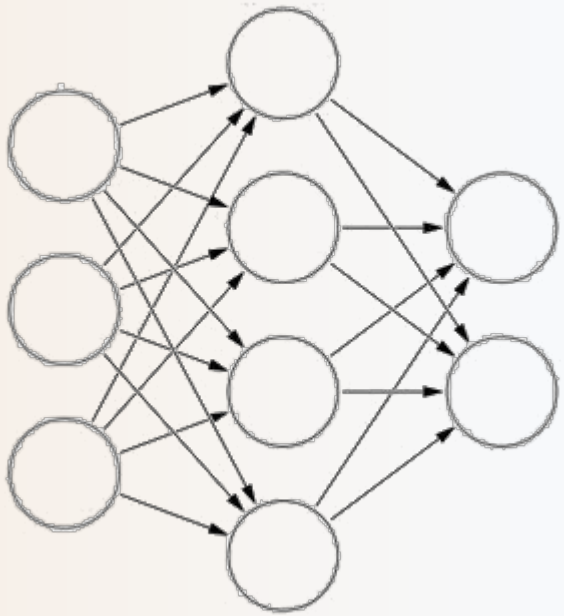
Deep Neural Networks for YouTube Recommendations

Paul Covington, Jay Adams, Emre Sargin

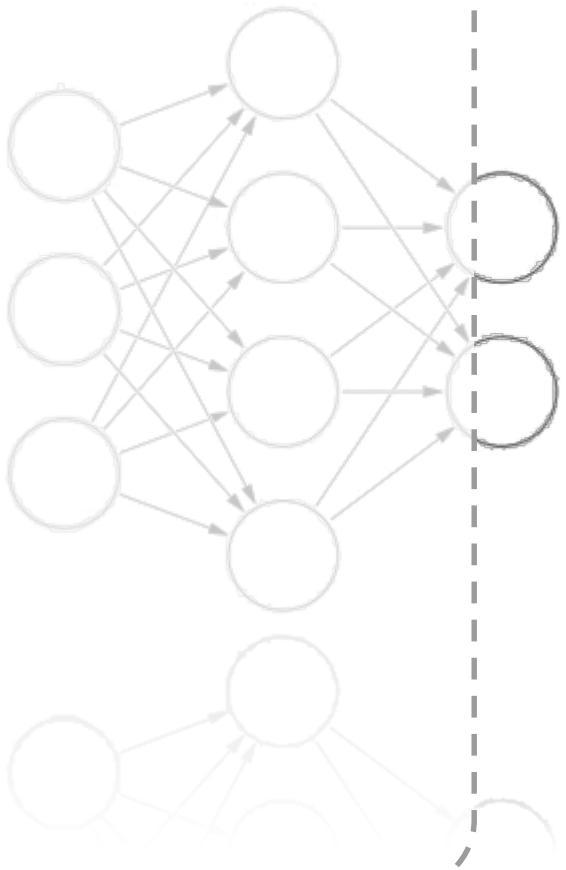


Korea University
INI Lab
윤주성

Research & Development

Index

- Overview
- Introduction
- Model
- Conclusion
- QnA



Overview

연구의 내용

Google Brain
TensorFlow
One billion parameters
Hundreds of billions of examples

YouTube 라는 스케일이 크고, 새로운 컨텐츠가 매순간 올라오는 플랫폼에
Deep Neural Network를 적용해서 추천시스템을 구축

Two-stage 접근 방법
[1] Candidate Generation
[2] Ranking

+ 다양한 Feature들 사용

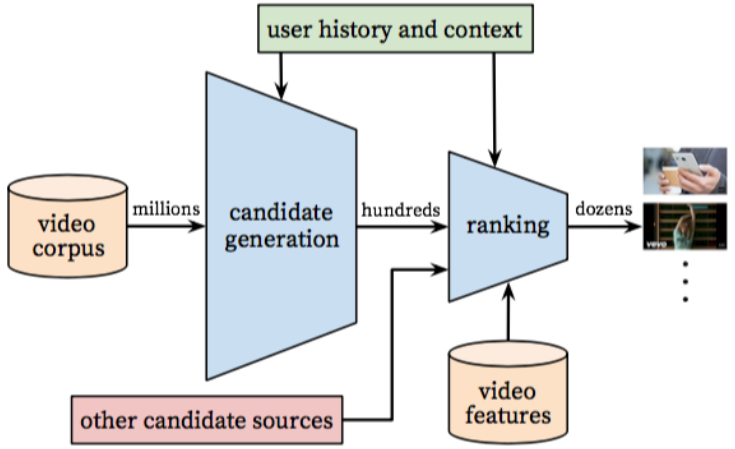


Figure 2: Recommendation system architecture demonstrating the "funnel" where candidate videos are retrieved and ranked before presenting only a few to the user.

Overview

연구의 내용

Google Brain
TensorFlow
One billion parameters
Hundreds of billions of examples

YouTube 라는 스케일이 크고, 새로운 컨텐츠가 매순간 올라오는 플랫폼에
Deep Neural Network를 적용해서 추천시스템을 구축

Two-stage 접근 방법
[1] Candidate Generation
[2] Ranking

+ 다양한 Feature들 사용

Deep neural networks for youtube recommendations
P Covington, J Adams, E Sargin - … of the 10th ACM Conference on …, 2016 - dl.acm.org
Abstract YouTube represents one of the largest scale and most sophisticated industrial recommendation systems in existence. In this paper, we describe the system at a high level and focus on the dramatic performance improvements brought by deep learning. The paper is split according to the classic two-stage information retrieval dichotomy: first, we detail a deep candidate generation model and then describe a separate deep ranking model. We also provide practical lessons and insights derived from designing, iterating and maintaining
☆ 99 110회 인용 관련 학술자료 전체 6개의 버전 »

Introduction

연구의 내용

유투브 추천시스템? Extremely Challenging from three major perspectives

[1] Scale
Billion user에게 Personalized content를 제공해야함
distributed learning algorithm & efficient serving system이 요구됨

[2] Freshness
newly uploaded content가 매우 많고. 기존 컨텐츠와 밸런스 있게 모델에 반영 해야함

[3] Noise
Historical user behavior는 sparse한 특성과 외부요인 때문에 예측하기 어려움
user satisfaction에 대한 ground truth를 얻기 어려움. 얻는 피드백은 노이즈가 있음
메타 데이터도 온톨로지가 없어서 구조화가 잘 안되어 있음.

Introduction

연구의 내용

본 논문의 제안 모델
Deep Neural Networks (Google Brain, TensorFlow, one billion parameters)

관련 연구들

Matrix factorization [19]
Recommending news [17], citations [8], review ratings [20]
Collaborative filtering (formulated as DNN [22], autoencoders[18])
Cross domain user modeling [5]
Music recommendation (content-based setting) [21]

Model

System Overview

Two neural networks:
[1] Candidate generation
[2] Ranking

Loss function:
Cross-entropy

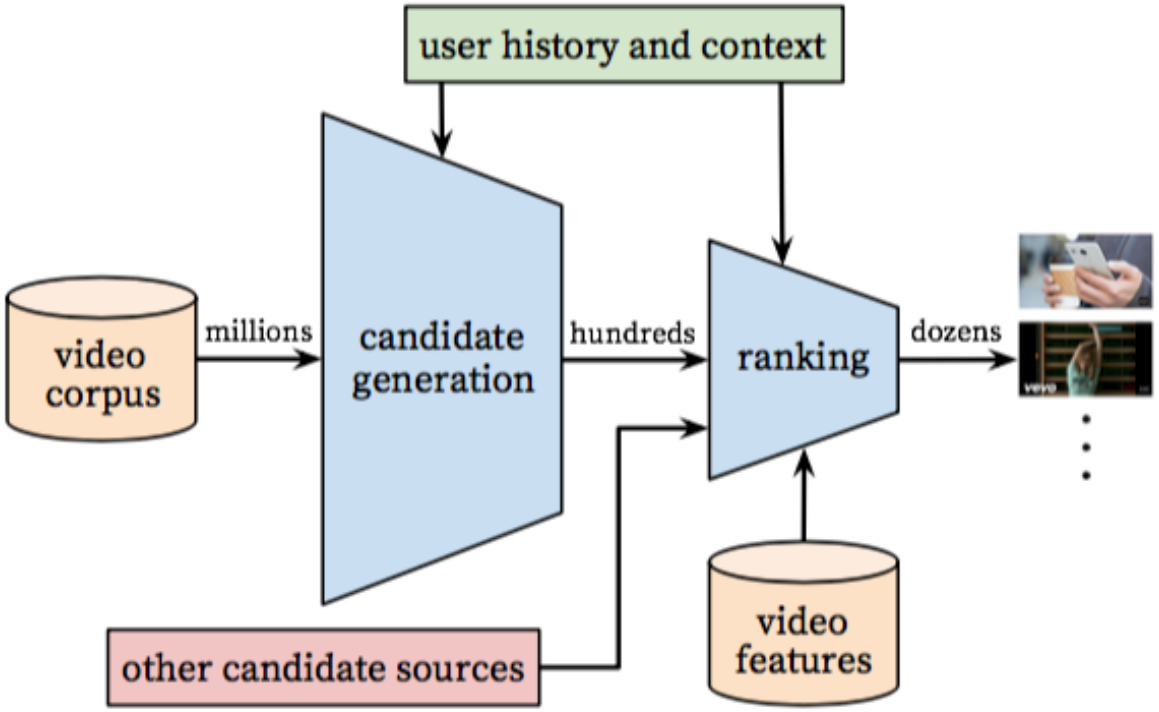


Figure 2: Recommendation system architecture demonstrating the "funnel" where candidate videos are retrieved and ranked before presenting only a few to the user.

Model

System Overview

[1] Candidate generation

-> provide broad personalization
Via collaborative filtering (CF)

User간의 유사도?

-> coarse feature로 표현함
Ex) IDs of video watches,
search query tokens
demographics

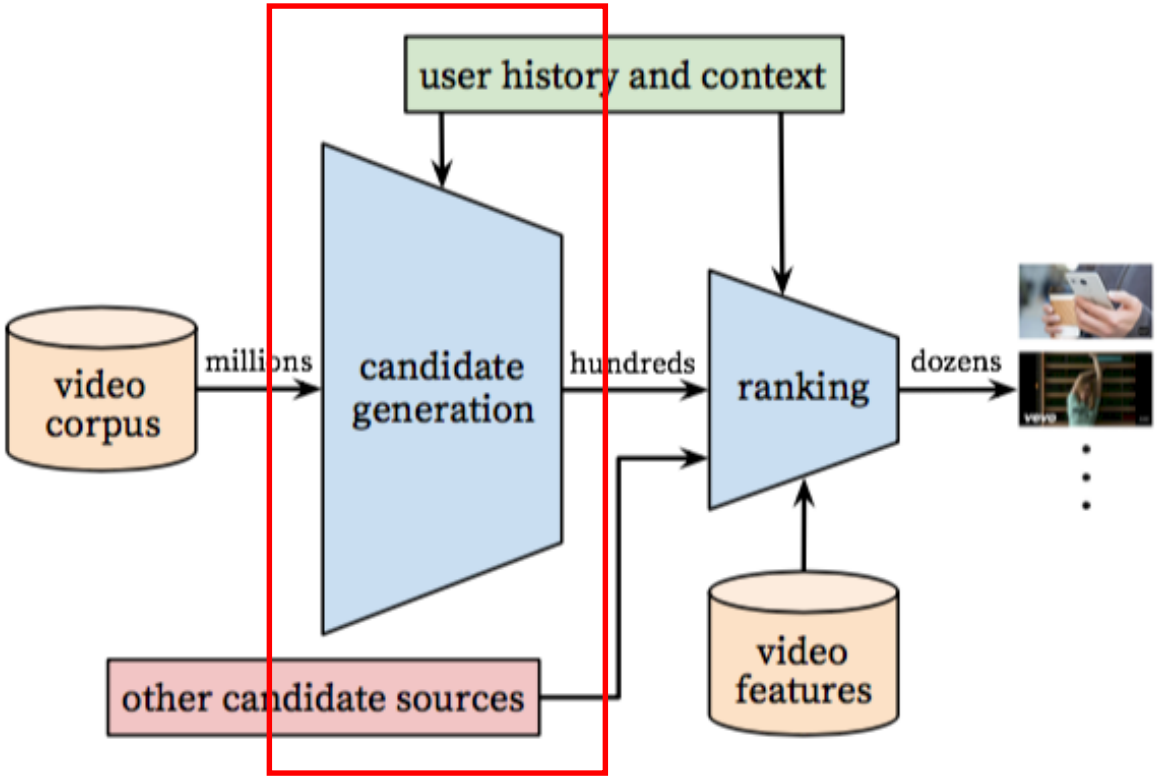


Figure 2: Recommendation system architecture demonstrating the "funnel" where candidate videos are retrieved and ranked before presenting only a few to the user.

Model

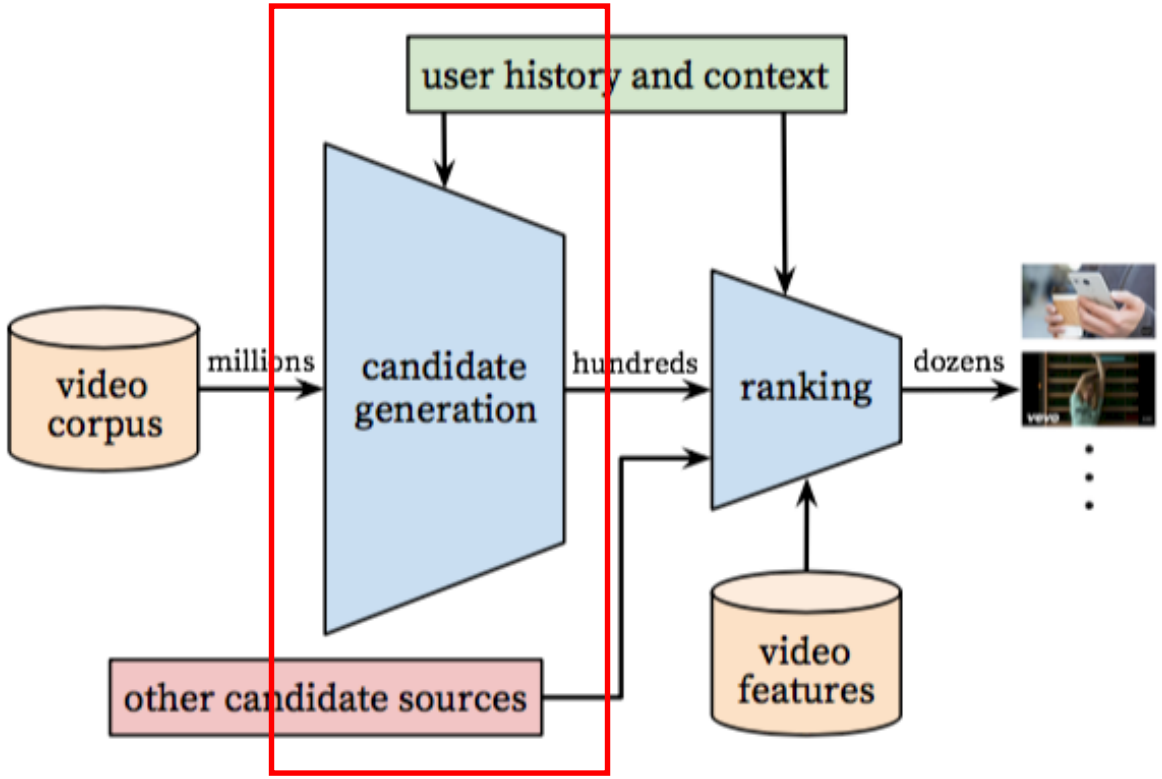
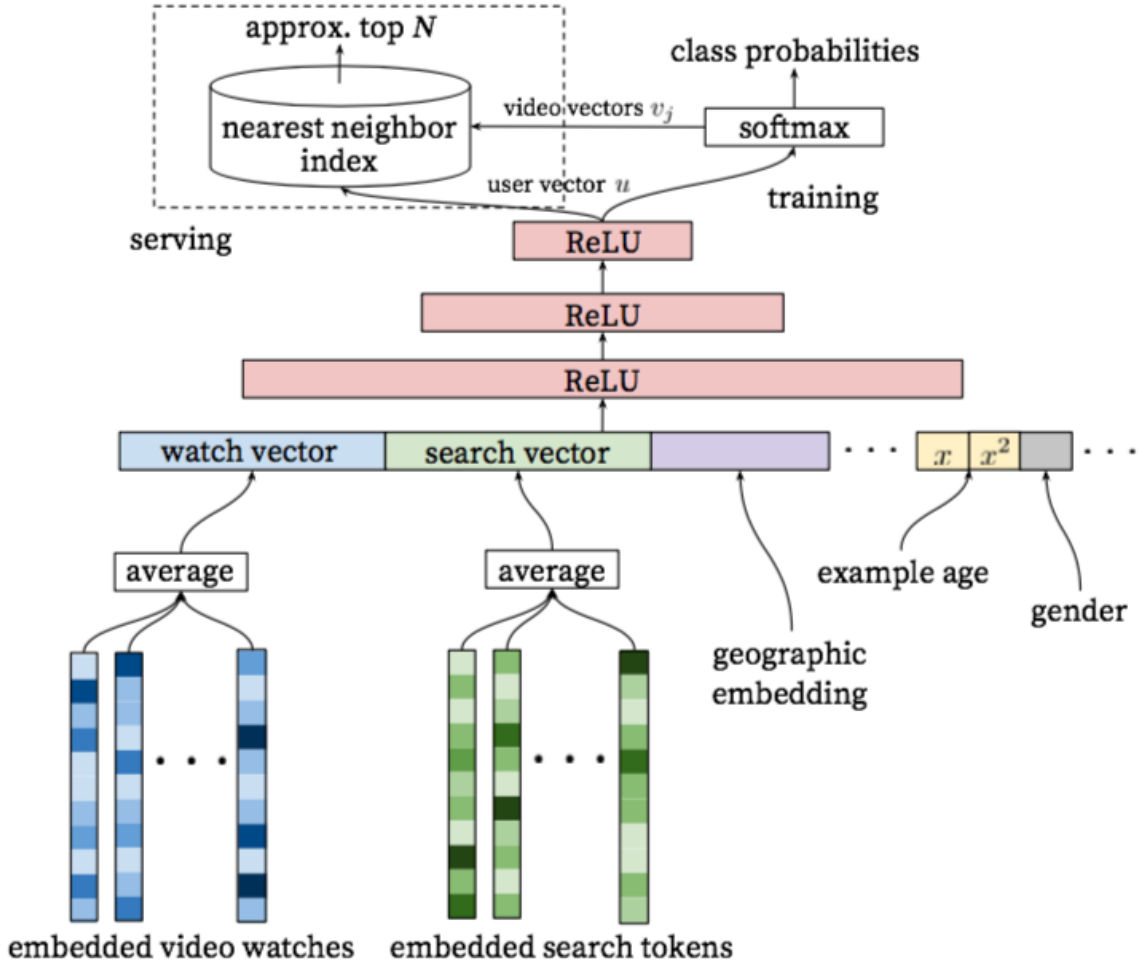
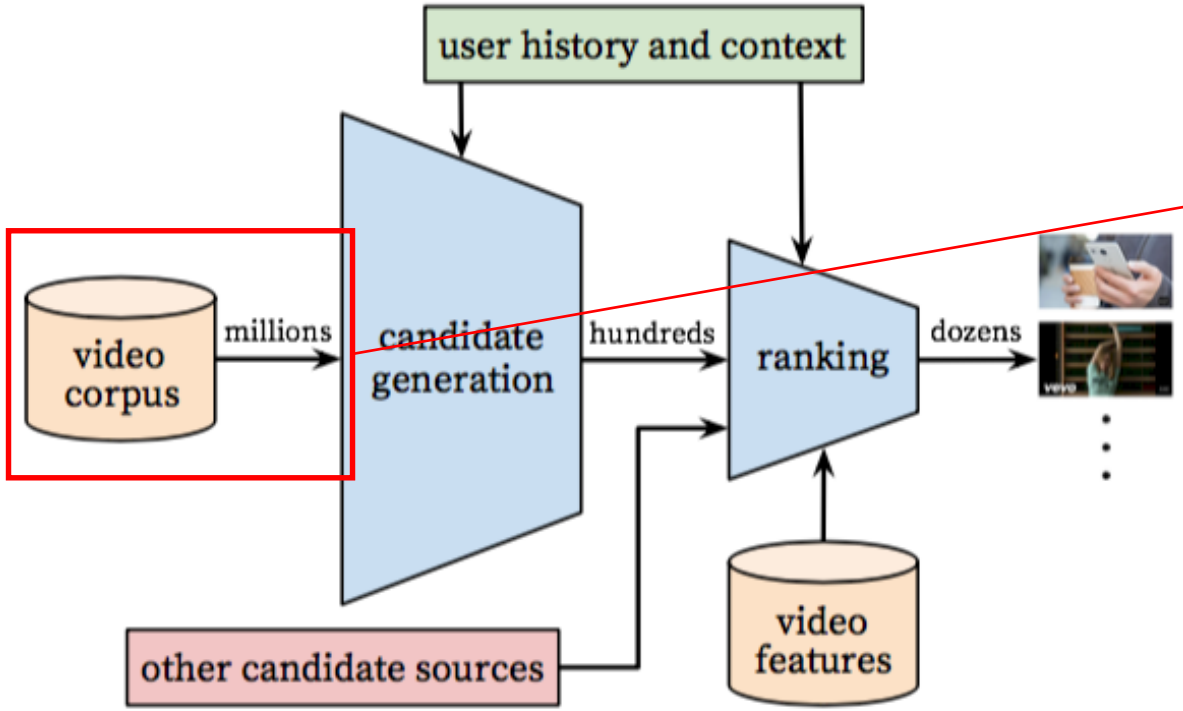


Figure 2: Recommendation system architecture demonstrating the "funnel" where candidate videos are retrieved and ranked before presenting only a few to the user.



Model



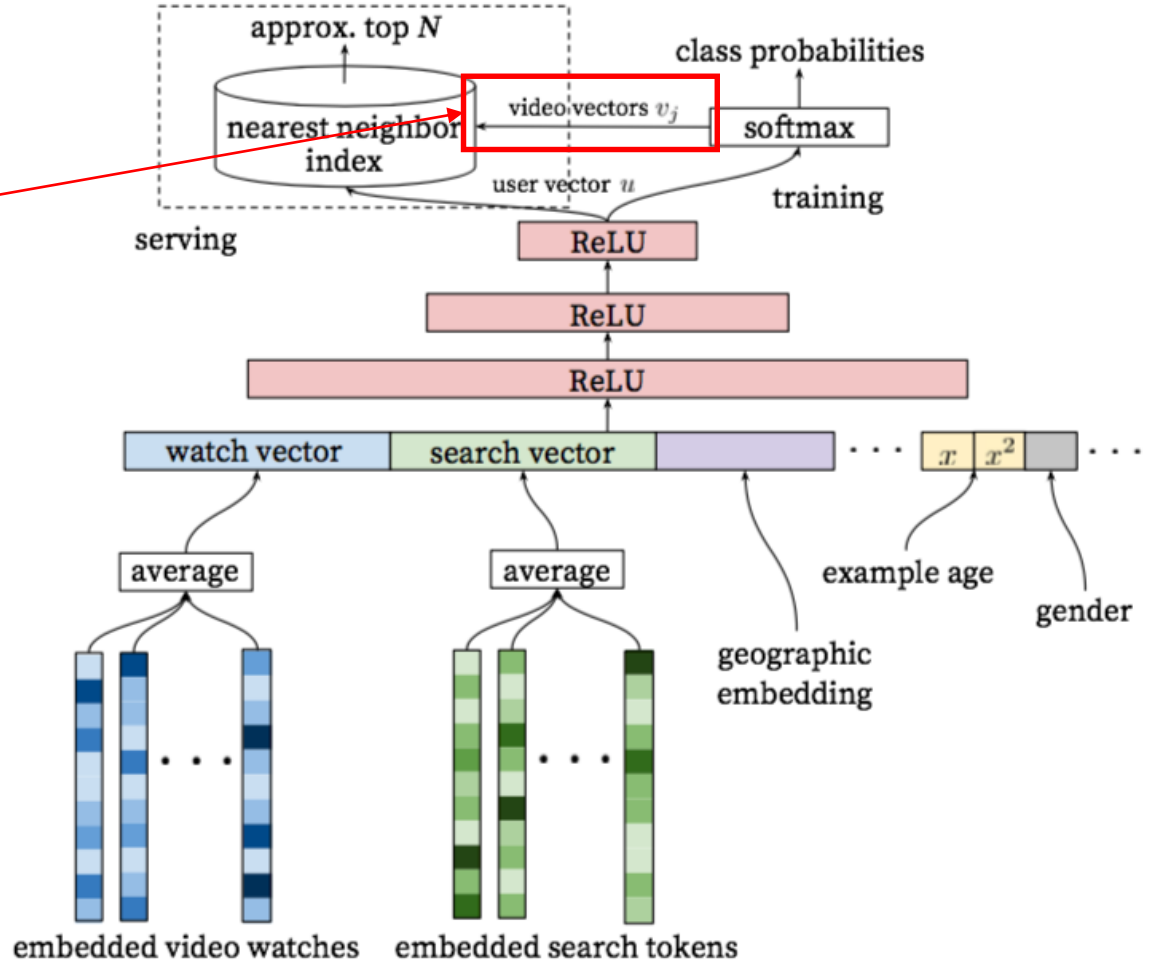


Figure 2: Recommendation system architecture demonstrating the "funnel" where candidate videos are retrieved and ranked before presenting only a few to the user.

Model

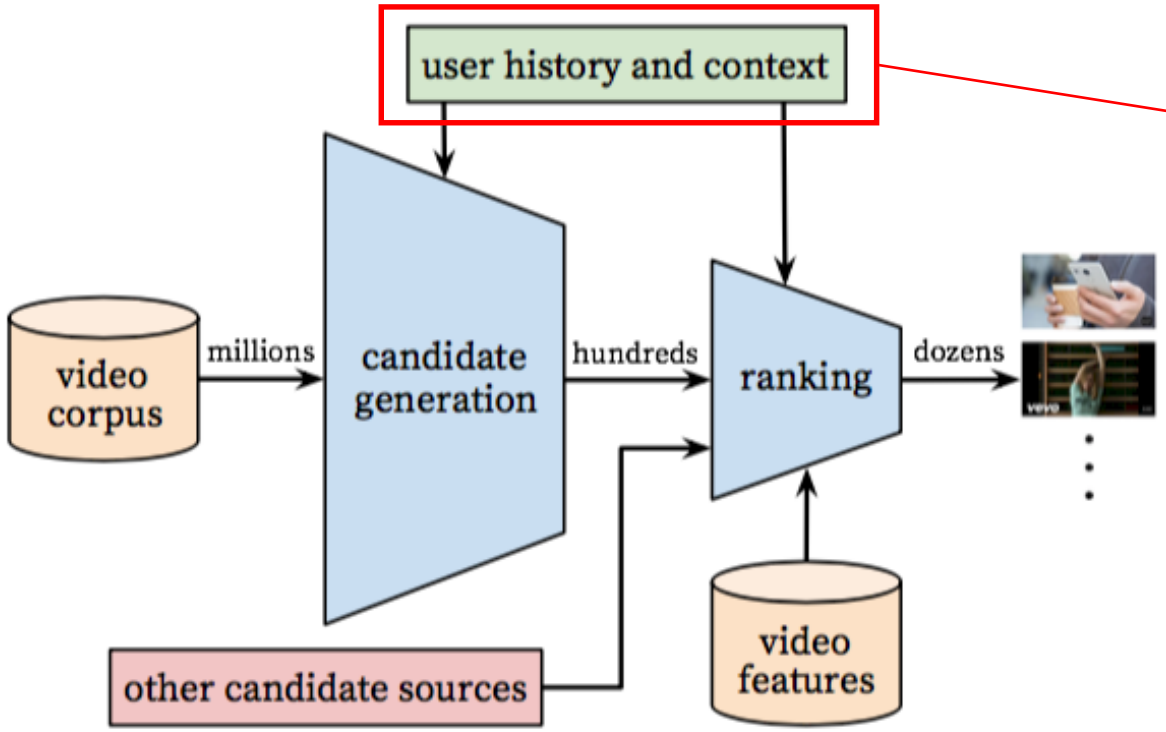
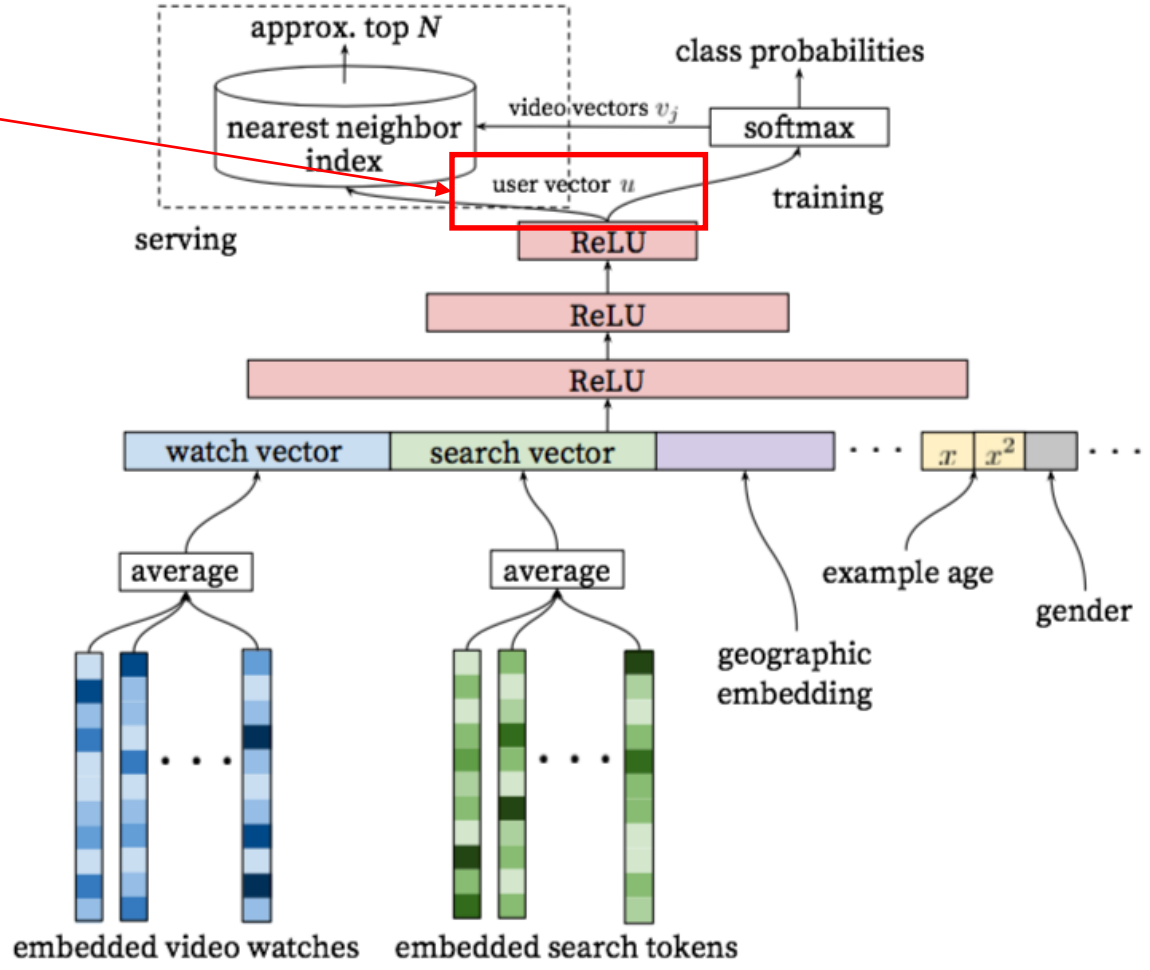


Figure 2: Recommendation system architecture demonstrating the "funnel" where candidate videos are retrieved and ranked before presenting only a few to the user.



Model

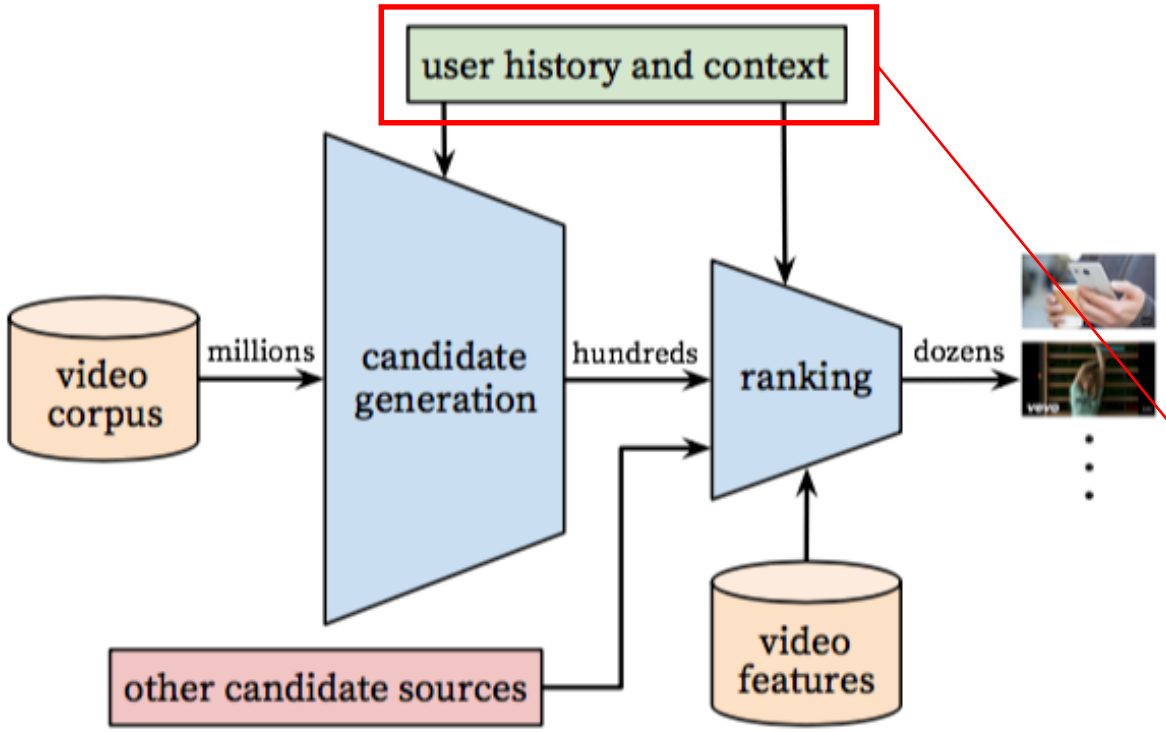
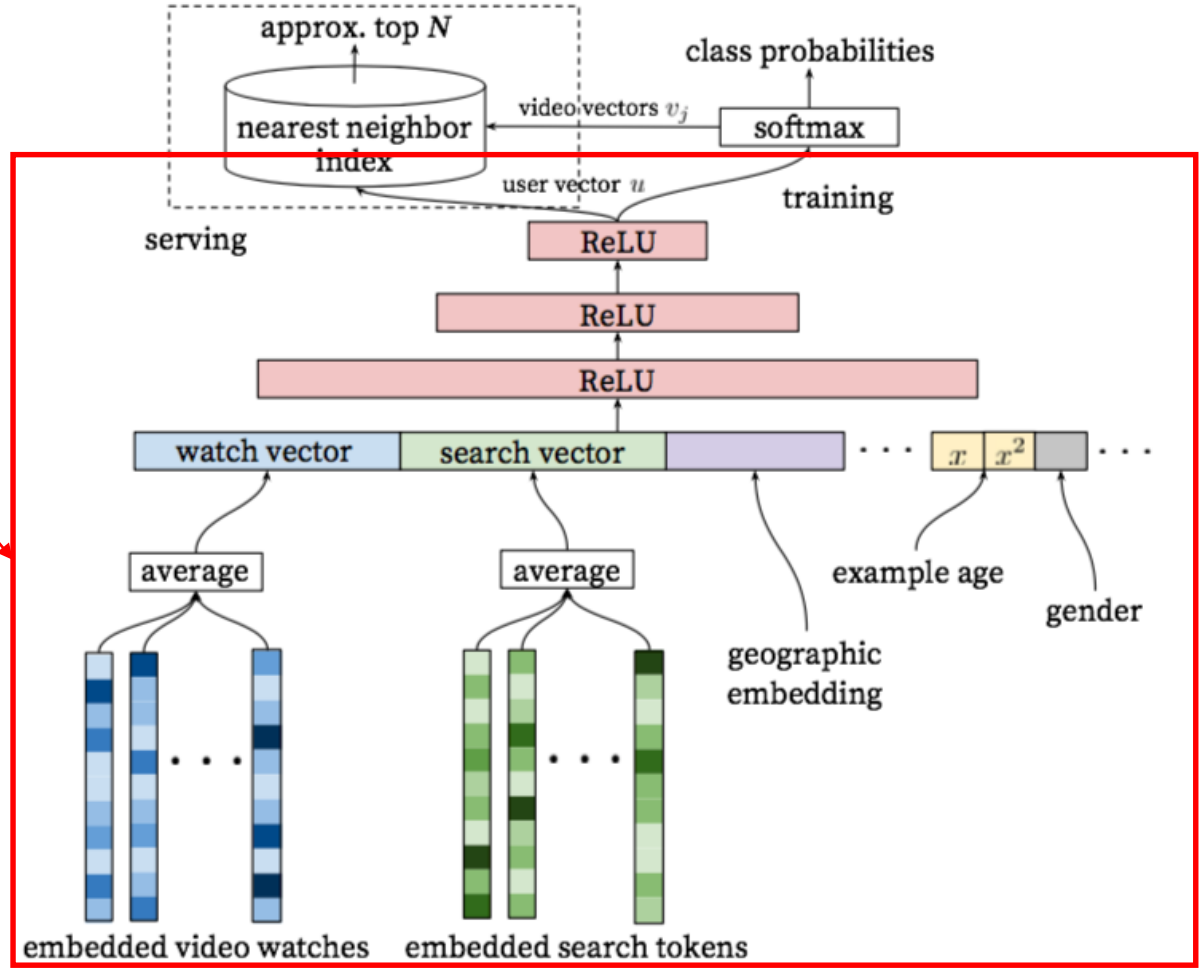


Figure 2: Recommendation system architecture demonstrating the "funnel" where candidate videos are retrieved and ranked before presenting only a few to the user.



Model

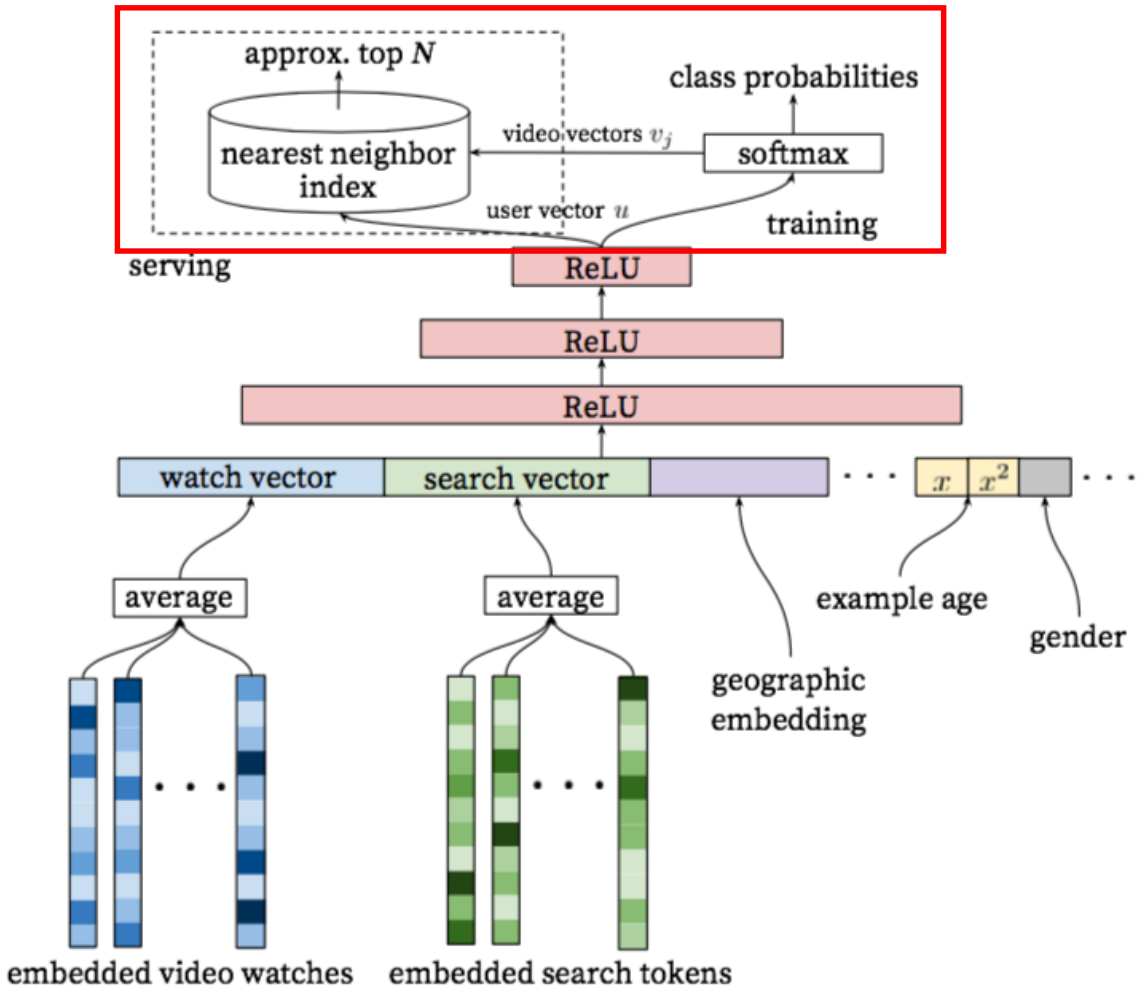
System Overview

Recommendation as Classification

추천을 extreme multiclass classification 문제로 보고 접근함

좋아요 같은 explicit feedback 이 이미 있지만, 매우 sparse하기 때문에 implicit feedback 인 시청완료 기록 등을 사용함

학습 효율을 위해 negative sampling 사용



Model

System Overview

Recommendation as Classification

추천을 extreme multiclass classification 문제로 보고 접근함

좋아요 같은 explicit feedback 이 이미 있지만. 매우 sparse하기 때문에 implicit feedback 인 시청완료 기록 등을 사용함

학습 효율을 위해 negative sampling 사용

sifying a specific video watch w_t at time t among millions of videos i (classes) from a corpus V based on a user U and context C,

특정 시간 t에 특정 비디오를 볼 확률

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

where $u \in \mathbb{R}^N$ represents a high-dimensional "embedding" of the user, context pair and the $v_j \in \mathbb{R}^N$ represent embeddings of each candidate video. In this setting, an embedding is simply a mapping of sparse entities (individual videos, users etc.) into a dense vector in $\mathbb{R}^N$. The task of the deep neural network is to learn user embeddings u as a function of the user's history and context that are useful for discriminating

More than x100 speedup

+) Serving time에는 softmax 대신 nearest neighbor (in dot product space)

Model

System Overview

[1] Candidate generation

User's watch history:
Variable-length sequence of sparse video IDs
-> averaging the embeddings로 변환됨
(각각의 비디오 임베딩의 평균 값)

-> 모델의 다른 파라미터와 함께 학습됨

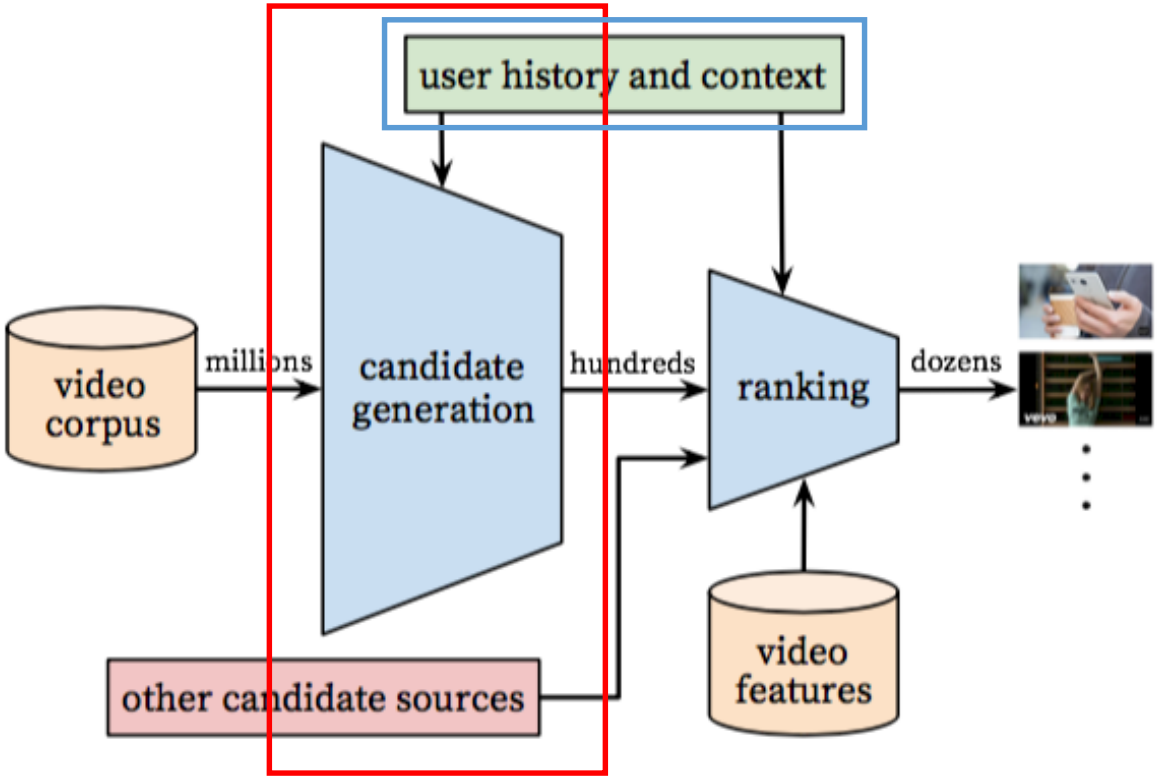


Figure 2: Recommendation system architecture demonstrating the "funnel" where candidate videos are retrieved and ranked before presenting only a few to the user.

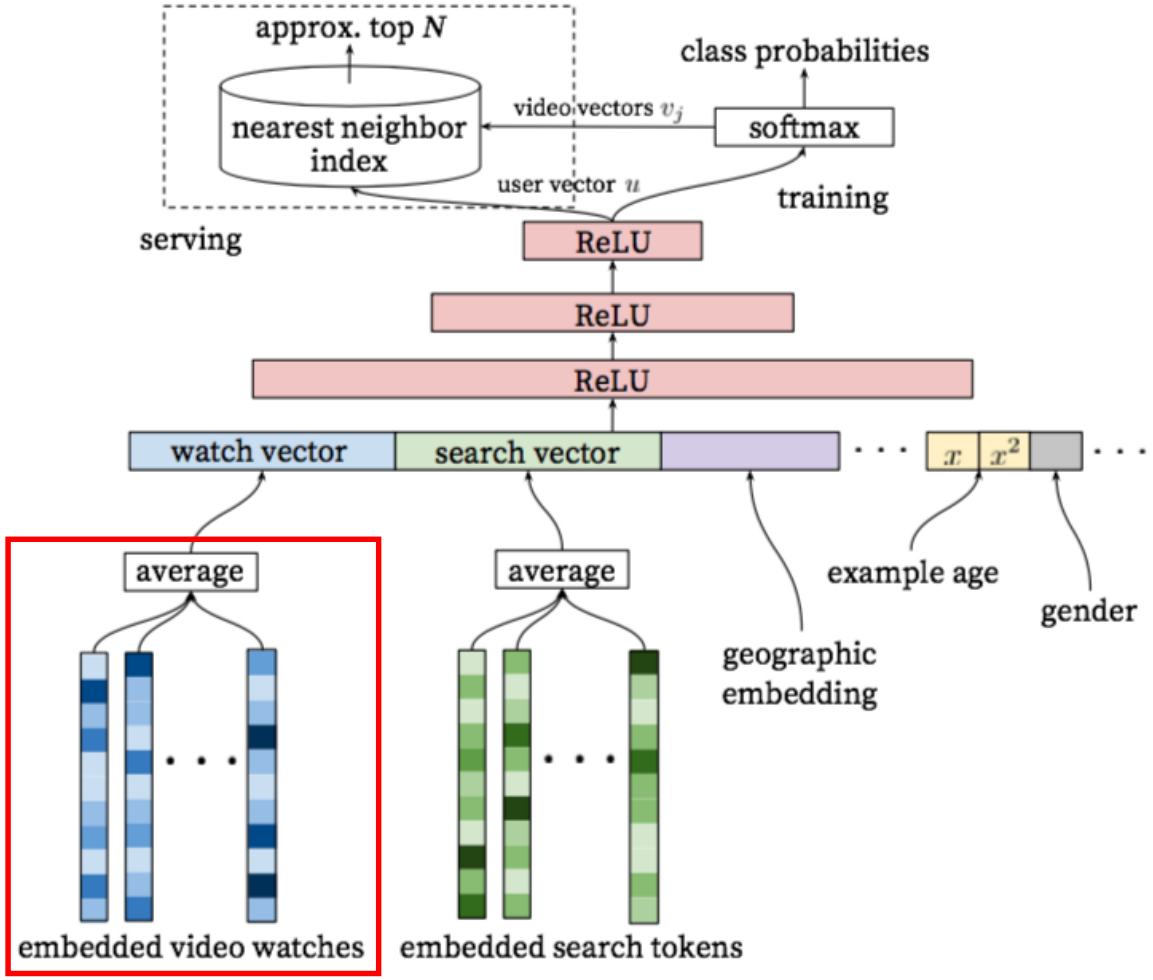
Model

System Overview

[1] Candidate generation

User's watch history:
Variable-length sequence of sparse video IDs
-> averaging the embeddings로 변환됨
(각각의 비디오 임베딩의 평균 값)

-> 모델의 다른 파라미터와 함께 학습됨

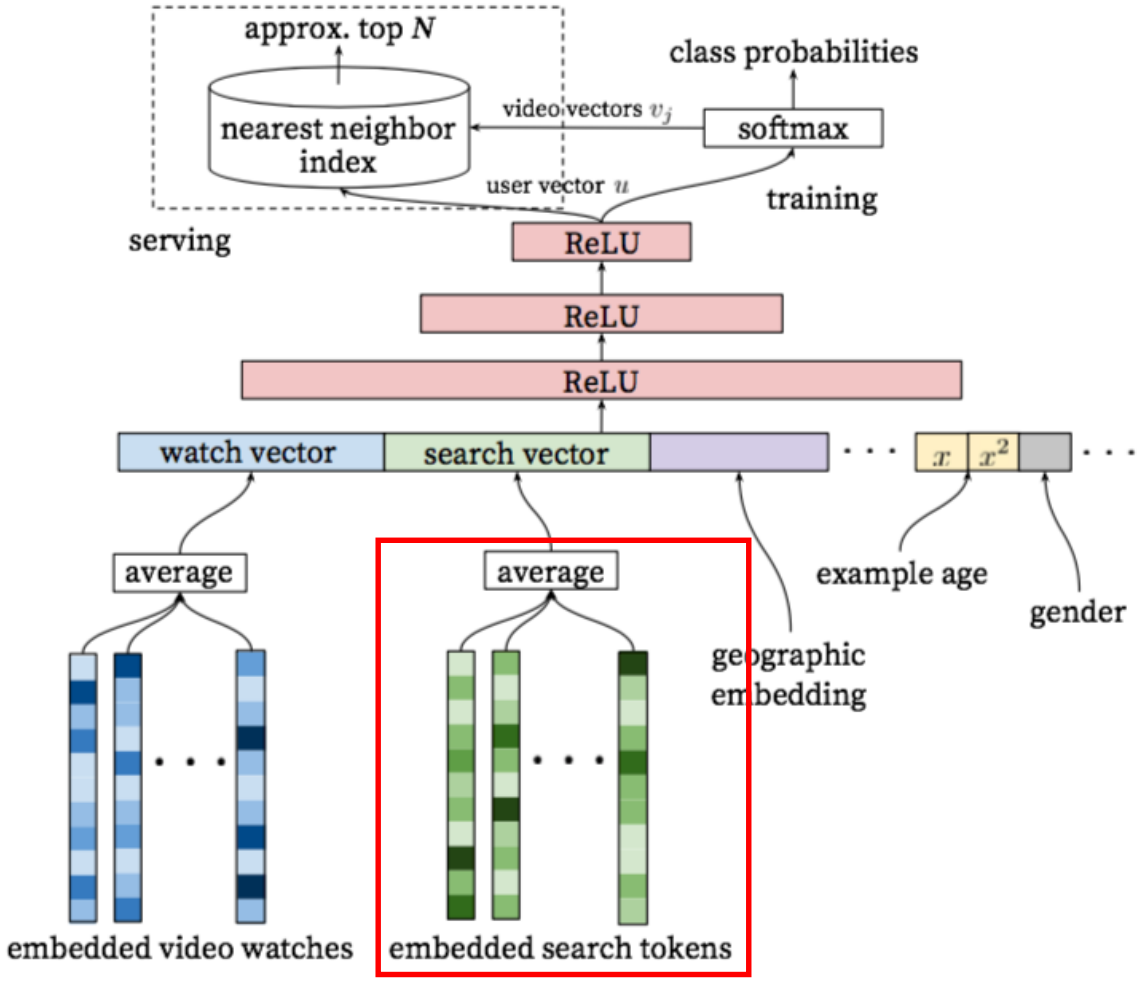


Model

System Overview

[1] Candidate generation

User's search history:
Query 를 unigram, bigram으로 토큰화 한 후 각 토큰을 임베딩하고 평균 값을 사용



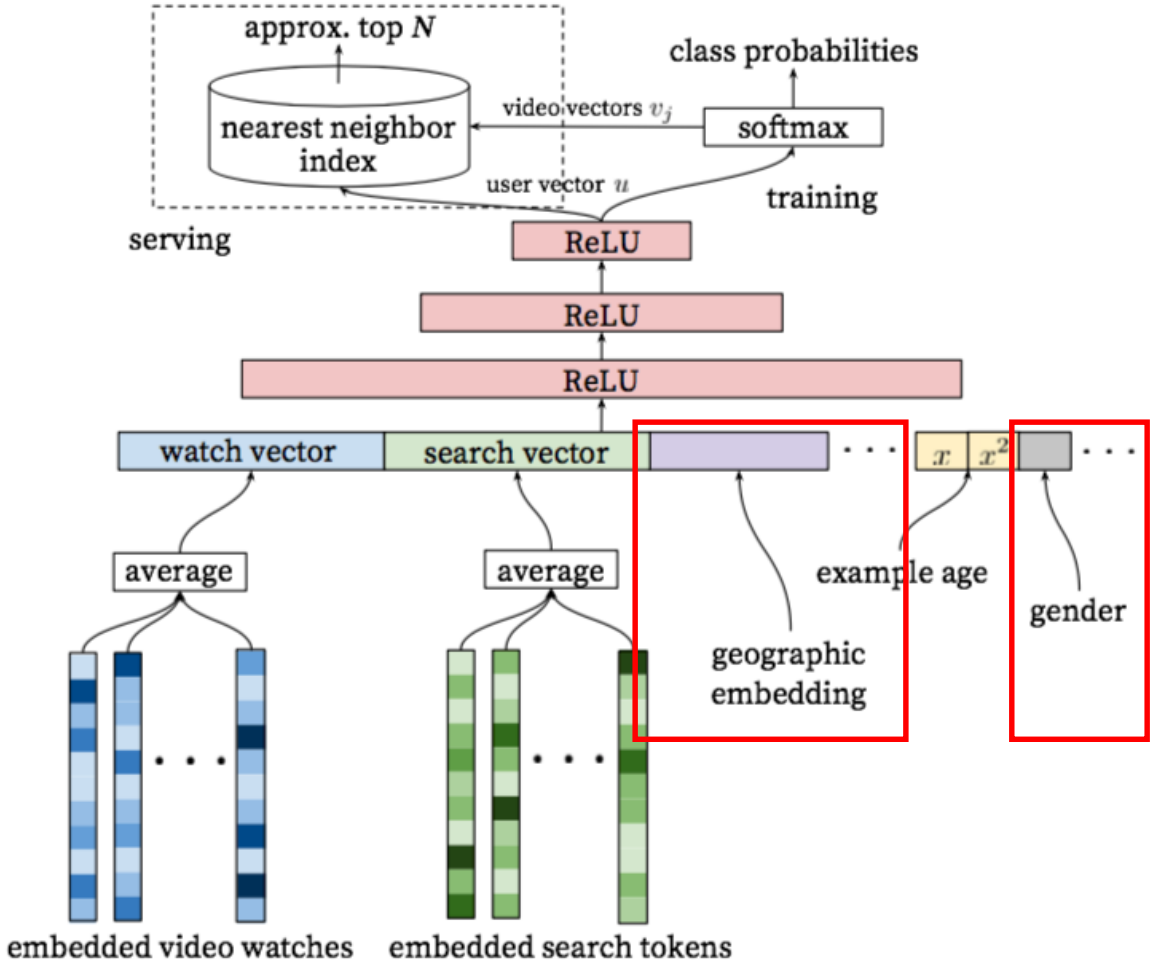
Model

System Overview

[1] Candidate generation

Demographic feature:
유저의 지역적 특징이나 디바이스등의 정보는 새로운 유저에게 reasonably 대응하기 위한 Prior 정보로 사용

(user's gender, logged-in sate, age등의 정보도 [0, 1] 사이로 정규화해서 사용됨)



Model

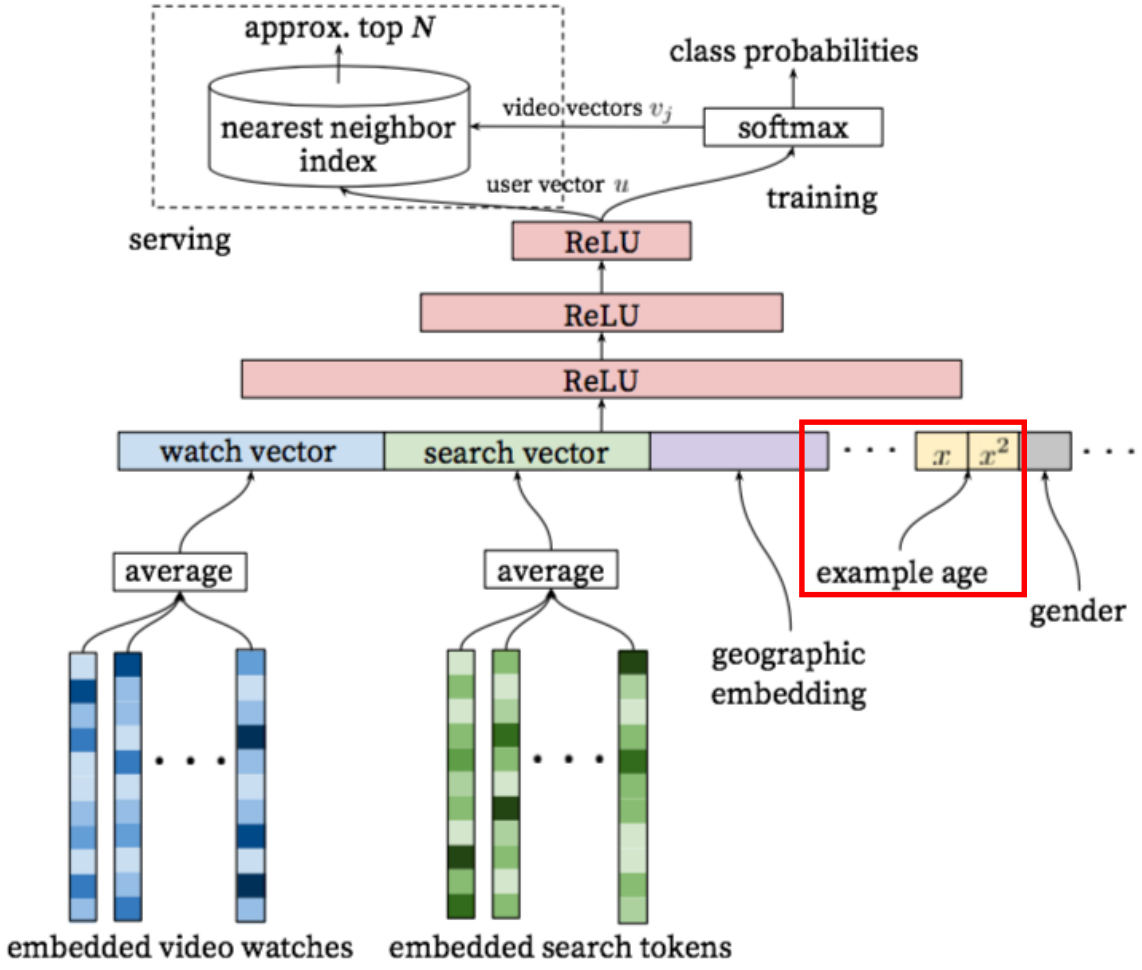
System Overview

[1] Candidate generation

Example Age feature:
추천은 특성상, "fresh" content 를 다루는 게 매우 중요하기 때문에 content 의 age를 고려해야함

안 그러면, 과거 데이터로 학습하다보니 주로 과거 영상이 많이 추천될 수 있음

실험결과 example age feature는 time-dependant popularity를 표현 할 수 있음



Model

System Overview

[1] Candidate generation

Example Age feature:
추천은 특성상, "fresh" content 를 다루는 게 매우 중요하기 때문에 content 의 age를 고려해야함

안 그러면, 과거 데이터로 학습하다보니 주로 과거 영상이 많이 추천될 수 있음

실험결과 example age feature는 time-dependant popularity를 표현 할 수 있음

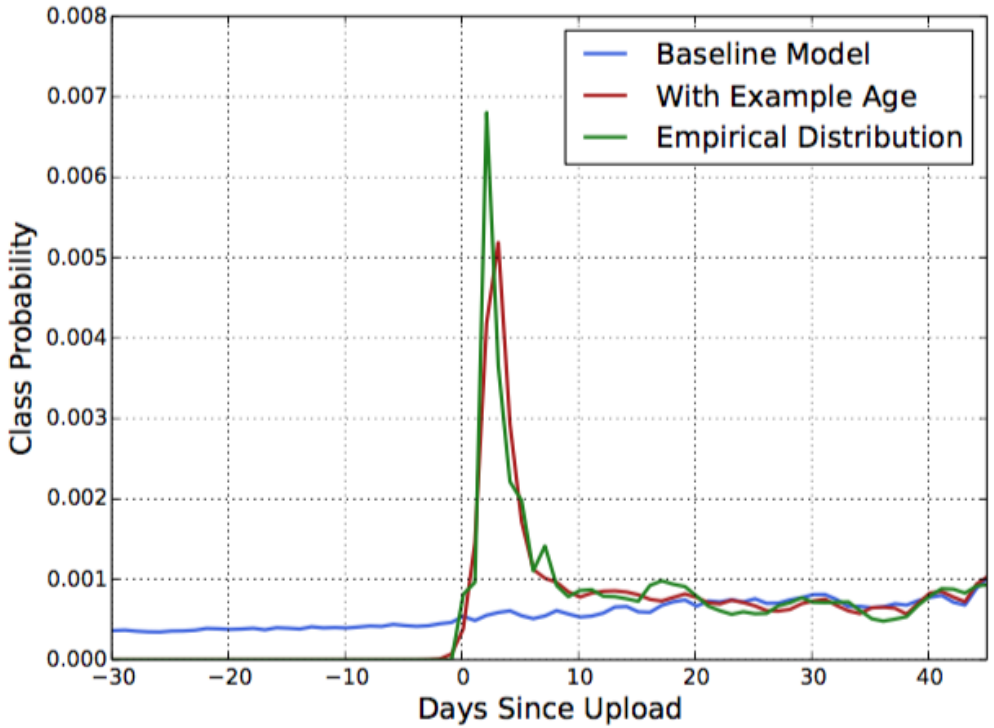


Figure 4: For a given video [26], the model trained with example age as a feature is able to accurately represent the upload time and time-dependant popularity observed in the data. Without the feature, the model would predict approximately the average likelihood over the training window.

Model

System Overview

[1] Candidate generation

Label and Context Selection:

추천 문제는 surrogate problem 을 통해 해결 할 수 있다

Ex) 영화 평점 알고리즘은 영화 추천으로 사용 가능

-> YouTube에서도 예측으로 추천을 하자!

예측을 위한 학습 데이터 구성)
- 학습 데이터는 밖에서 사용된 모든 비디오 -> 이미 추천된 결과 위주의 bias 방지

- Fixed number of training examples per user 사용 -> 특정 active users의 dominating the loss를 막아줌

- Withhold information from the classifier -> surrogate problem에 overfit 되는 걸 막아줌

Model

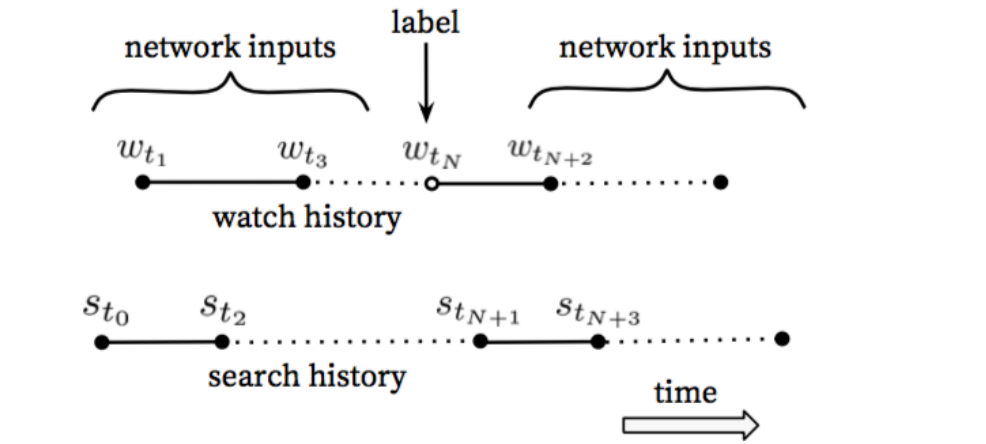
System Overview

[1] Candidate generation

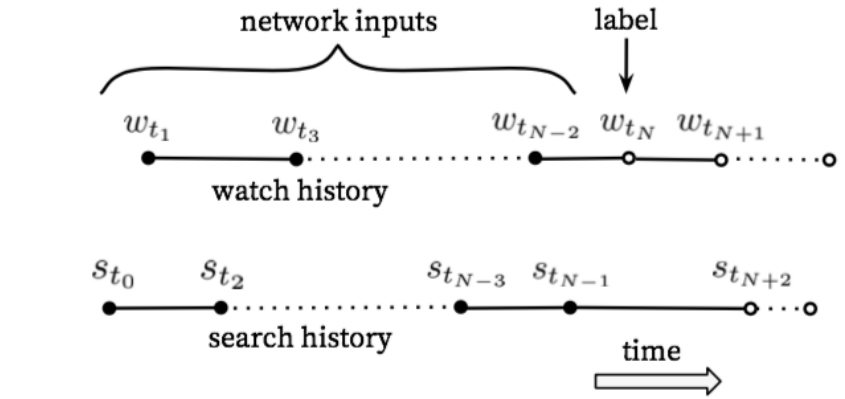
Label and Context Selection:

(b) 처럼 "Sequence" 형태를 input으로 사용해서 future watch를 예측하는 것이 A/B Testing에서 더 좋은 결과를 기록함

-> User's history를 random watch 선택 후 rollback (이전 기록 까지만 input)
-> 감상 패턴은 '비대칭 ' 이라고 할 수 있음



(a) Predicting held-out watch



(b) Predicting future watch

Model

System Overview

[1] Candidate generation

Experiments with Features & Depth :

구성 : 1M videos & 1M search tokens
Maximum bag size: 최근 50 watches & searches
임베딩 벡터크기: 256 floats로 임베딩됨

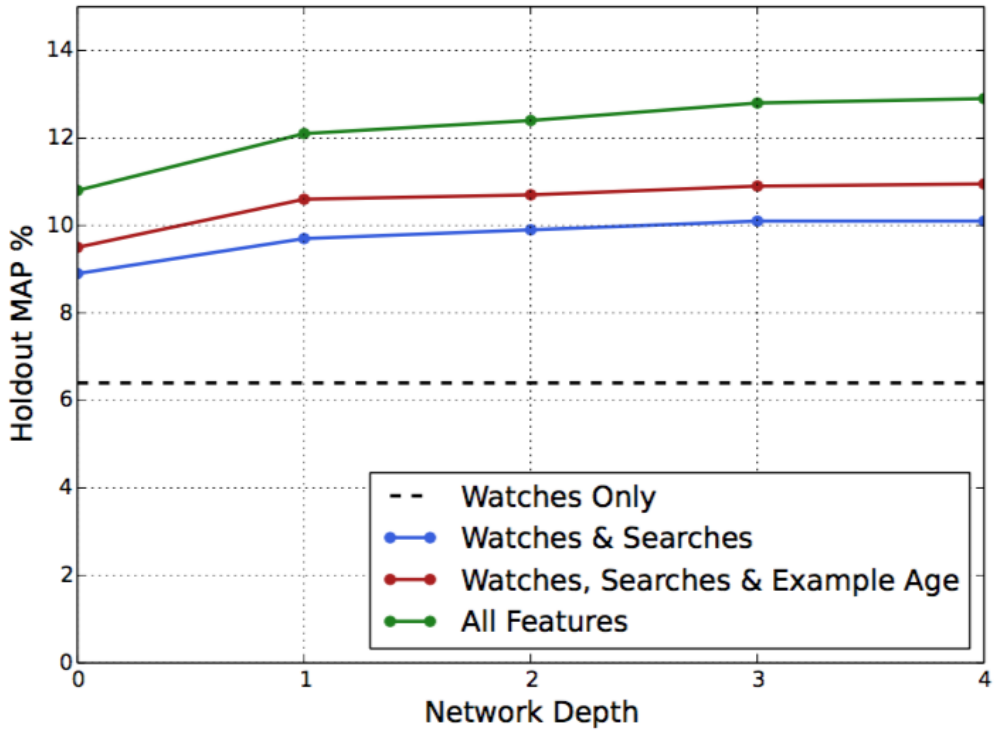


Figure 6: Features beyond video embeddings improve holdout Mean Average Precision (MAP) and layers of depth add expressiveness so that the model can effectively use these additional features by modeling their interaction.

Model

System Overview

[1] Candidate generation

Experiments with Features & Depth :

구성 : 1M videos & 1M search tokens
Maximum bag size: 최근 50 watches & searches
임베딩 벡터크기: 256 floats로 임베딩됨

- Depth 0: A linear layer simply transforms the concatenation layer to match the softmax dimension of 256
- Depth 1: 256 ReLU
- Depth 2: 512 ReLU → 256 ReLU
- Depth 3: 1024 ReLU → 512 ReLU → 256 ReLU
- Depth 4: 2048 ReLU → 1024 ReLU → 512 ReLU → 256 ReLU

Model

System Overview

[1] Candidate generation

Experiments with Features & Depth :

Search & Example Age feature가 성능을 높이는데 중요한 것을 확인할 수 있음

전체적으로 Layer 개수가 많을 때 성능이 향상됨

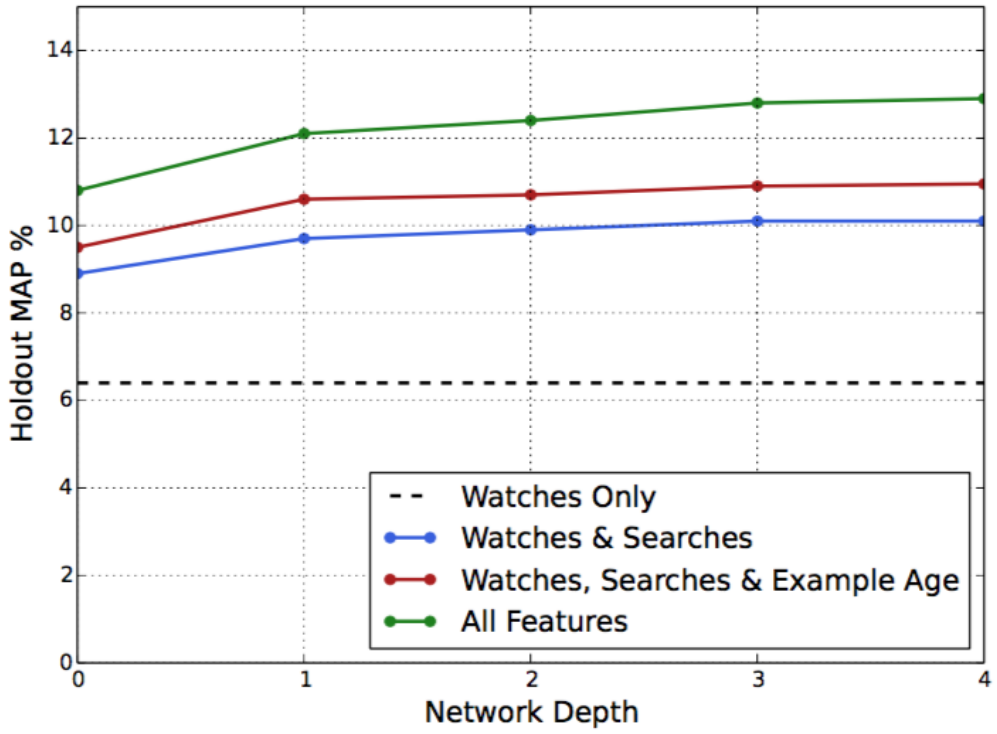


Figure 6: Features beyond video embeddings improve holdout Mean Average Precision (MAP) and layers of depth add expressiveness so that the model can effectively use these additional features by modeling their interaction.

Model

System Overview

[2] Ranking

각각의 비디오에 점수를 부여해서 순위 결정

Candidate generation과 비슷한 NN 사용

Ranking objective:
= Expected watch time / impression
(tuned based on live A/B testing)

(watch time이 click-through rate보다 좋음)

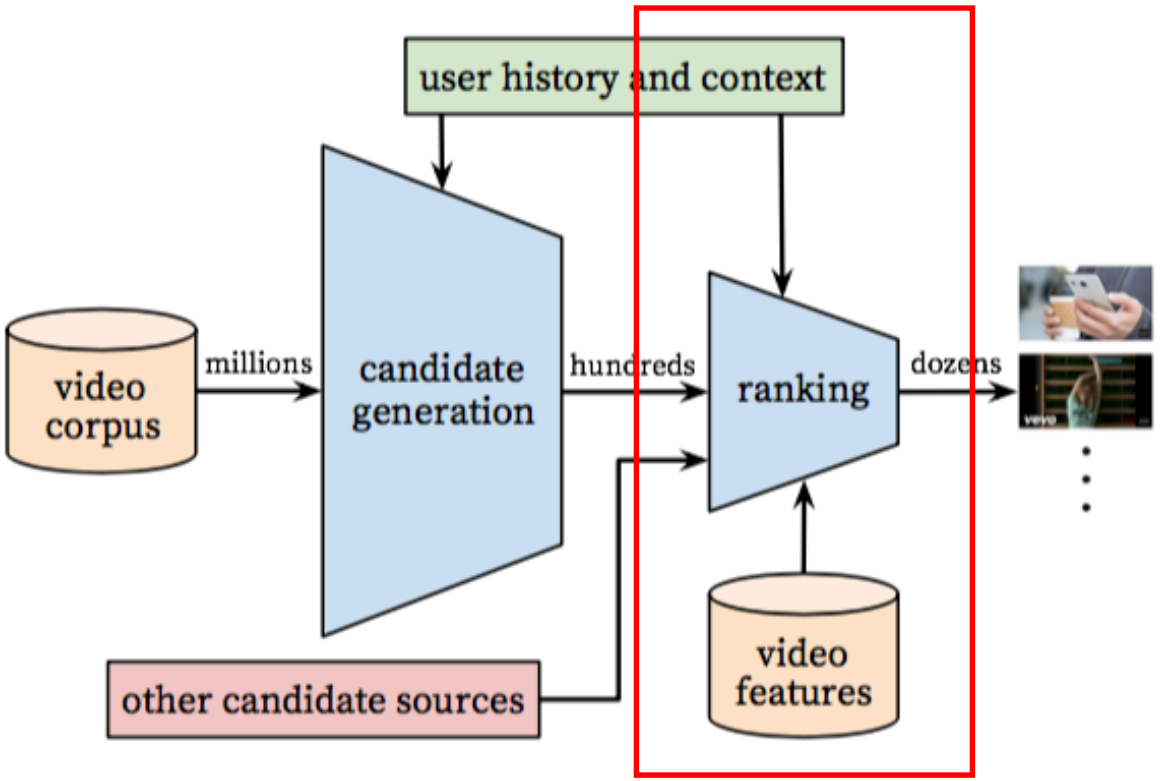


Figure 2: Recommendation system architecture demonstrating the "funnel" where candidate videos are retrieved and ranked before presenting only a few to the user.

Model

System Overview

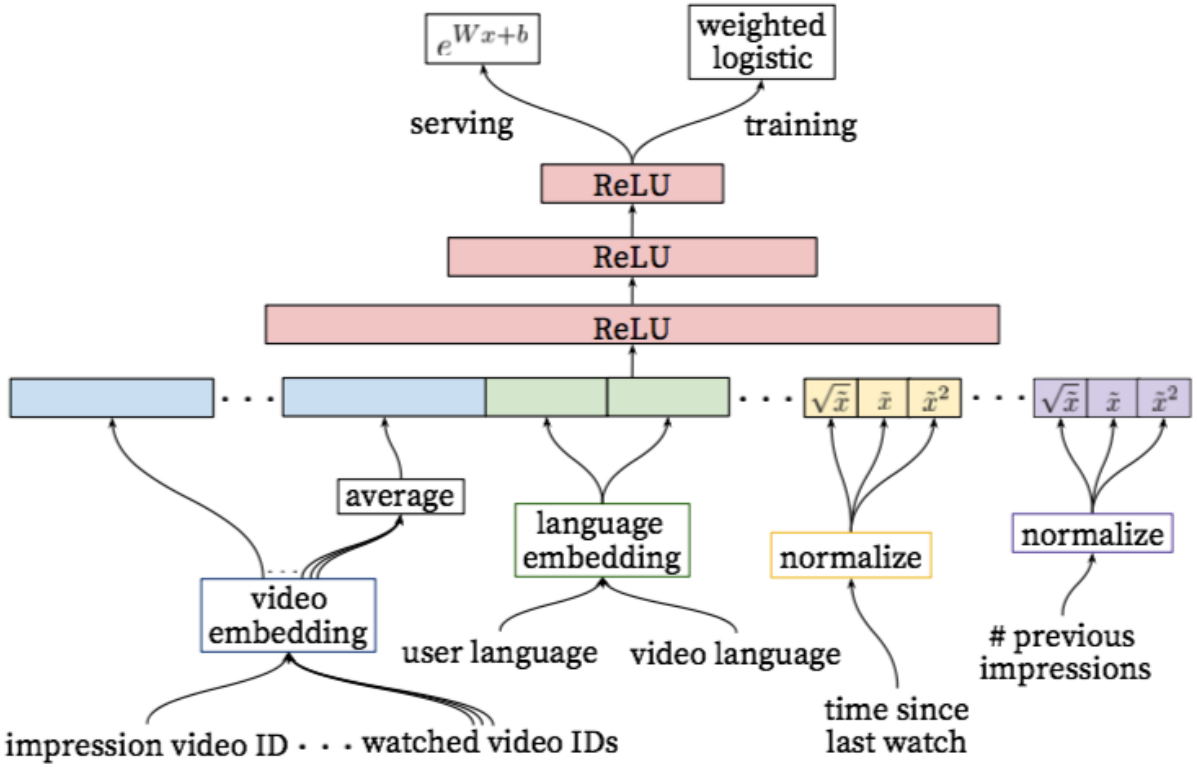
[2] Ranking

각각의 비디오에 점수를 부여해서 순위 결정

Candidate generation과 비슷한 NN 사용

Ranking objective:
= Expected watch time per impression
(tuned based on live A/B testing)

(watch time이 click-through rate보다 좋음)



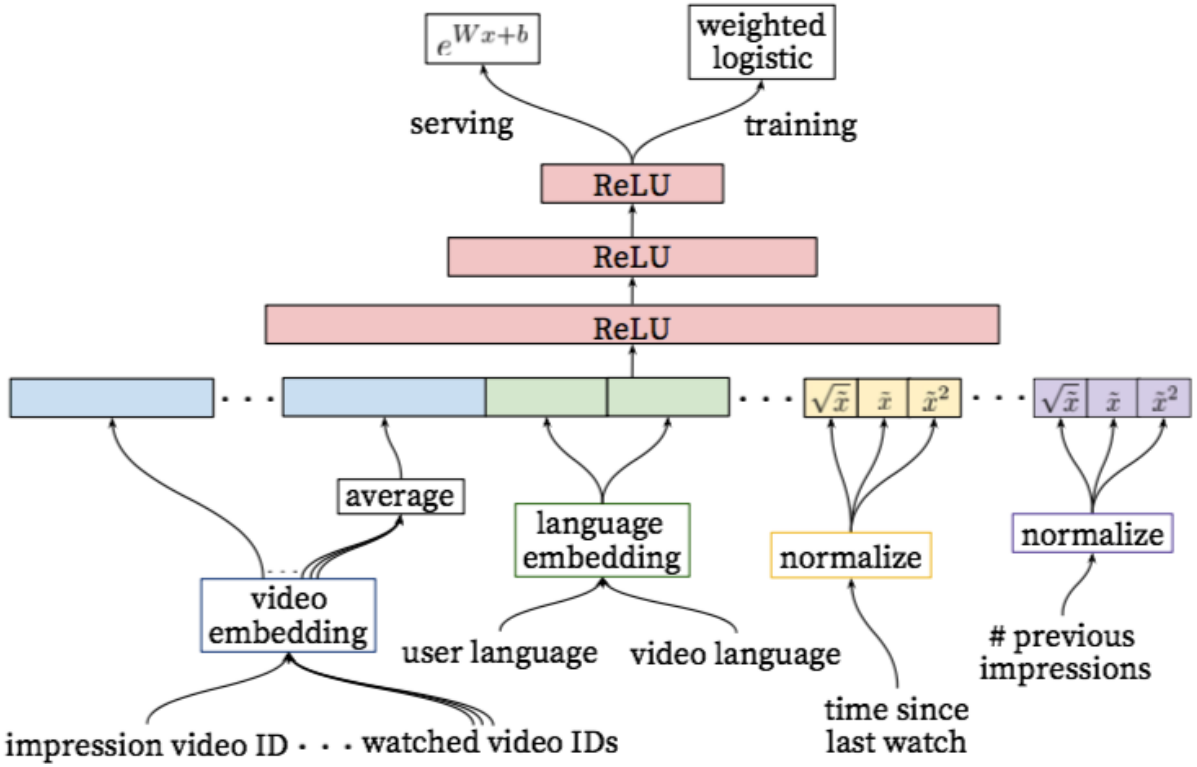
Model

System Overview

[2] Ranking

Feature Representation:
Hundreds of features are fed into NN

- Logged-in (binary)
- Last search query
- Video ID of the impression (univalent)
- Bag of the last N video IDs (multivalent)
- etc

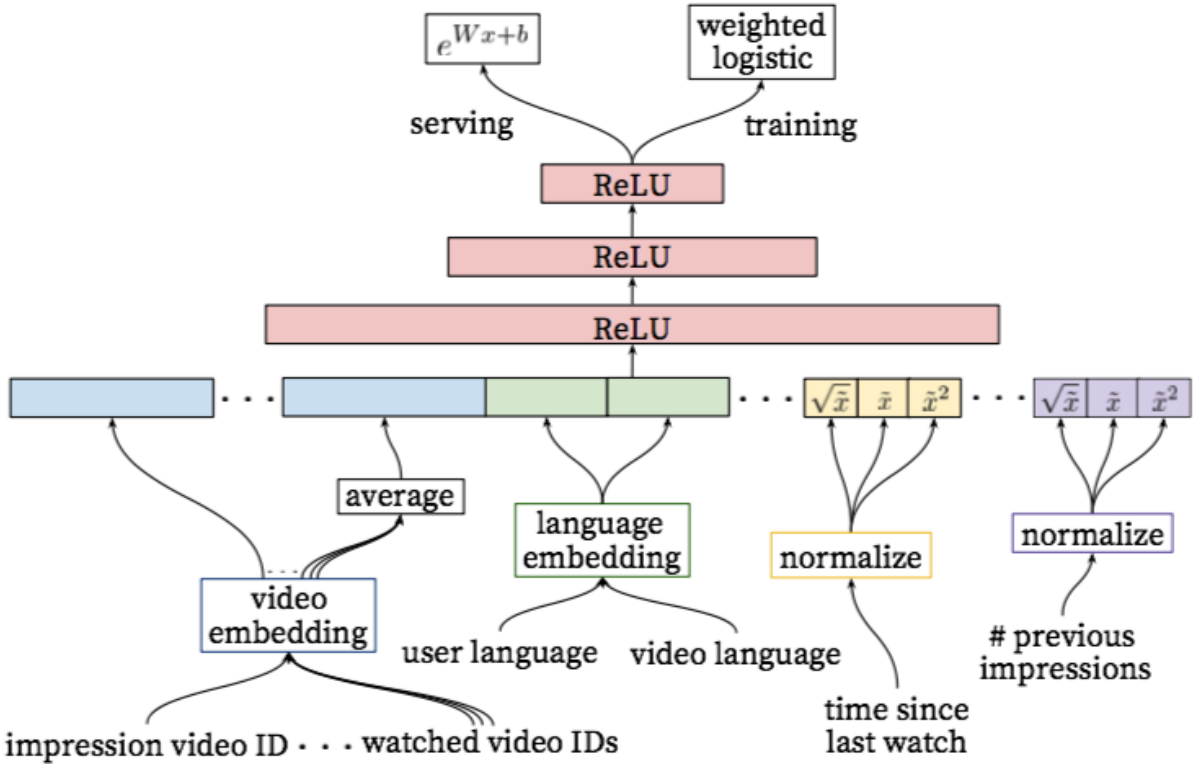


Model

System Overview

[2] Ranking

Feature Representation:
- Multivalent categorical features는 이전과 같이 average 된 값을 사용함
- 학습을 위해 normalization 적용, (super-, sub- linear) 형태 적용
- Same ID space에 있는 embedding은 같은 걸 사용함 (ex. Global embedding of video IDs) -> generalization, 학습속도 향상에 도움



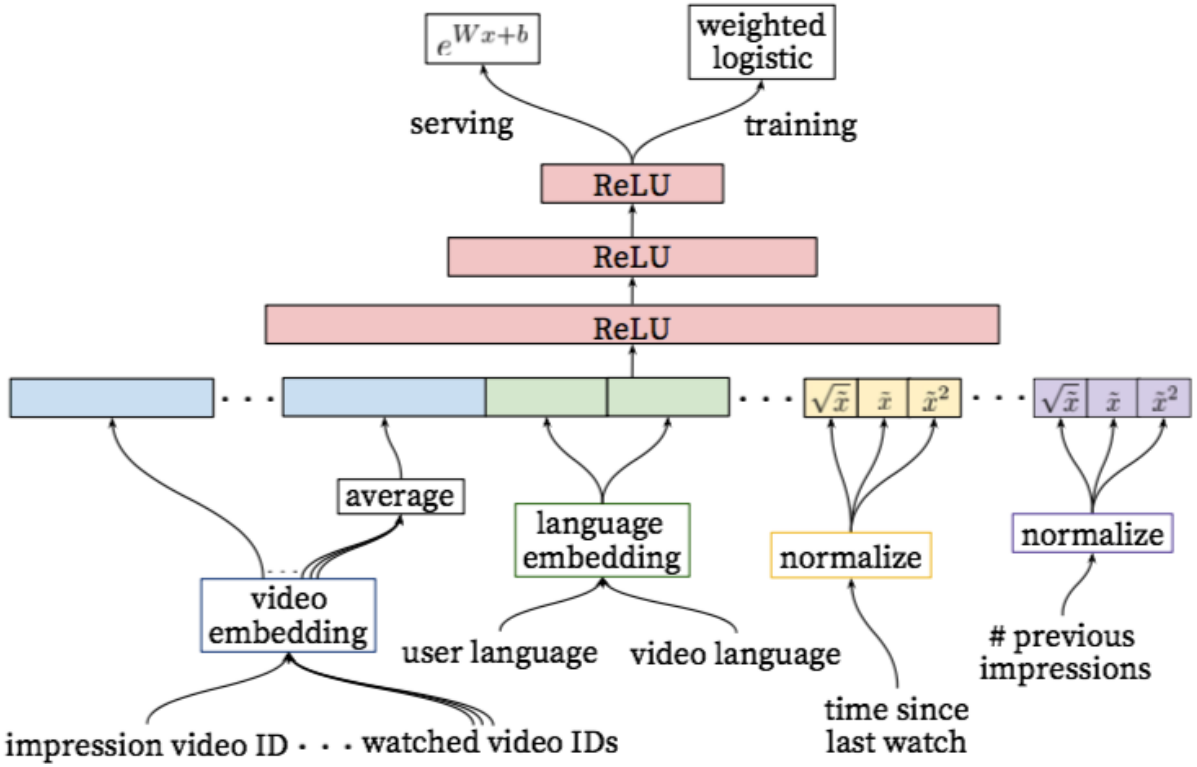
Model

System Overview

[2] Ranking

Modeling Expected Watch Time:
- Goal : predict expected watch time
- Positive set: impression (clicked)
- Negative set: impression (not clicked)

- Weighted 규칙
  positive impression의 경우 observed watch에 의해 가중치 부여
  negative impression은 균일하게 부여됨



Model

System Overview

[2] Ranking

Experiments with Hidden Layers:

- Negative impression 의 점수가 positive impression 보다 높다면 watch time을 잘못 예측한 것이라고 판단함
- Weighted, per-user loss는 이러한 상황에서 mispredicted watch time의 비율을 나타낸 것임
- Weighted equally 적용한 모델의 경우, loss가 4.1% 증가함

Hidden layers	weighted, per-user loss
None	41.6%
256 ReLU	36.9%
512 ReLU	36.7%
1024 ReLU	35.8%
512 ReLU → 256 ReLU	35.2%
1024 ReLU → 512 ReLU	34.7%
1024 ReLU → 512 ReLU → 256 ReLU	34.6%

Table 1: Effects of wider and deeper hidden ReLU layers on watch time-weighted pairwise loss computed on next-day holdout data.

Model

System Overview

Evaluation

During development,
Offline metrics
(precision, recall, ranking loss, etc)

During live experiments,
A/B Test
(changes in click-through rate, watch time, etc)

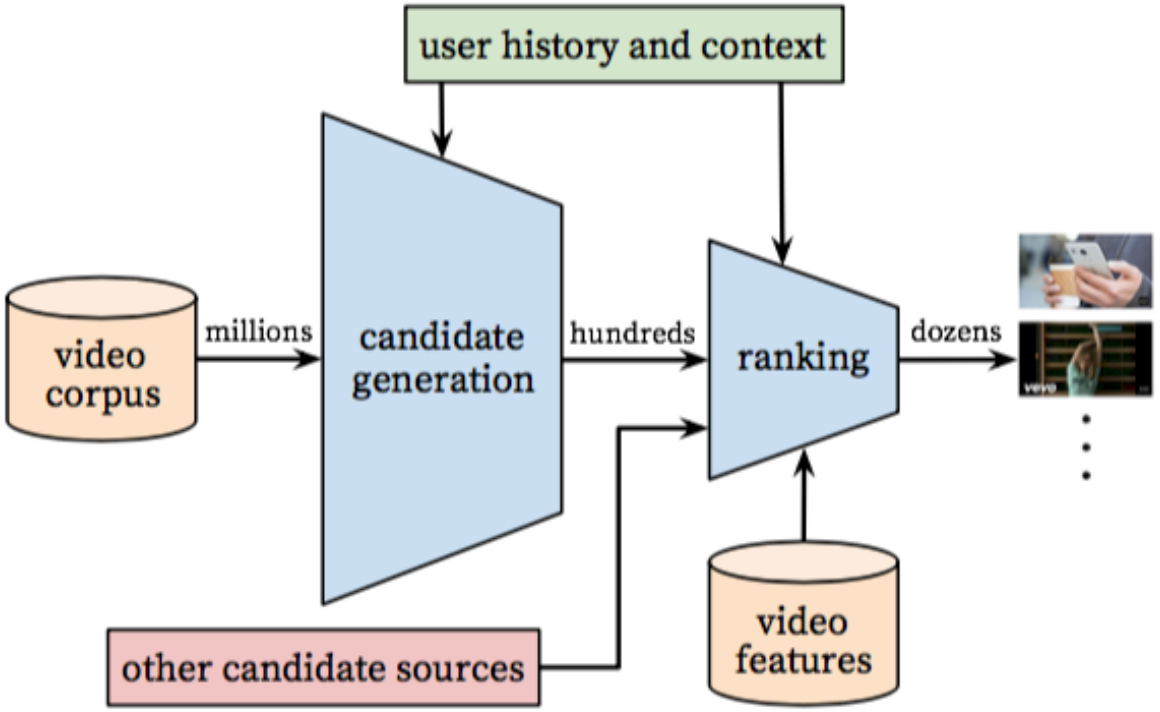


Figure 2: Recommendation system architecture demonstrating the "funnel" where candidate videos are retrieved and ranked before presenting only a few to the user.

Conclusion

추천 문제를 해결하기 위해 [1] Candidate generation, [2] Ranking 두 문제로 나눠서 접근

기존엔 주로 Matrix factorization방식이었으나 본 논문에서는 Deep Learning을 적용

기존의 방법보다 높은 성능을 보여주었으며 이를 가능케한 feature 및 모델 구조를 분석함

Example age는 time-dependent behavior 를 나타내는데 효과적인 자질로 사용됨

weighted logistic regression 을 통해 감상 시간별로 가중치를 주는 것이 click-through rate를 사용할 때보다 더 좋은 성능을 보임

QnA

Thank you :)