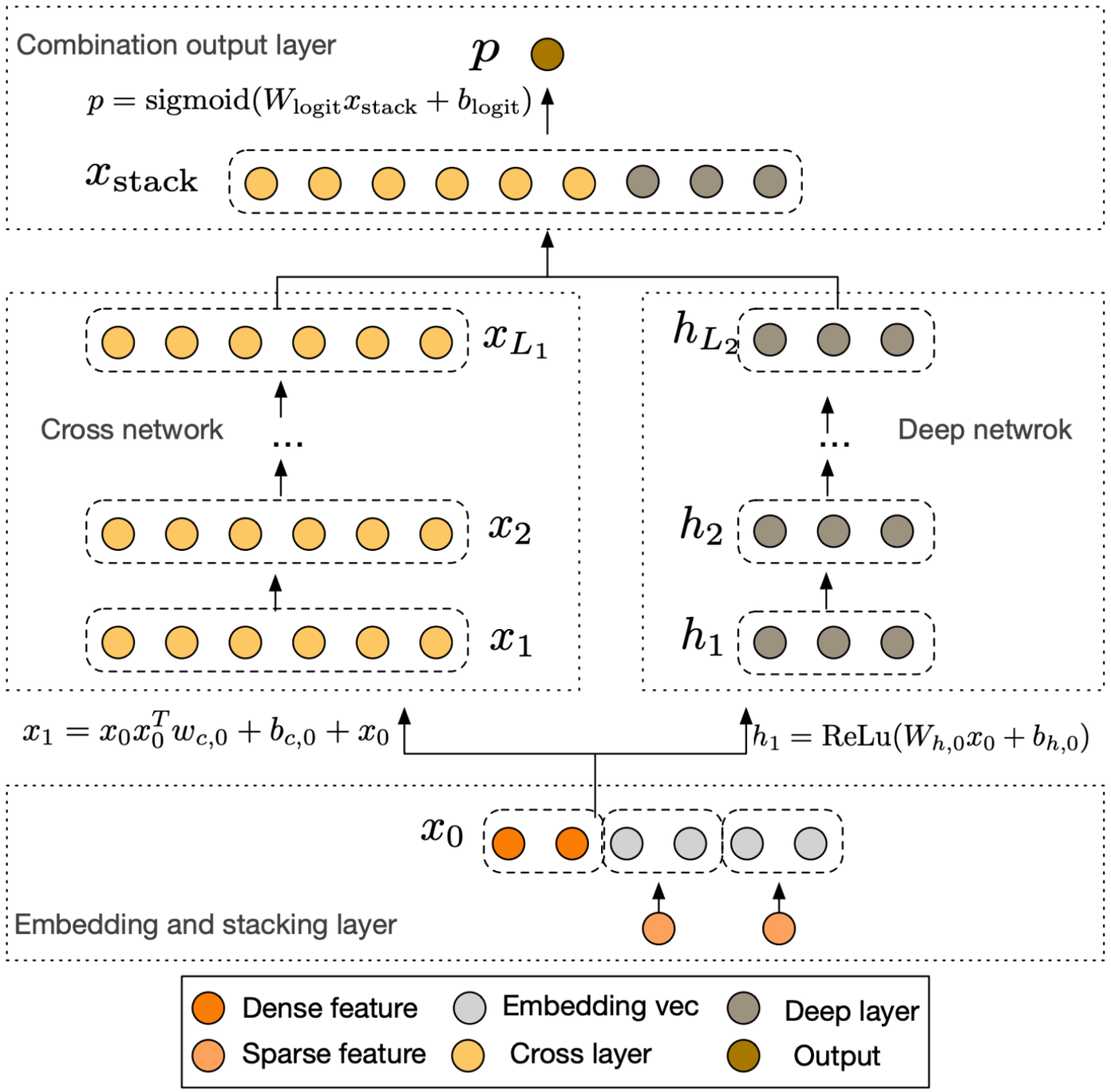



- Data in CF are extremely sparse
  - Cold-start problems
  - Risk of under-fitting or over-fitting
- To overcome the issue:
  - Memorization (LR, wide) + Generalization (DNN, deep)
- 적절한 wide feature combination 을 찾는 것은 여전히 이슈

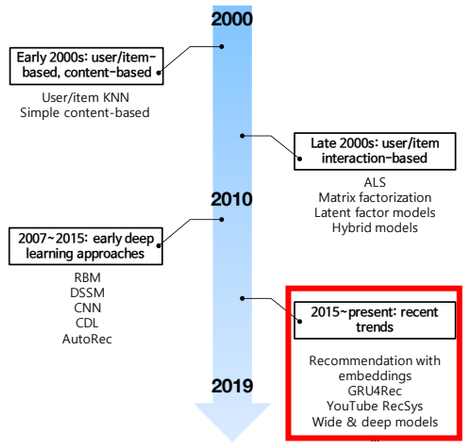
# Deep & Cross Network (DCN)




Google의 production 광고 시스템에 적용됨 (2017. 08)
Feature들 간의 pairwise interaction weight를 cross network를 통해 계산함



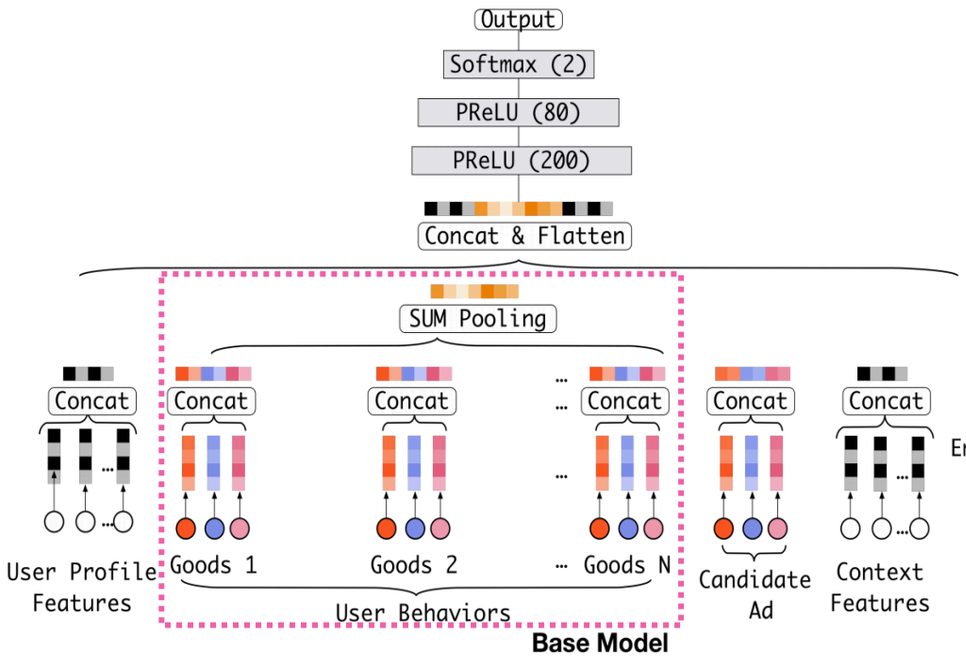
# Deep Interest Network (DIN)

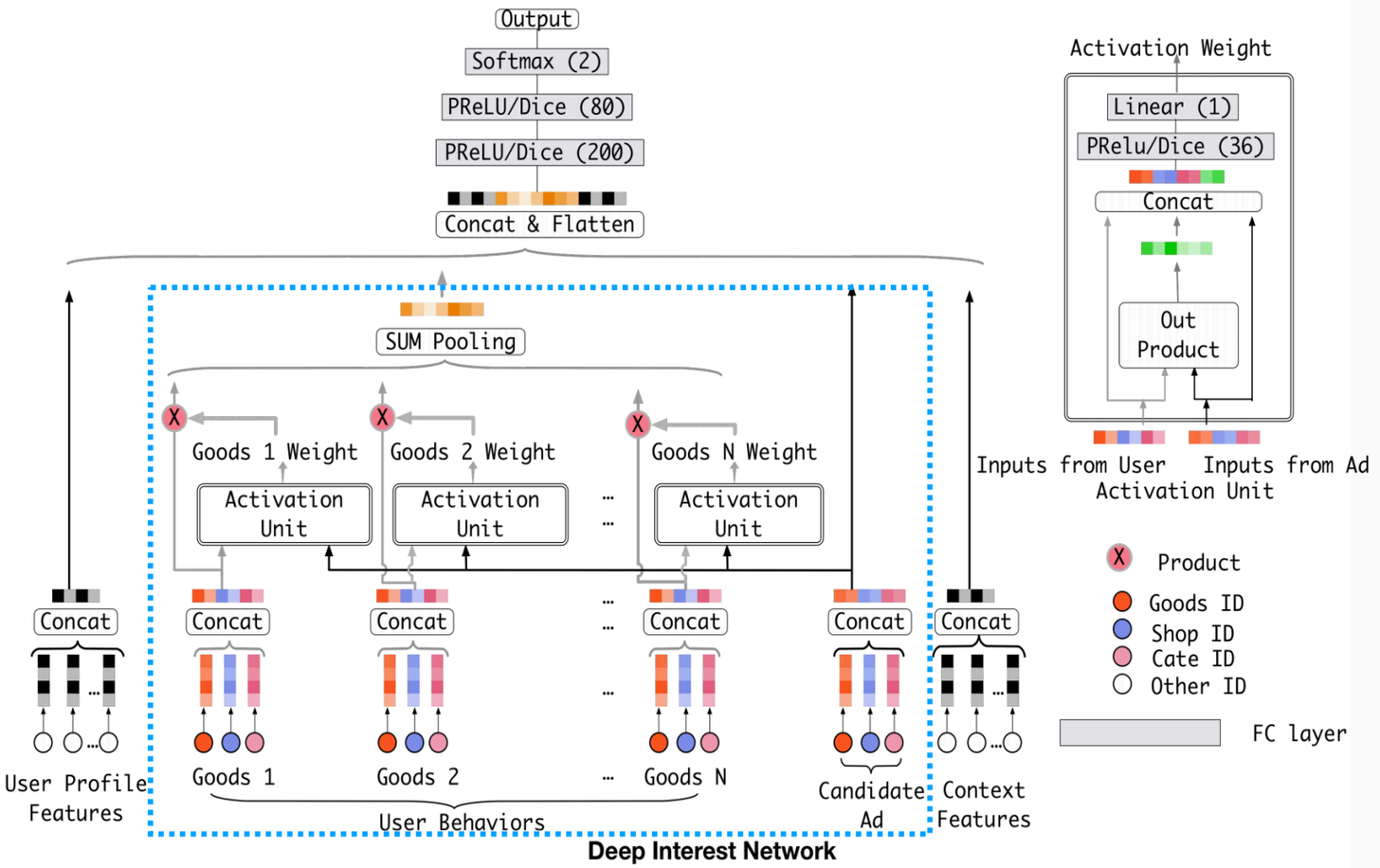



## CTR Prediction in Alibaba Group (2017 arXiv, 2018.09 KDD)

기존: Candidate Ad에 관계 없이 하나의 global user representation vector를 가짐

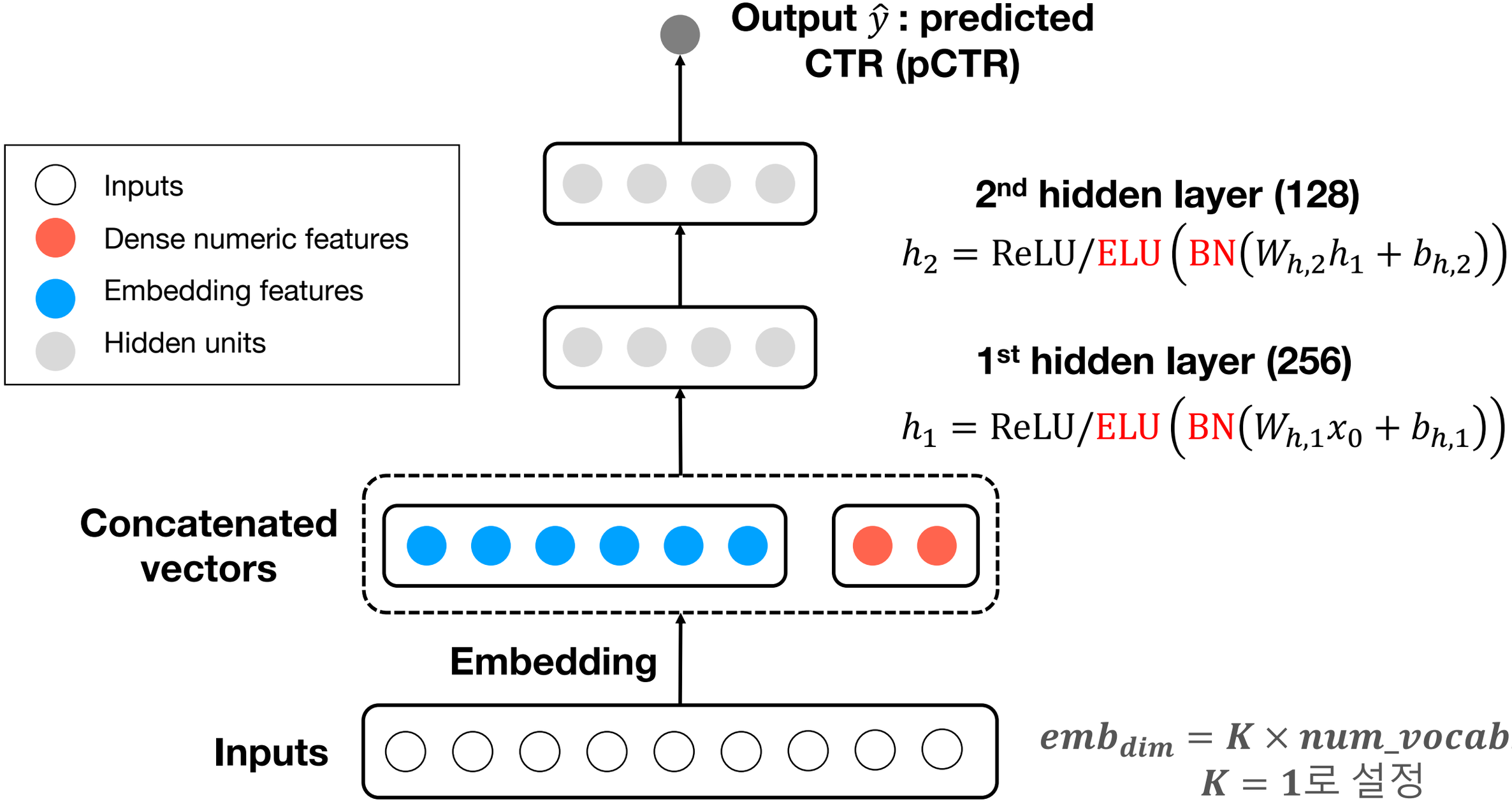
DIN: ad와 user behavior간의 interaction을 고려하여, 서로 다른 ad에 따라 user behavior 별 중요도가 달라짐. 결과적으로 보다 정밀한 user representation을 얻을 수 있음



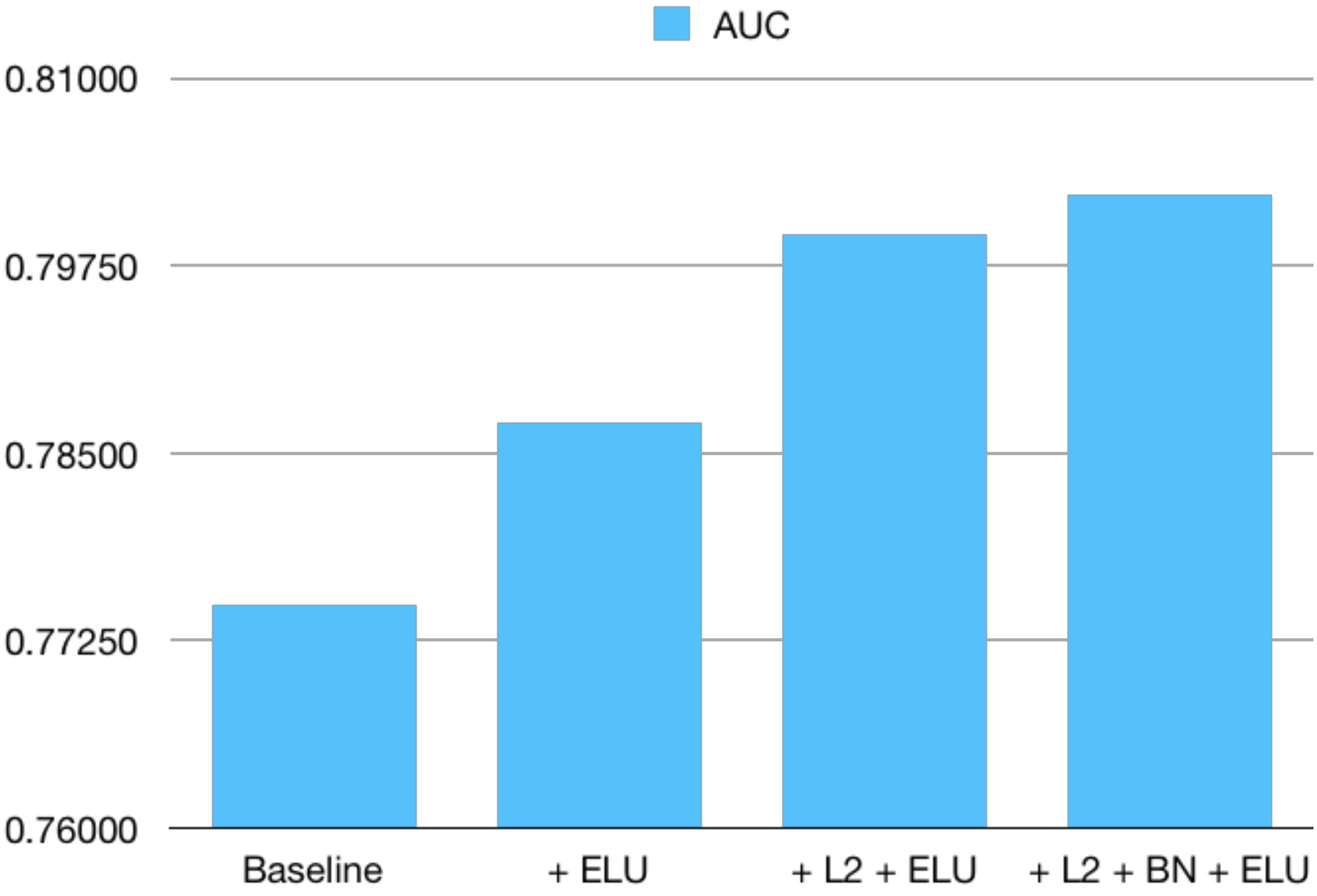



# Experiments: Regularization Effects

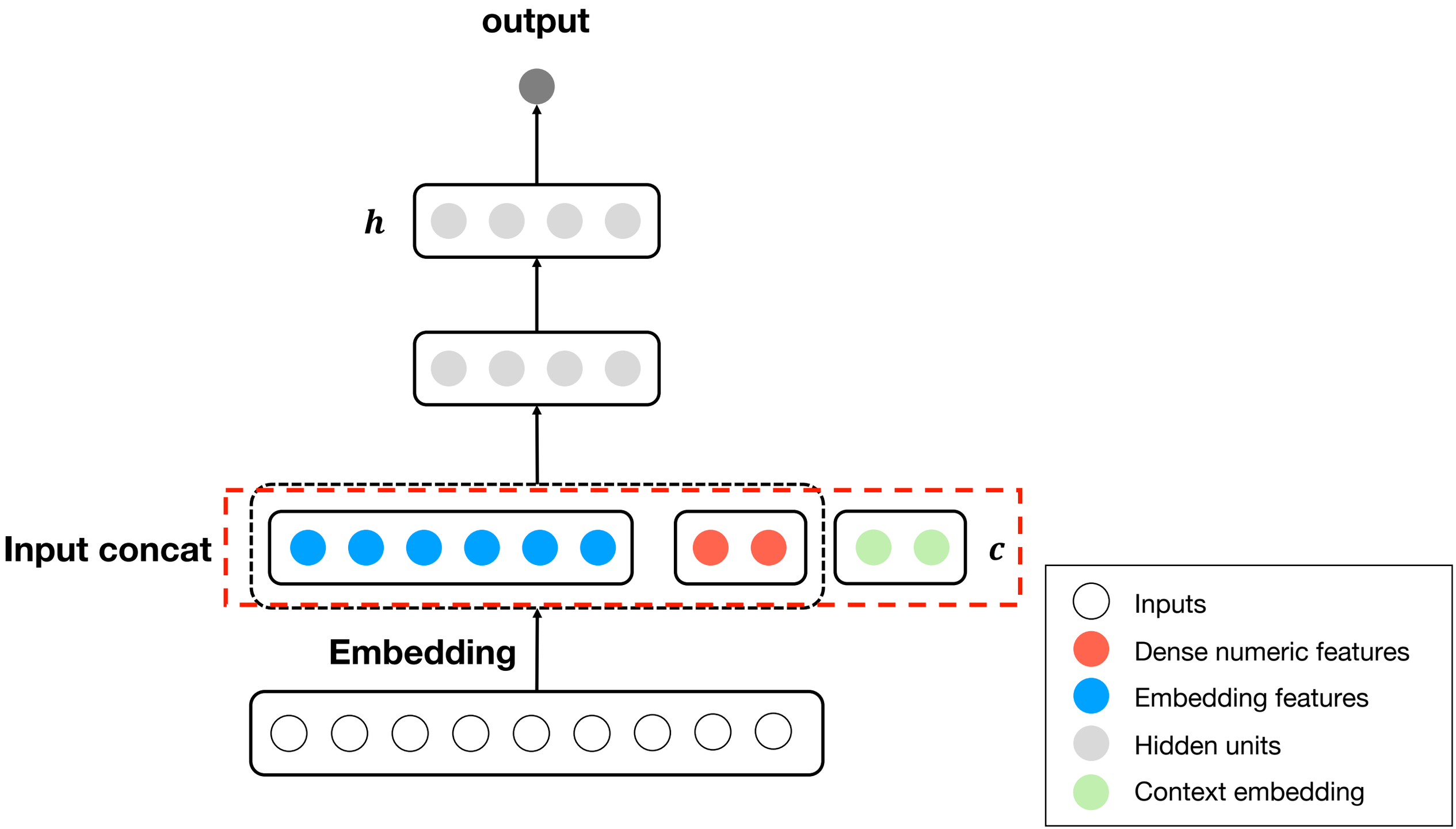
$$Loss = -ylog(\hat{y}) - (1-y)\log(1-\hat{y}) + \lambda \|W\|_2^2$$



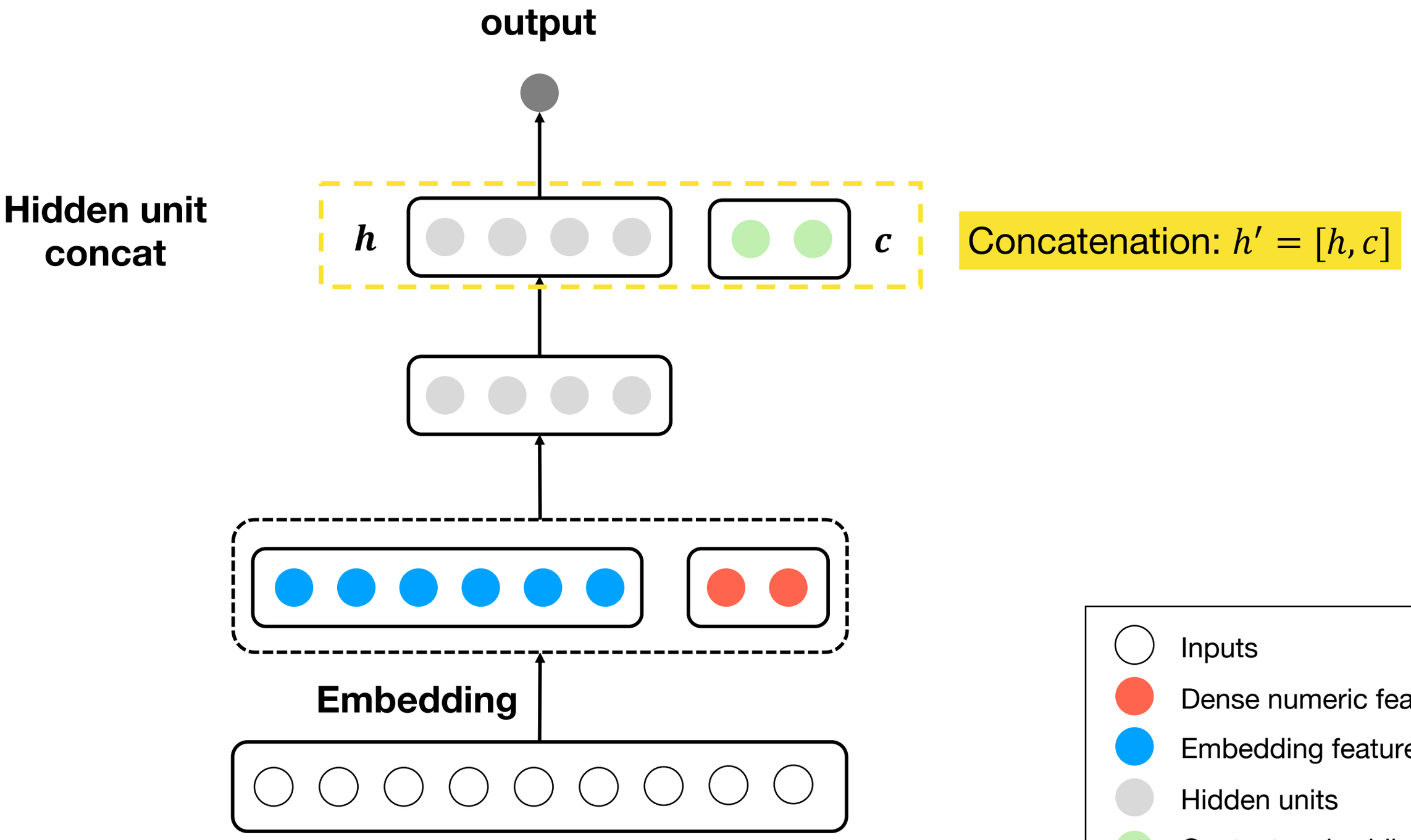
# Experiments: Regularization Effects



# How to apply context as features

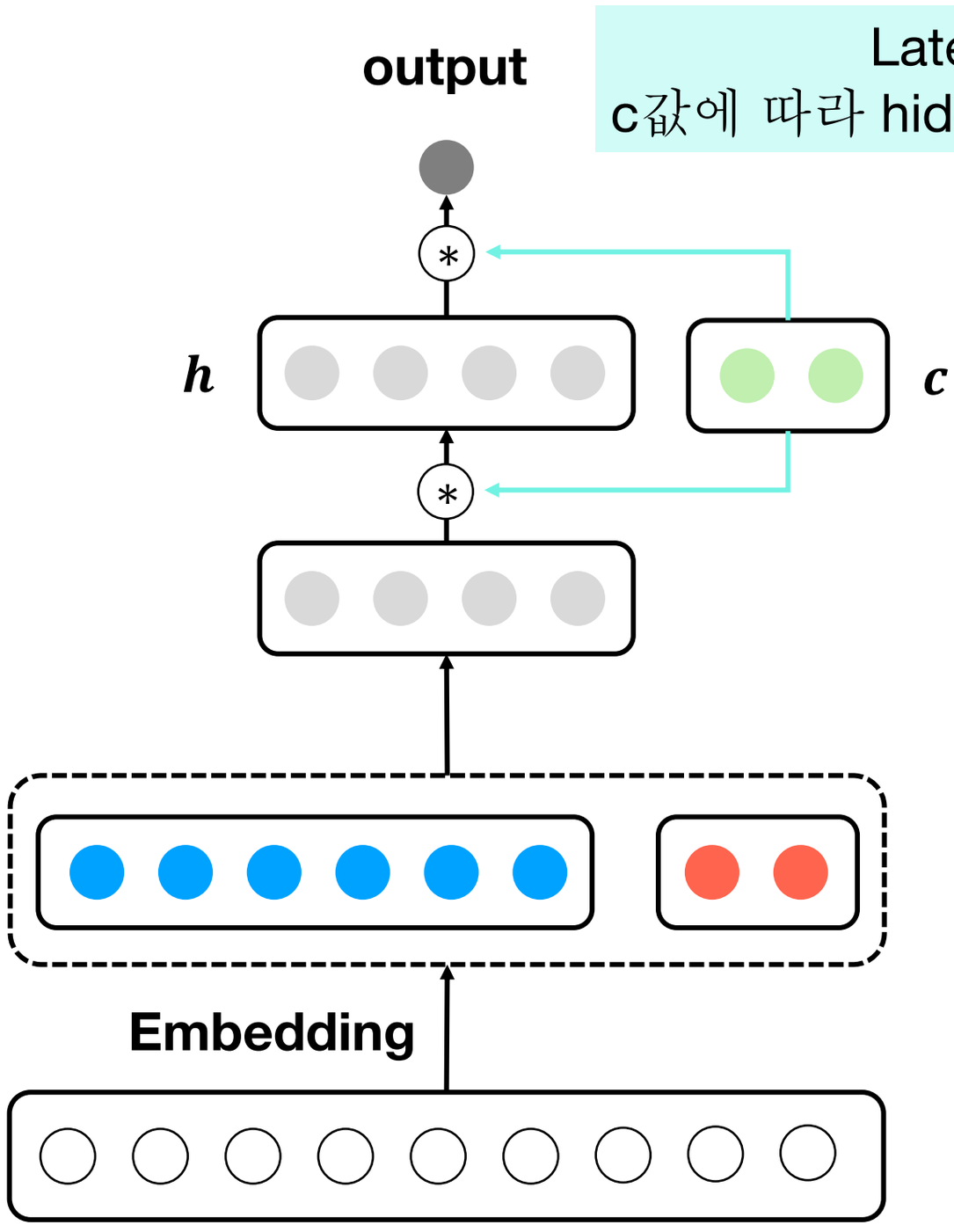


# How to apply context as features



# How to apply context as features

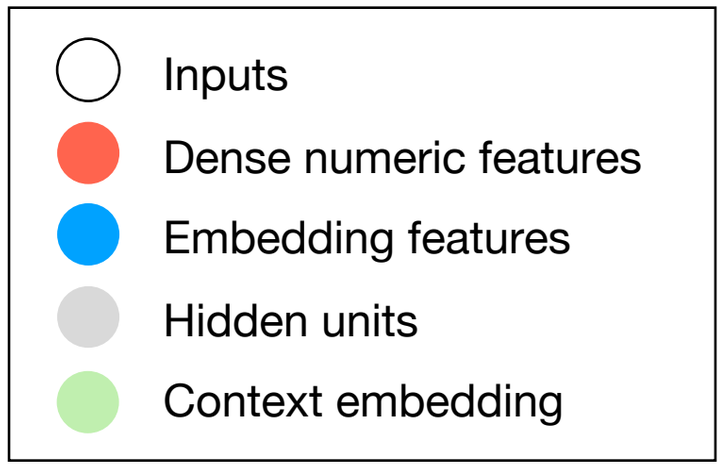
Latent Cross: $h' = (1 + c) * h$
c값에 따라 hidden unit의 attention or masking 역할



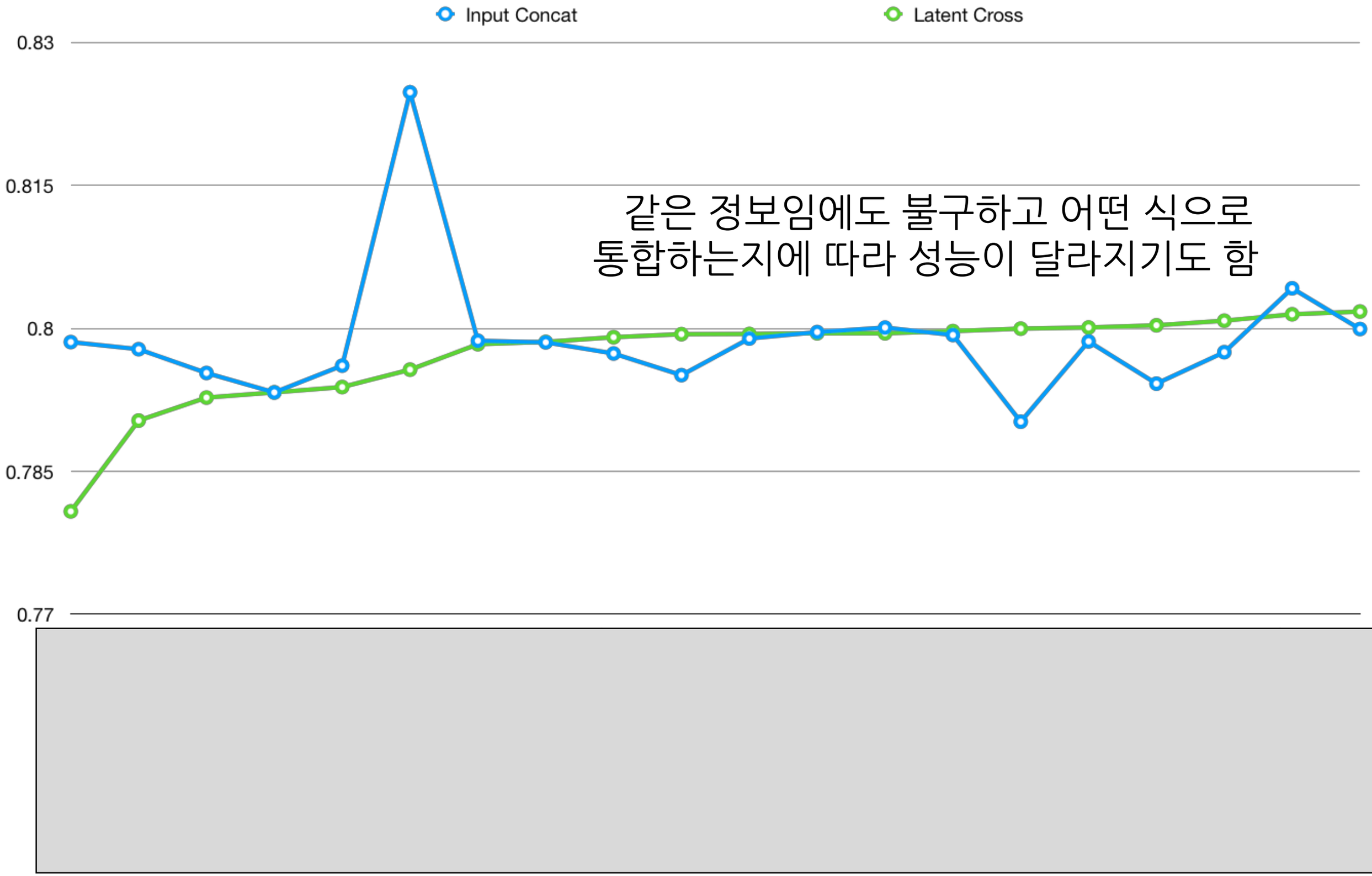

**Latent Cross:**
**Element-wise product**
**on hidden units**

Element-wise product
(Hadamard product)

$$\boldsymbol{a} * \boldsymbol{b} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{M}} a_i \cdot b_i$$



# Input Concat vs. Latent Cross



# History Summary

- From shallow to deep
  - ALS, MF,LR → Neural networks
  - Memorized features → Generalized features
  - Linear transformation, limited modeling capacity: inferior prediction power
  - Encoding of nonlinear rich user/item interaction: superior prediction power
- From static to dynamic
  - RNN for sequential recommendation
  - To reflect time-drifting user behaviors
- Use of various context
  - Demographics, Time, Place, Occasion (TPO)
- Industrial use-case
  - Designing the overall architecture: art or technique, rather than science
  - Tuning is the key to success, but extremely time-consuming
  - Online A/B Test, rather than offline evaluation

# Future Research Directions

- Recent deep learning advances
  - Attention mechanisms
  - Adversarial learning

- Reinforcement learning
  - Exploration and exploitation (Multi-armed bandit)
  - Near real-time reflection of user feedback

- Explainable/interpretable recommendation
  - Natural language generation of the reason why
  - Causal inference for recommendation

- Evaluation
  - Minimize algorithmic bias
  - Offline A/B Testing

# THANK YOU

**Contact**
이원성: **wonsung.lee@sk.com**