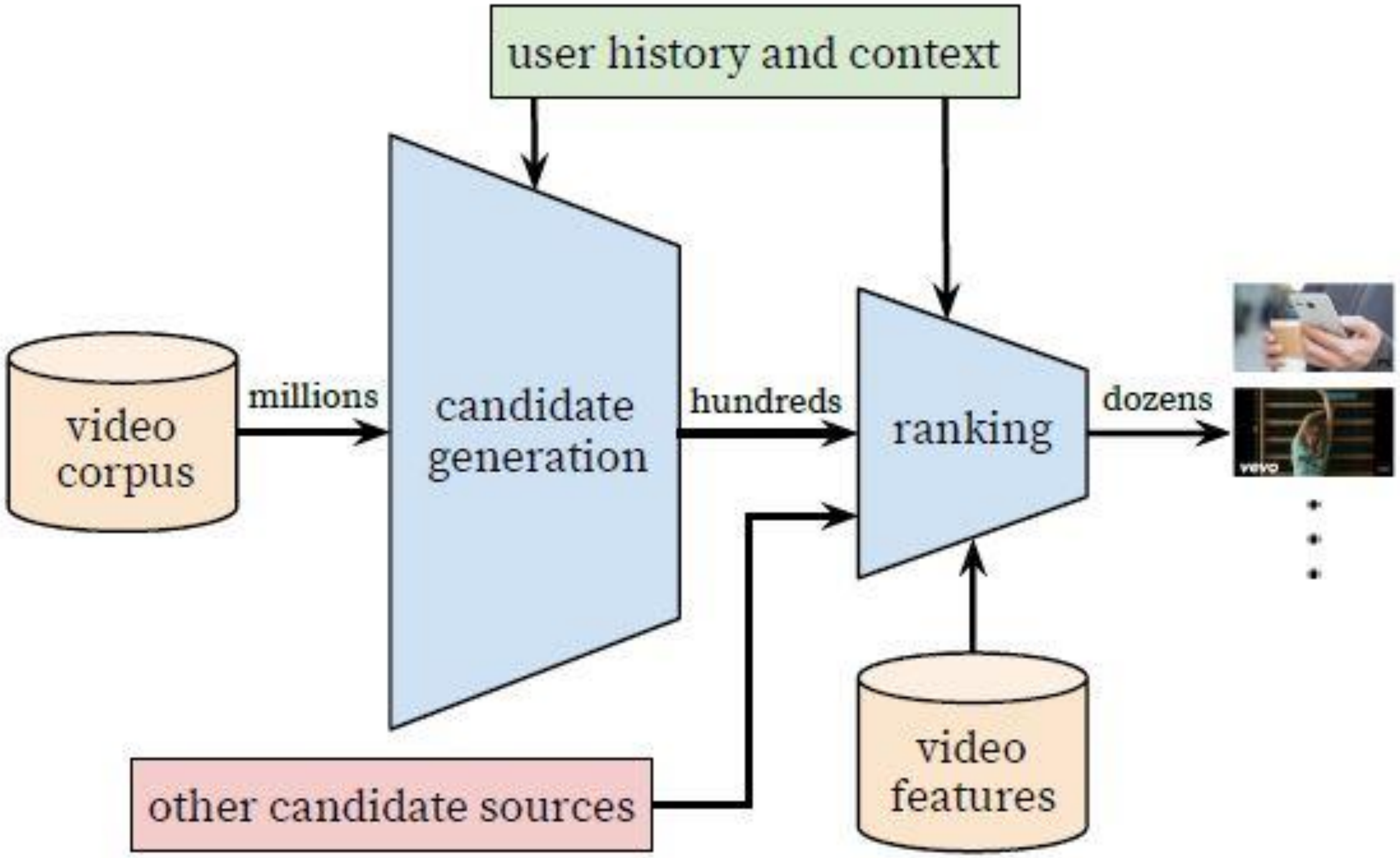
Deep Neural Networks for Youtube Recommendations

Summary

- Fully connected layer

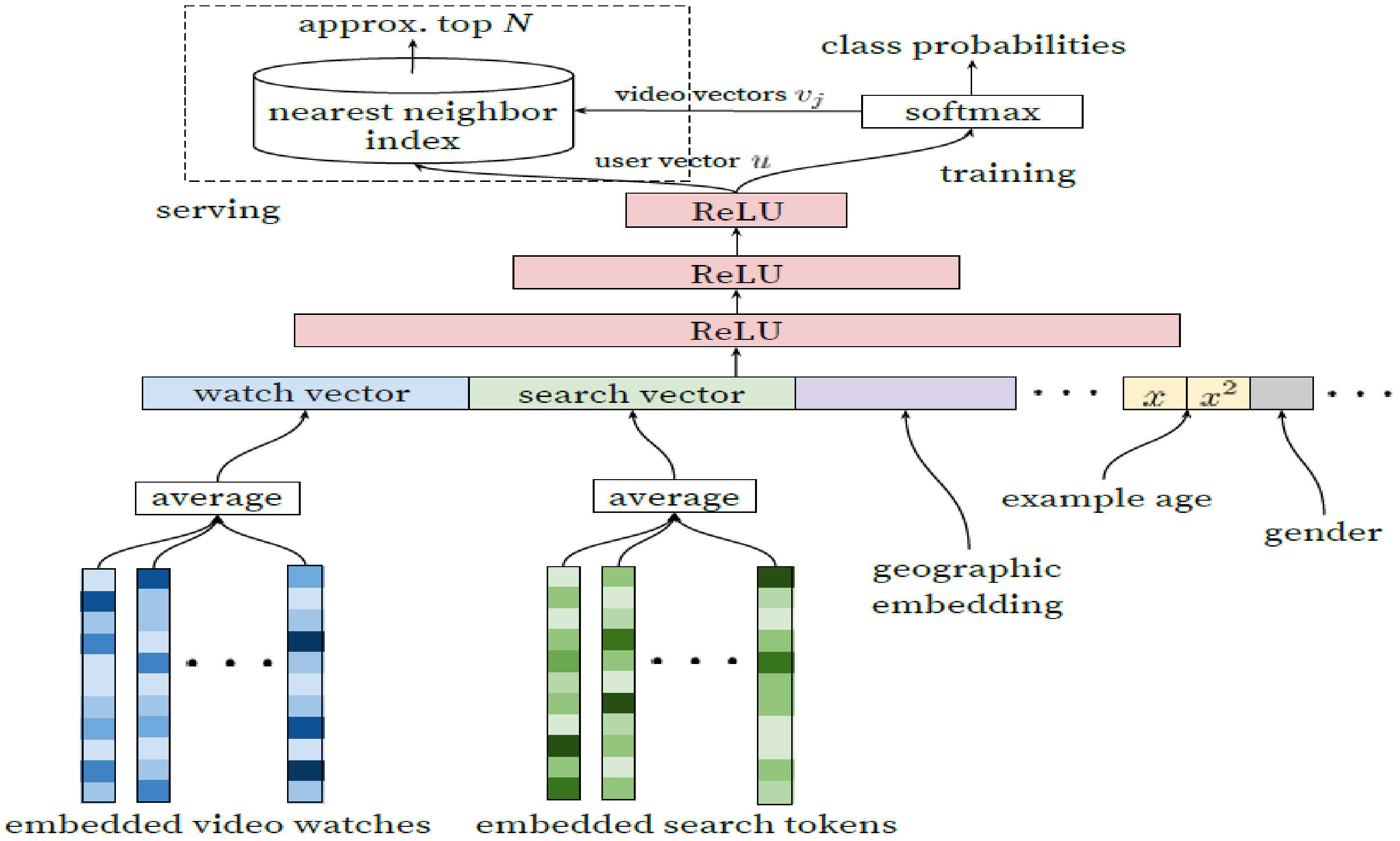
- Using various features…

Overview


user history and context
video
corpus
millions
candidate
generation
hundreds
ranking
dozens
other candidate sources
video
features

Candidate Generation

- 1m videos -> hundreds



Candidate Generation

- Output : user vector(256 dim)

- Classify with softmax with negative sampling
  sampling by importance

- Training : cross entropy

- Generate a fixed number of training examples.

Candidate Generation

- Watch history
  Fixed vocabulary -> embedding vectors
  Average sequence of watched videos.

- Search history
  tokenize query into unigrams and bigrams and embed, average

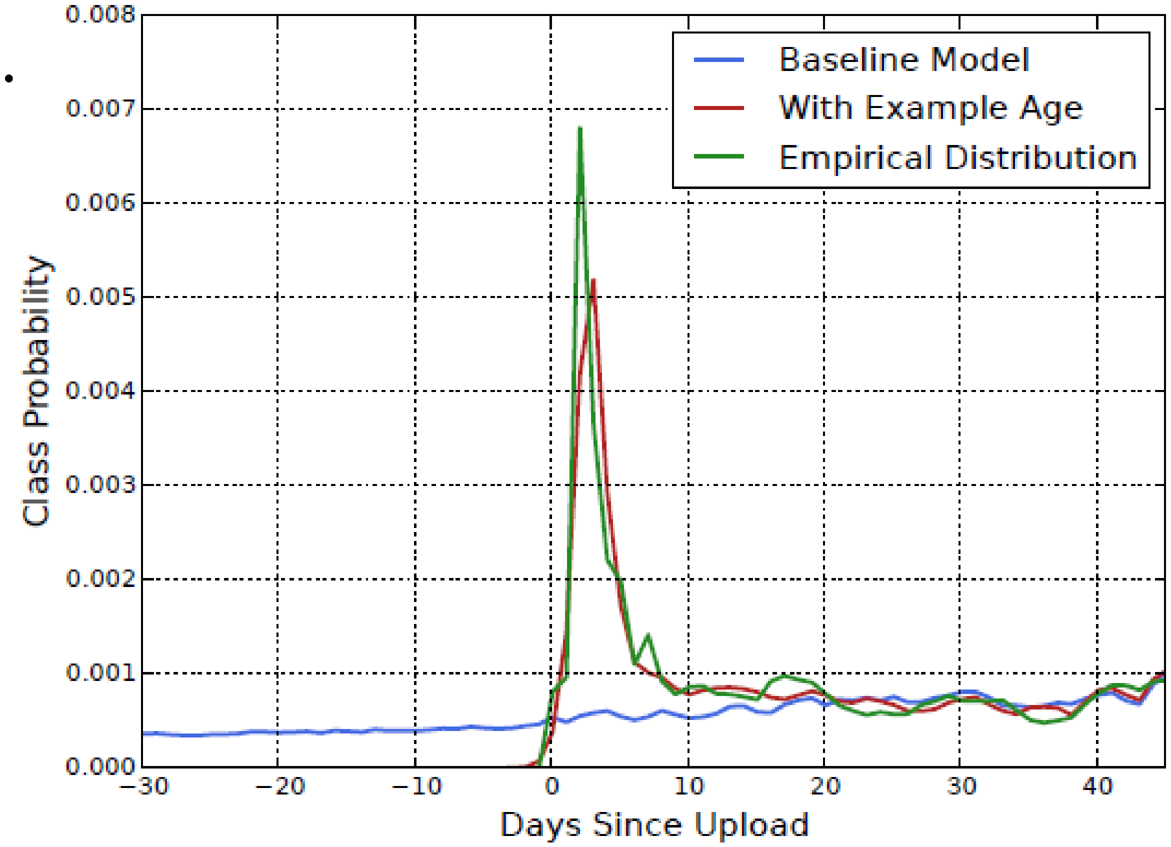
- More embedding :
  region, device…

- Categorical features : gender
- Continuous features[0, 1] : age

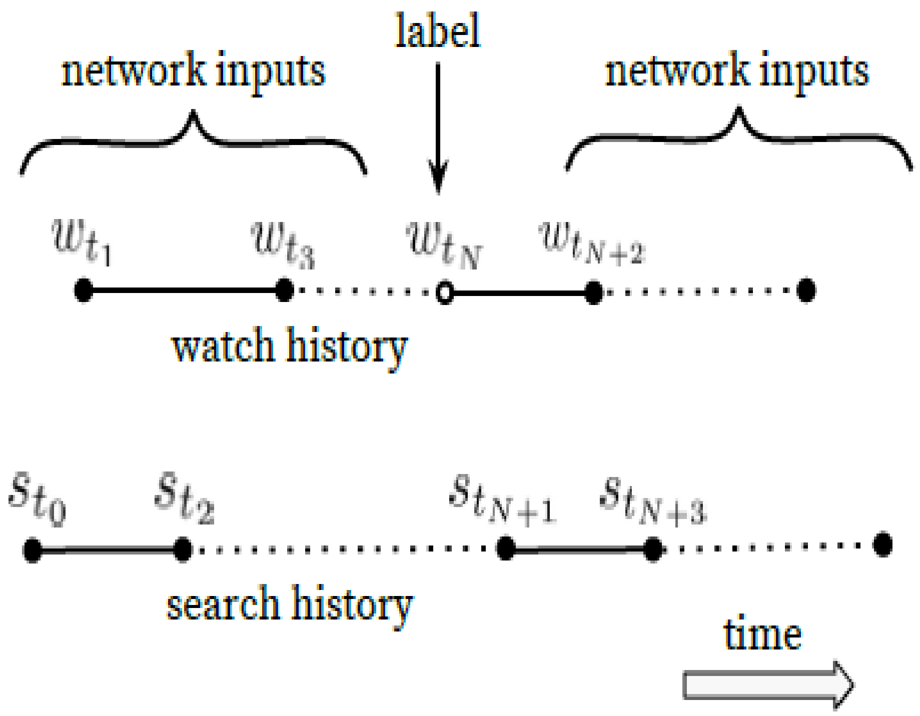
Candidate Generation

- Users prefer fresh content… But, bias towards the past

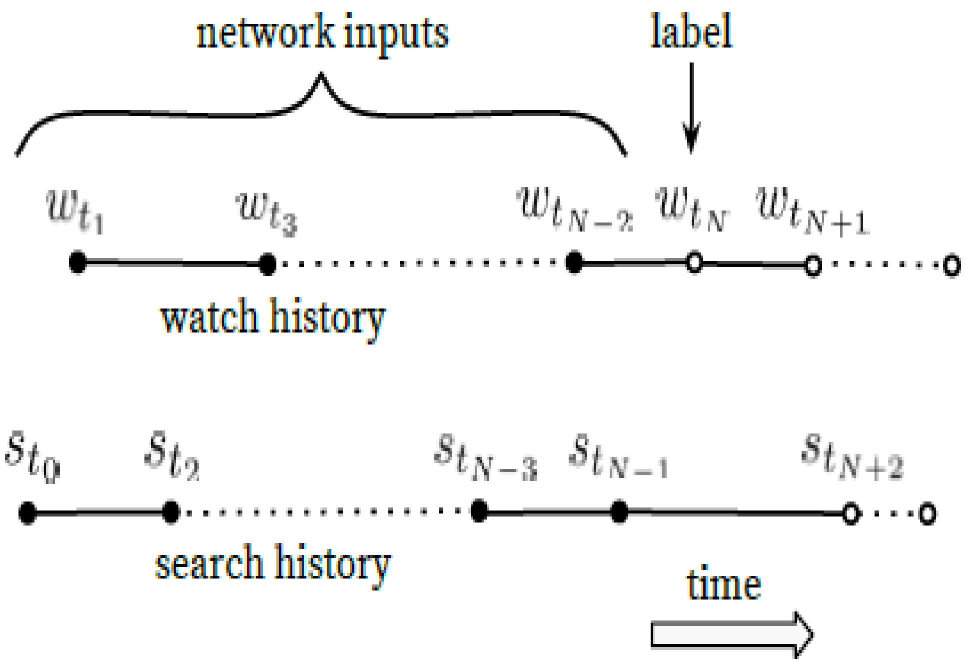
- Feed the age of training example



Candidate Generation



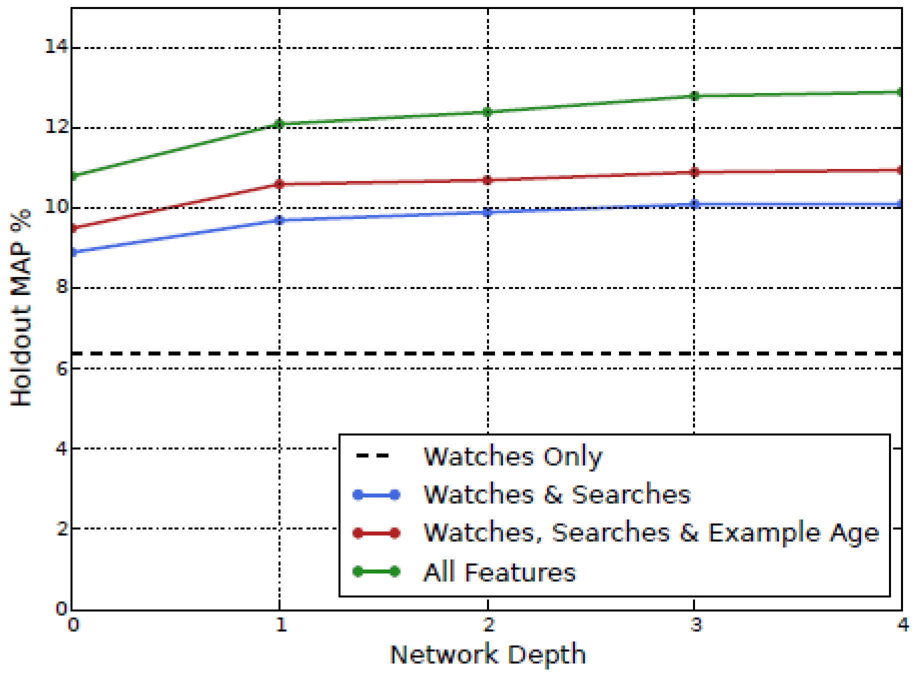
(a) Predicting held-out watch



(b) Predicting future watch

Candidate Generation

- 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



MAP : Mean Average Precision

Candidate Generation

- Average Precision :
  Recall을 늘려가며 Precision의 Average를 구한다.

- Mean Average Precision :
  모든 추천 결과에 대한 AP들의 평균

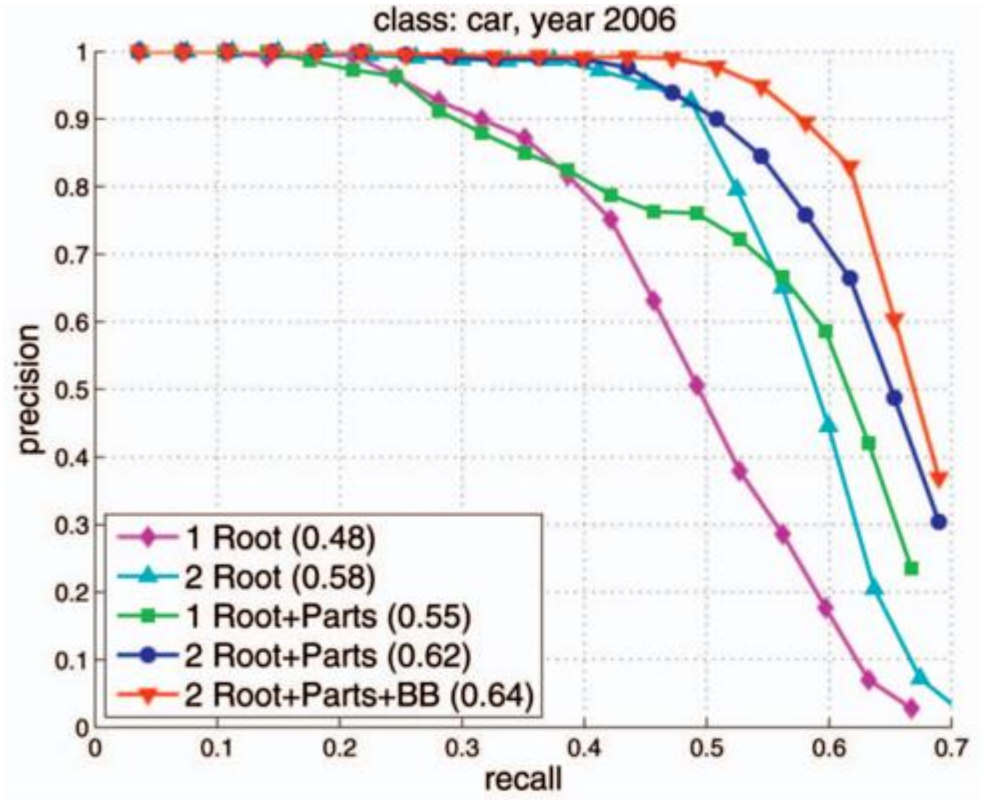
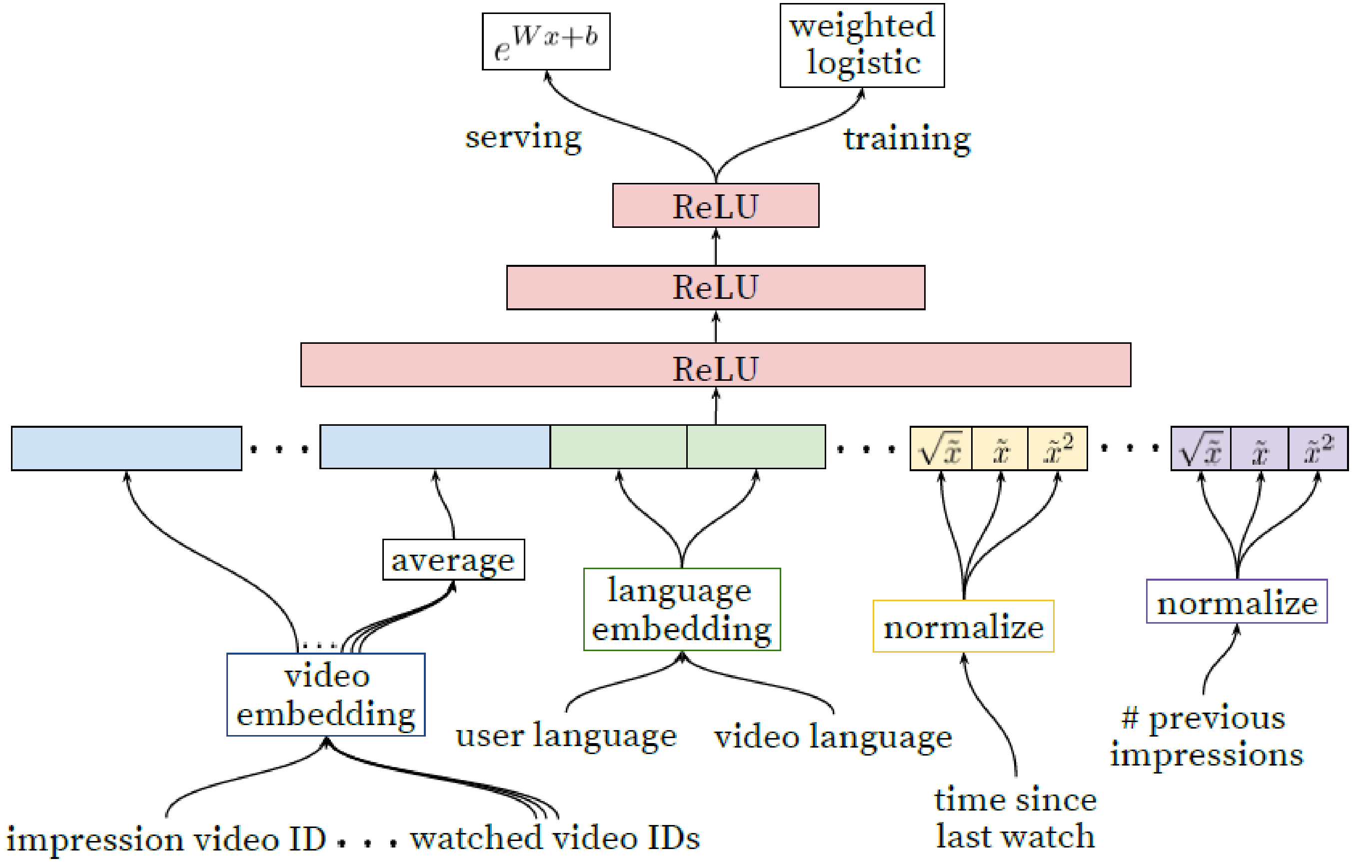


그림 1. precision-recall 그래프의 예

Ranking



직접 사용된 input들
Impression video : 노출된 동영상
중요한 동영상...
바로 전에 본 동영상
추천 결과로 나왔지만 보지 않은 동영상 등등...

루트, 제곱 등의 사용 이유:
무슨 값을 넣어야 좋을 지 모르겠으니 신경망이 알아서 좋은 걸 쓰겠지.
+
신경망이 값을 필요한 형태로 변환하지 않도록 전처리

Ranking

- 예측값 : expected watch time
  e^(Wx+b : output)

- Training with weighted logistic regression under cross-entropy.

Ranking

- If negative impression > positive impression,

  watch time of positive impression -> mispredicted watch time

- Mispredicted watch time / Total watch time

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%

Conclusion

- Matrix factorization보다 좋은 성능을 보였다.
- 과학이 아니라 예술에 가깝다…
  파라미터 설계, feature 선택
- 영상의 나이가 중요하다.
- 시청 시간을 예측하는 것이 클릭률을 예측하는 것보다 좋았다.

- Thank you.