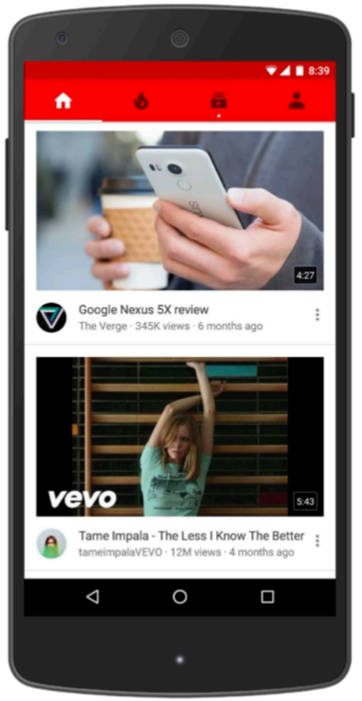
Deep Neural Networks for YouTube Recommendations

Paul Covington, Jay Adams, Emre Sargin (2016)

박민주

Motivation

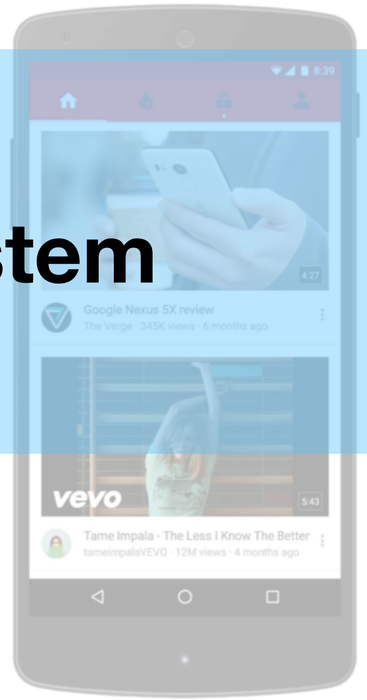
- **Recommending YouTube videos is extremely challenging**
  - **Scale**
    need to handle YouTube's massive user base and corpus
  - **Freshness**
    need to model newly uploaded content as well as the latest actions taken by the user (exploration / exploitation)
  - **Noise**
    need to be robust to sparsity, noisy implicit feedback, poorly structured content



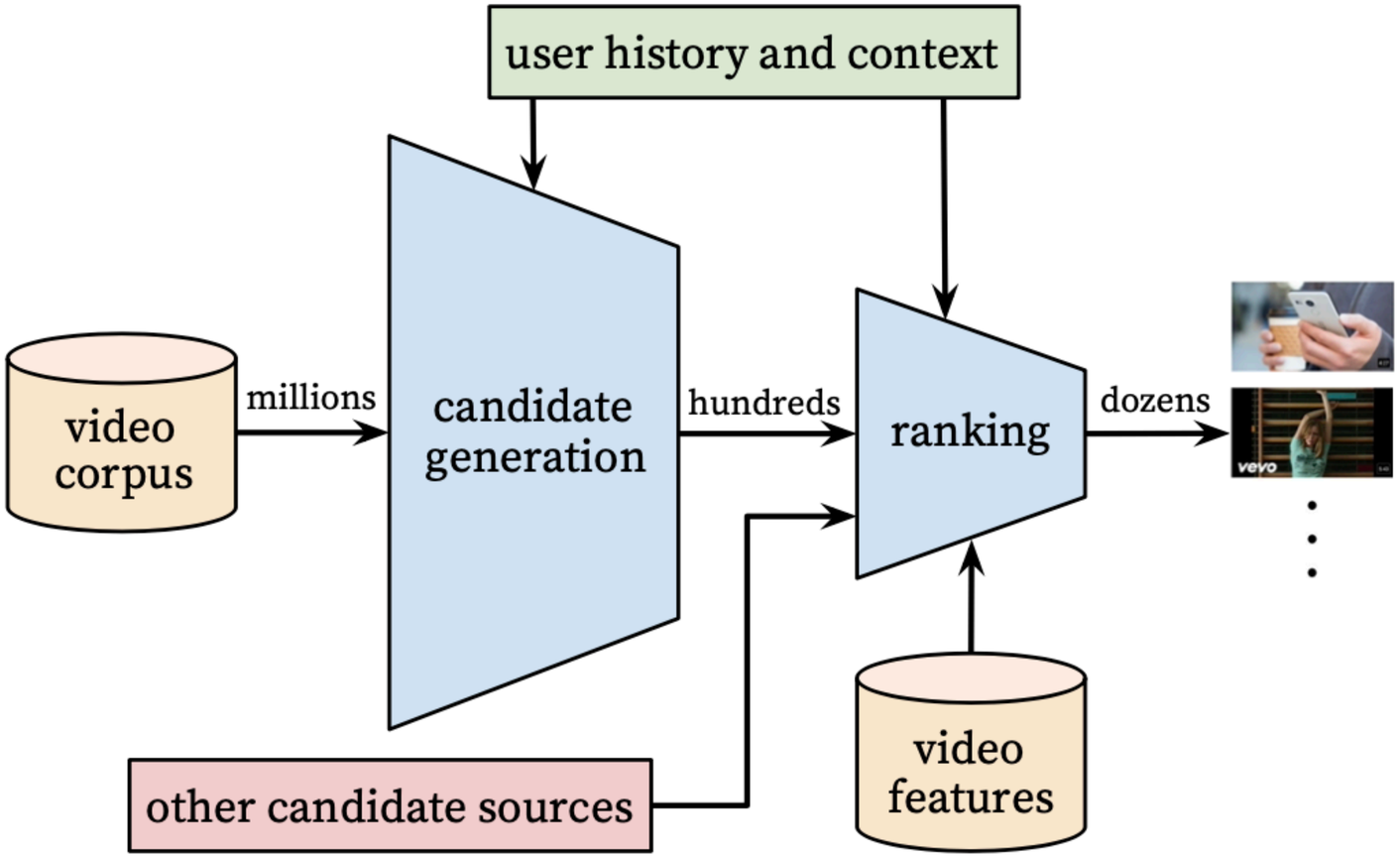
Motivation

- **Recommending YouTube videos is extremely challenging**
  - **Scale**
    need to handle YouTube's massive user base and corpus
  - **Freshness**
    need to model newly uploaded content as well as the latest actions taken by the user (exploration / exploitation)
  - **Noise**
    need to be robust to sparsity, noisy implicit feedback, poorly structured content

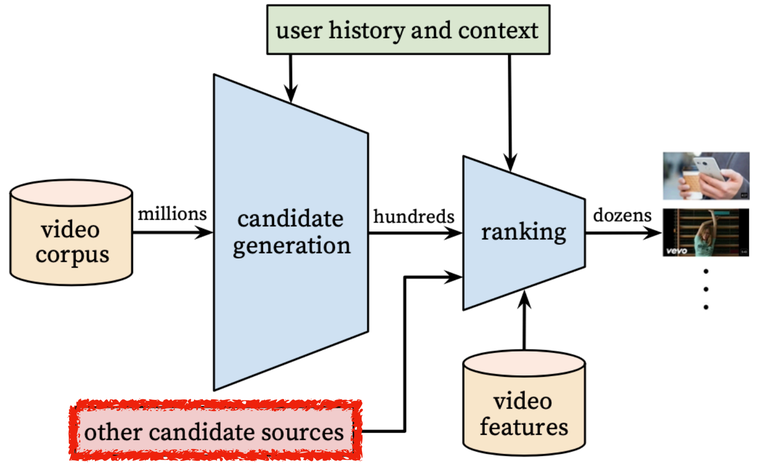
Deep neural network for recommendation system



System Overview



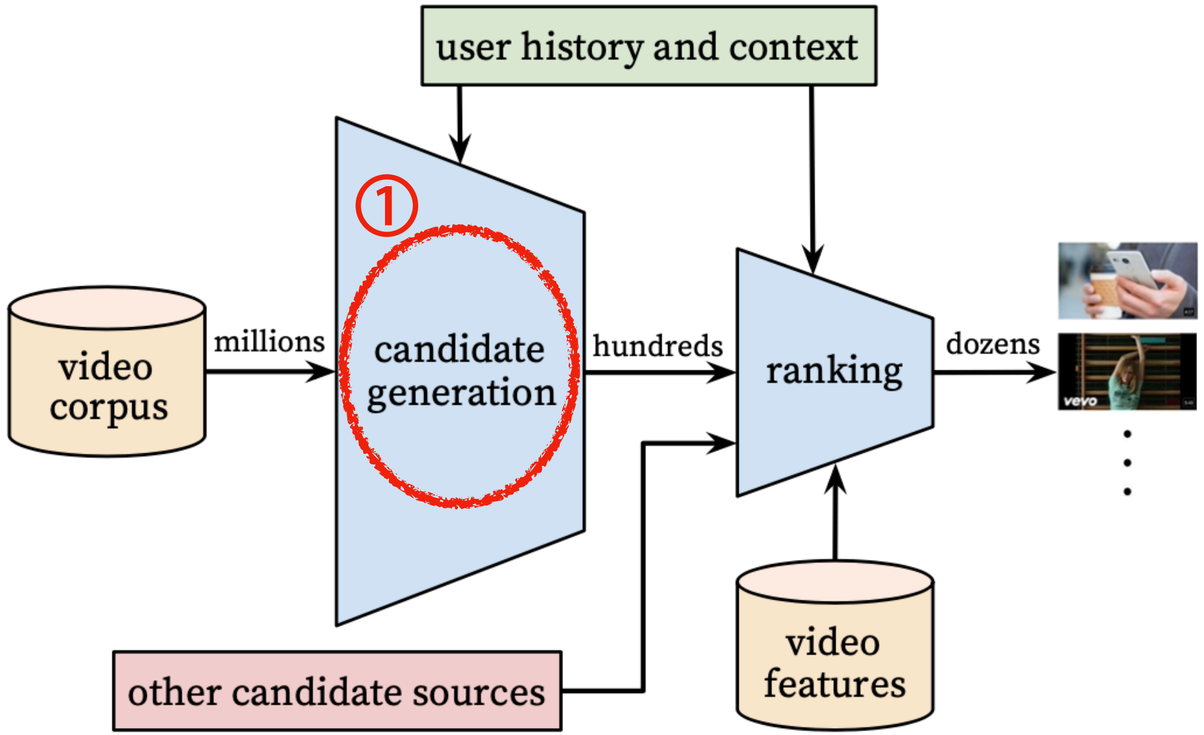
System Overview



- **Why "*Candidate generation → Ranking*" ?**
  - Scalability
  - Easy to blend candidates generated by other sources
  - Easy to debug

- **During development**
  - Use offline metrics (precision, recall, ranking loss, etc)
  - For the final determination, rely on A/B testing (CTR, watch time, etc)

① Candidate Generation



User's YouTube activity history

↓ CF * high precision

Small subset of videos from a large corpus

① Candidate Generation

- **Recommendation as classification**
  - Classify specific video watch w_t at time t

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

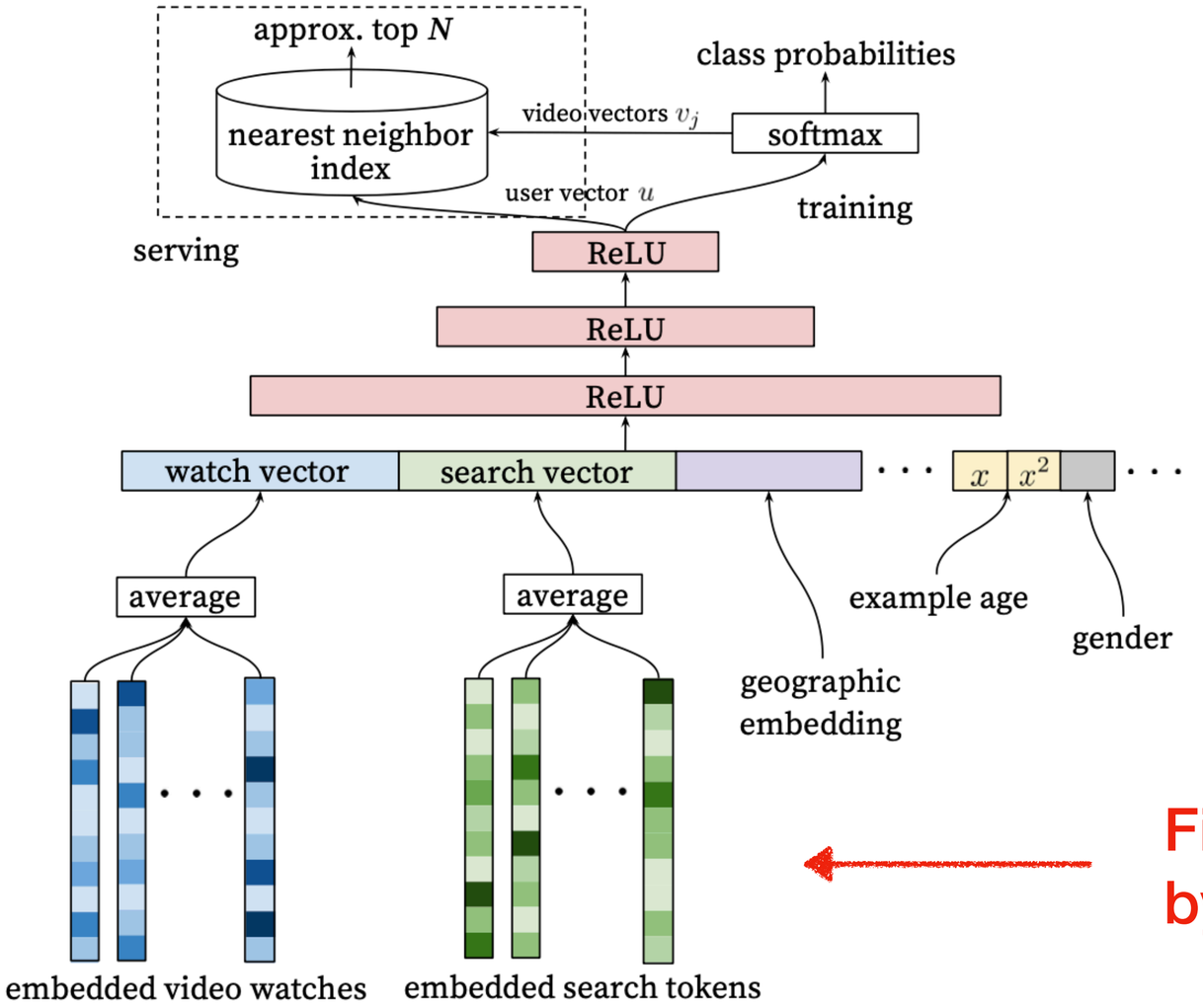
V: corpus
U: user
C: context
$u \in \mathbb{R}^N$: high-dimensional embedding of the user, context pair
$v_j \in \mathbb{R}^N$: embeddings of each candidate video

  - The task is to learn user embeddings u
  - Use implicit feedback to train the model (completing a video = positive)

① Candidate Generation

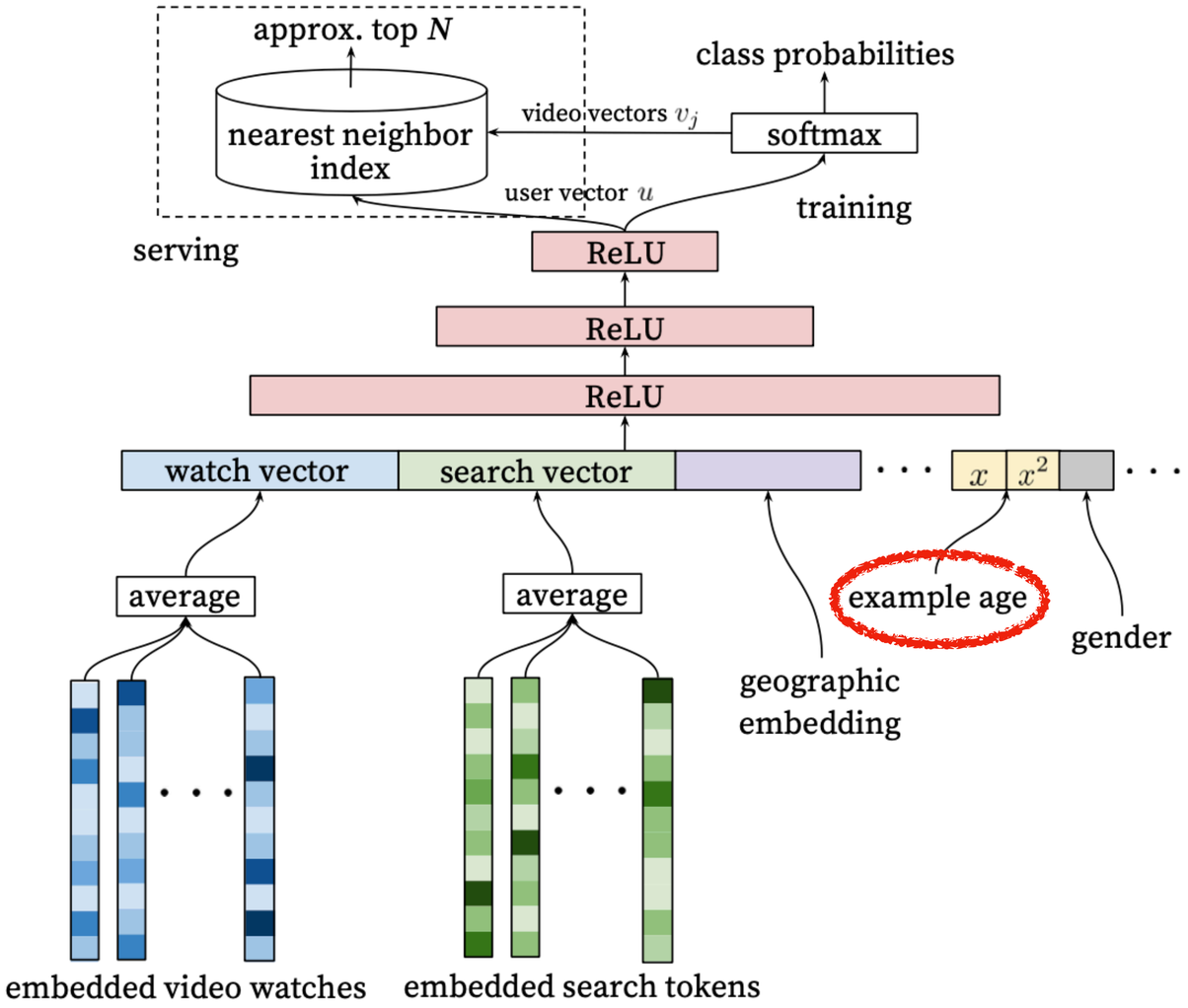
- **Recommendation as classification**
  - **Model training**
    - Candidate sampling (negative sampling)
    - Cross-entropy loss for the true label and sampled negative samples
  - **Model serving**
    - Approximate kNN search on output space

① Candidate Generation



Fixed-sized dense inputs
by averaging the embeddings

① Candidate Generation



- **"Example Age" Feature**

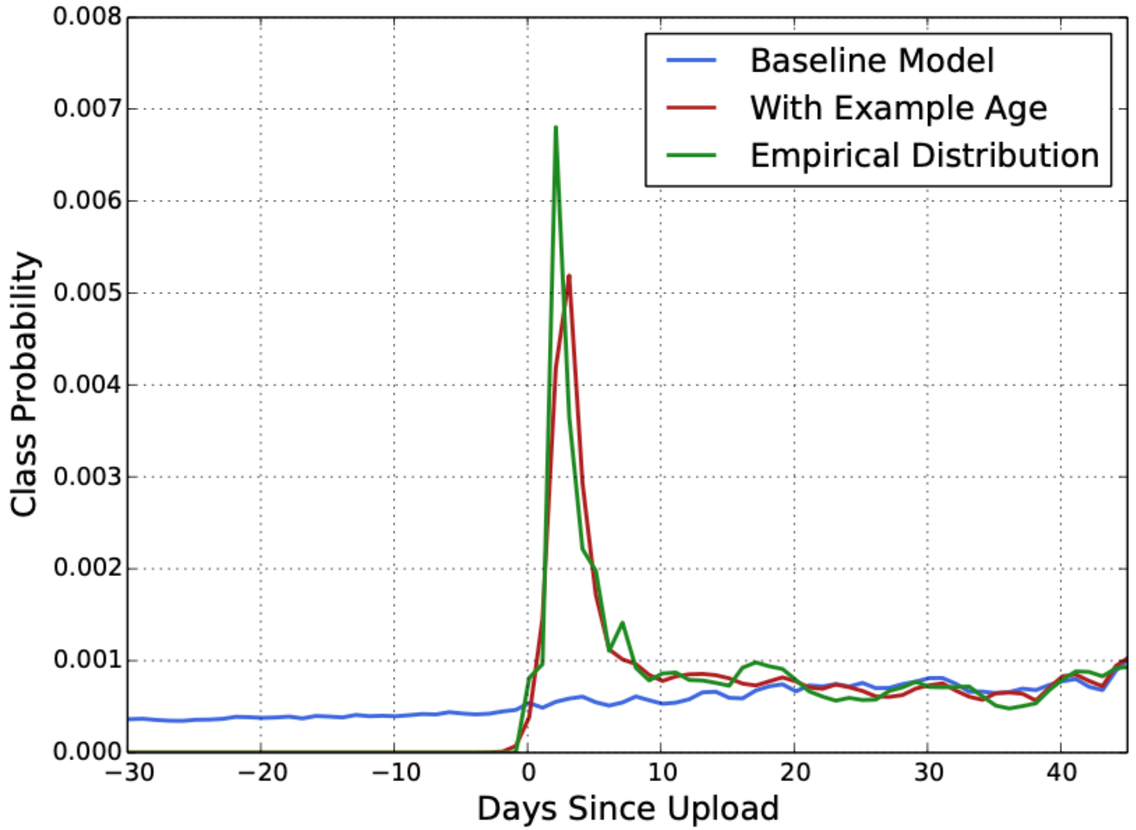
  - Recommending recently uploaded content is important for YouTube

  - but machine learning systems are trained to predict future behavior from historical examples

➡ feed the age of the training example

① Candidate Generation

- **“Example Age” Feature**



① Candidate Generation

- **Label and context selection**
  - **Label**

    generated from all YouTube watches (even those on other sites)

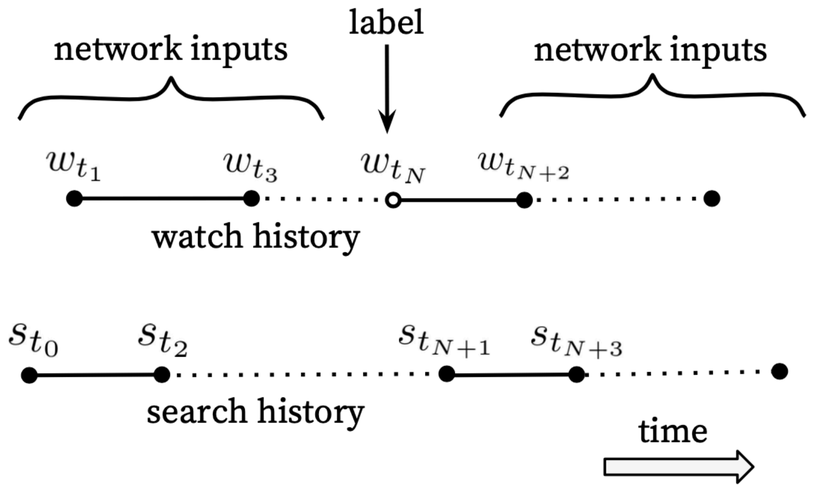
    → prevent exploitation and overfitting

    * generate fixed number of training examples per user

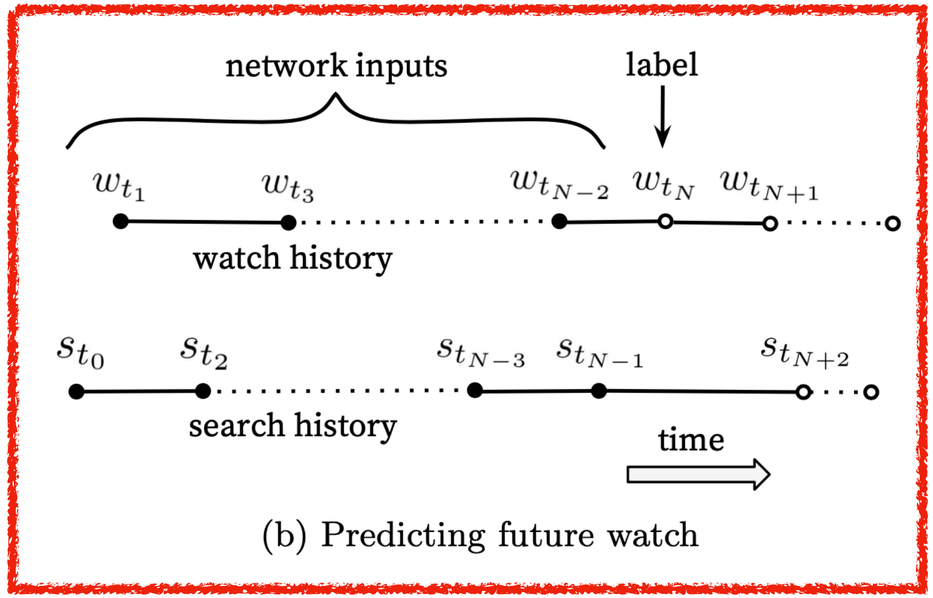
① Candidate Generation

- **Label and context selection**
  - **Context**

    predict the user's next watch rather than predicting a random watch



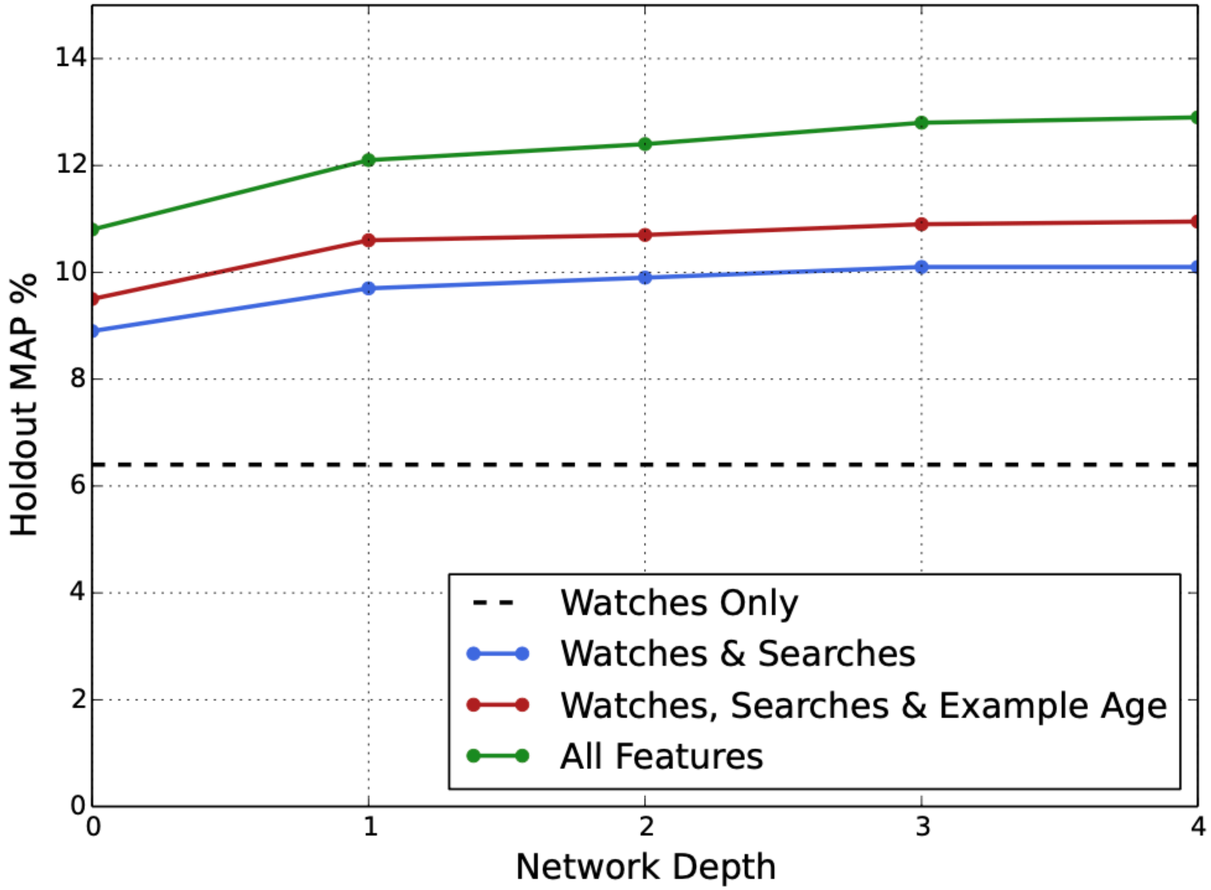
(a) Predicting held-out watch



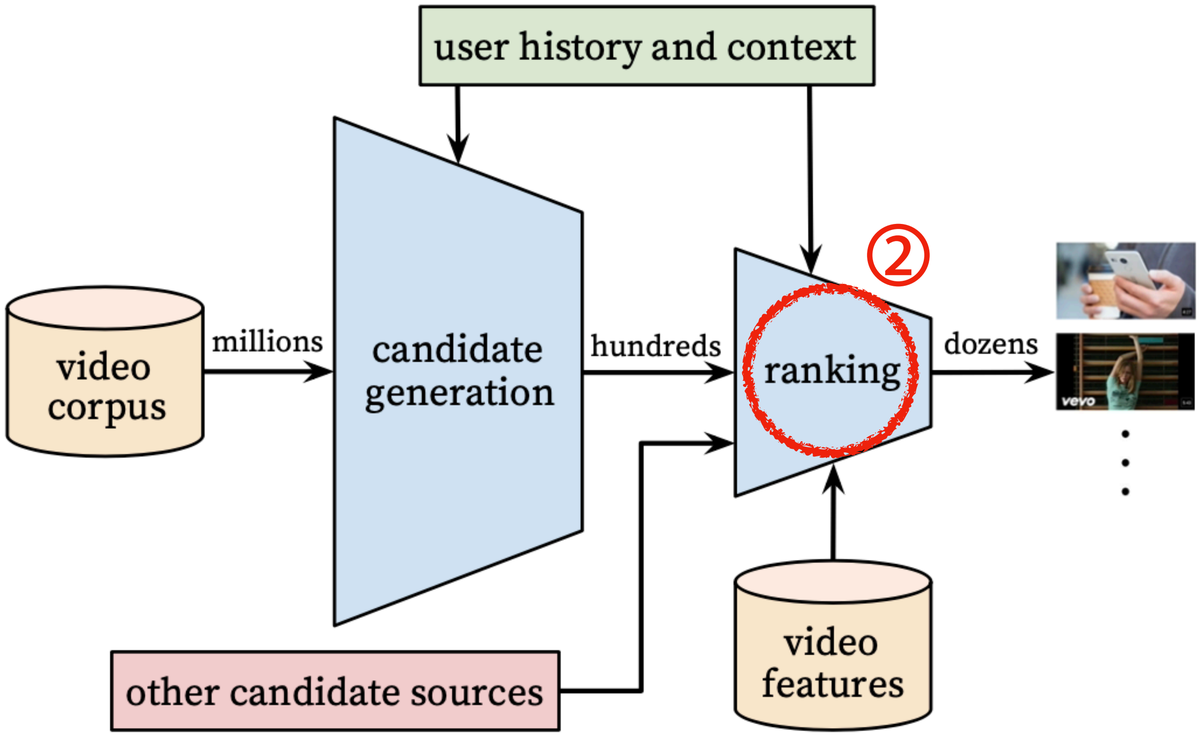
(b) Predicting future watch

① Candidate Generation

- **Experiment**



② Ranking



Rich set of features

Objective function

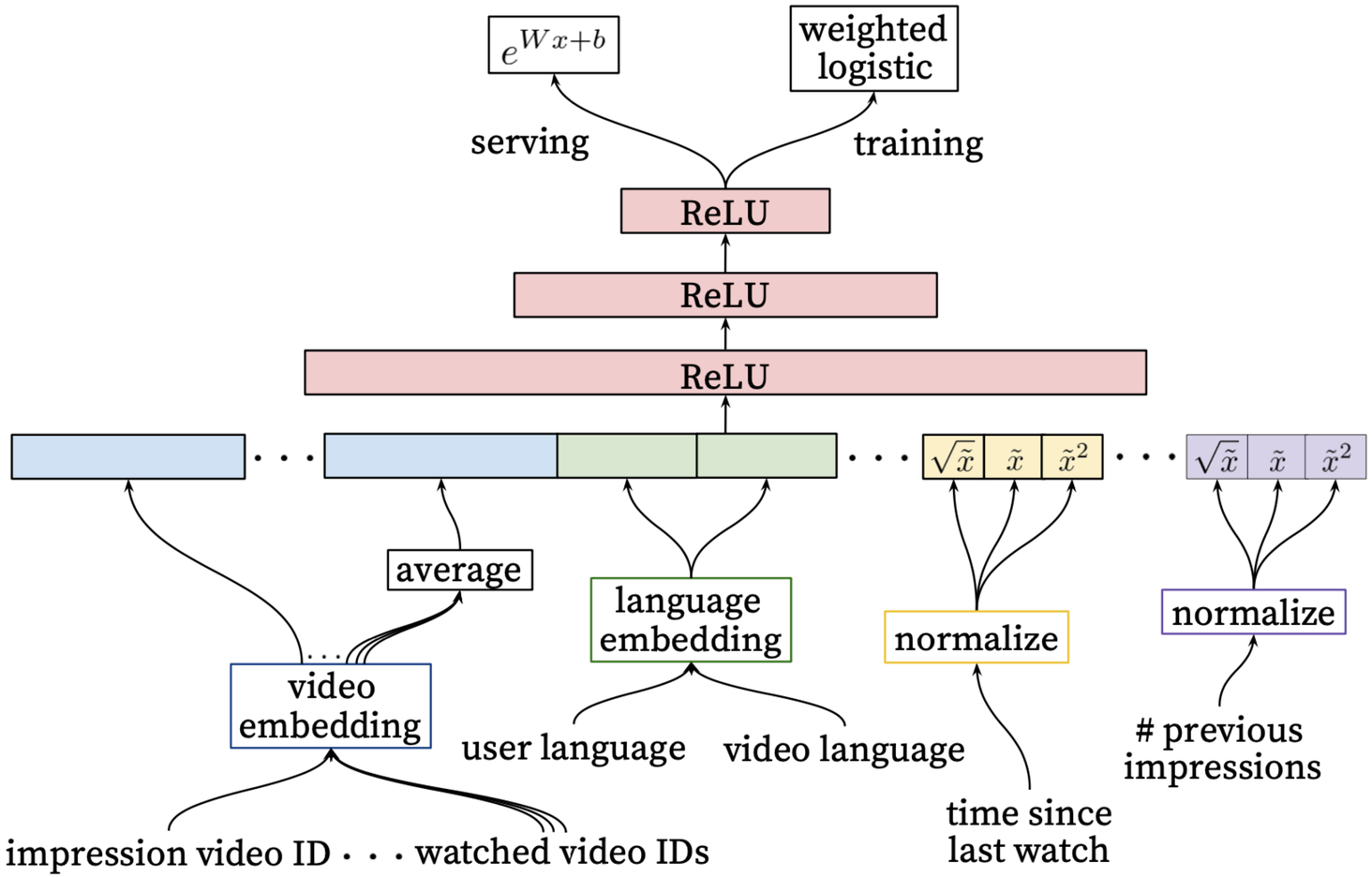
* high recall

Assign a score to each video

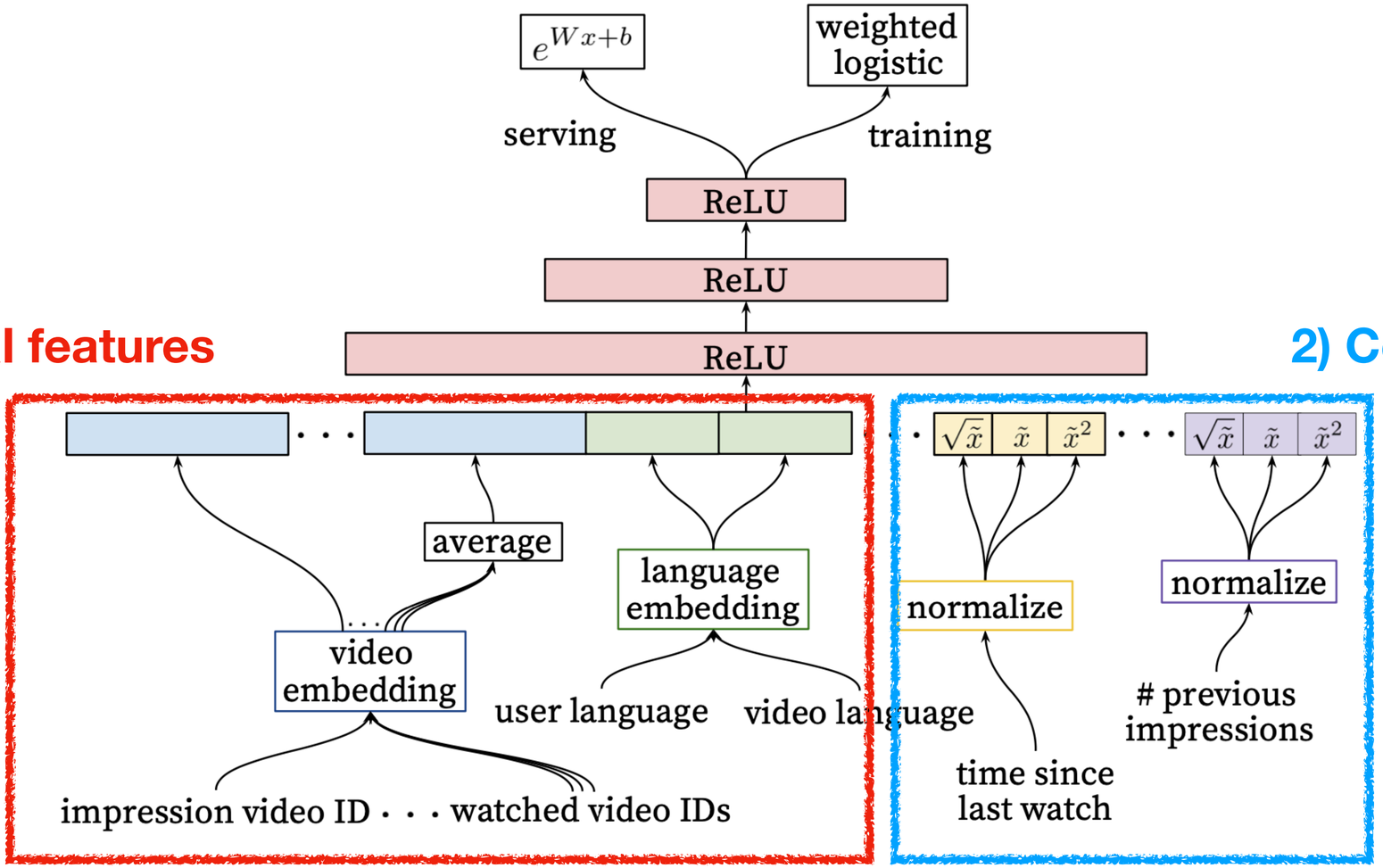
② Ranking

- **Feature representation**
  - Main challenge is in representing a temporal sequence of user actions and how these actions relate to the video impression being scored
  - Observations
    - User's previous interactions are the most important signals
    - Crucial to propagate information from candidate generation into ranking in the form of features
    - Features describing the frequency of past video impressions are critical

② Ranking

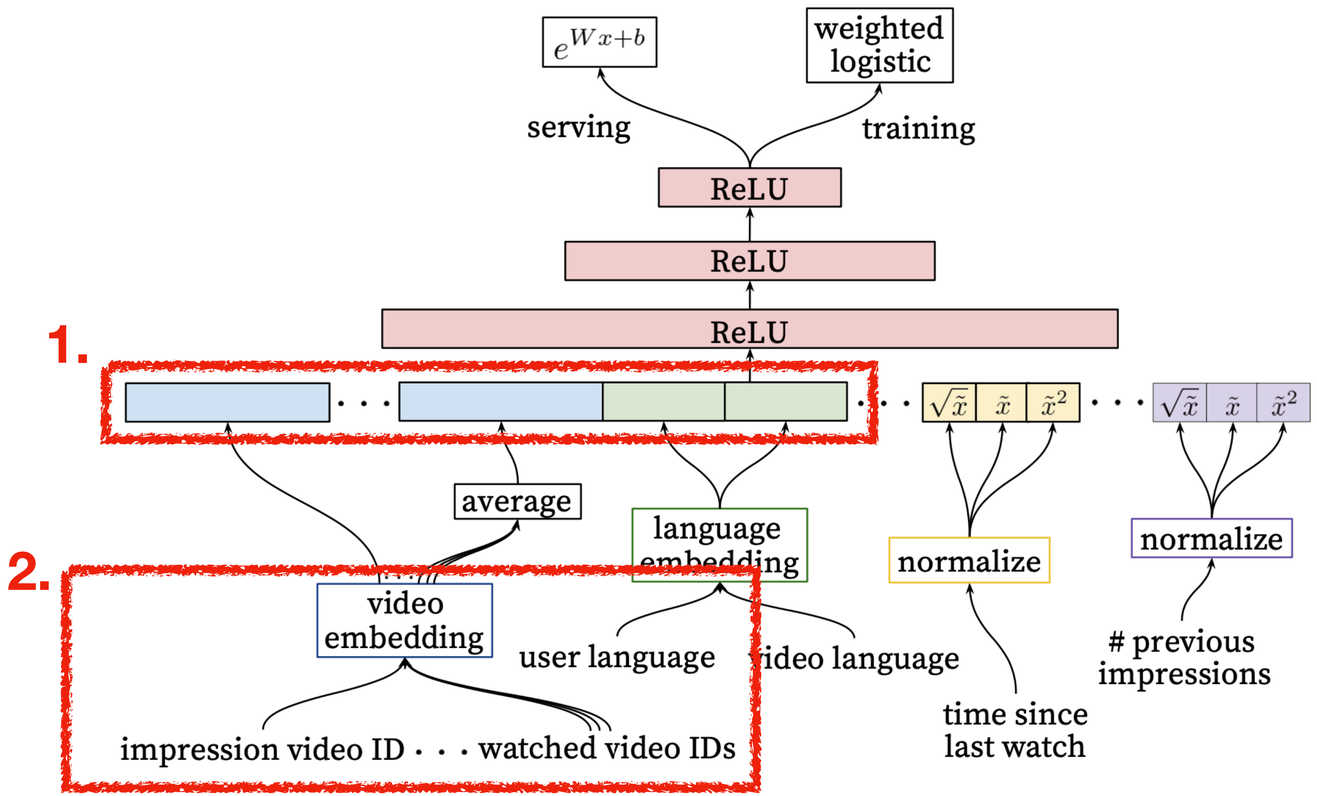

e^{Wx+b}
weighted logistic
serving
training
ReLU
ReLU
ReLU
$\sqrt{\tilde{x}}$ $\tilde{x}$ $\tilde{x}^2$
$\sqrt{\tilde{x}}$ $\tilde{x}$ $\tilde{x}^2$
average
language embedding
normalize
normalize
video embedding
user language
video language
previous impressions
time since last watch
impression video ID
watched video IDs

② Ranking


e^{Wx+b}
weighted logistic
serving
training
ReLU
ReLU
ReLU
1) Categorical features
2) Continuous features
average
language embedding
video embedding
user language
video language
impression video ID
watched video IDs
normalize
normalize
time since last watch
previous impressions

② Ranking

- **Feature representation 1) Categorical features**

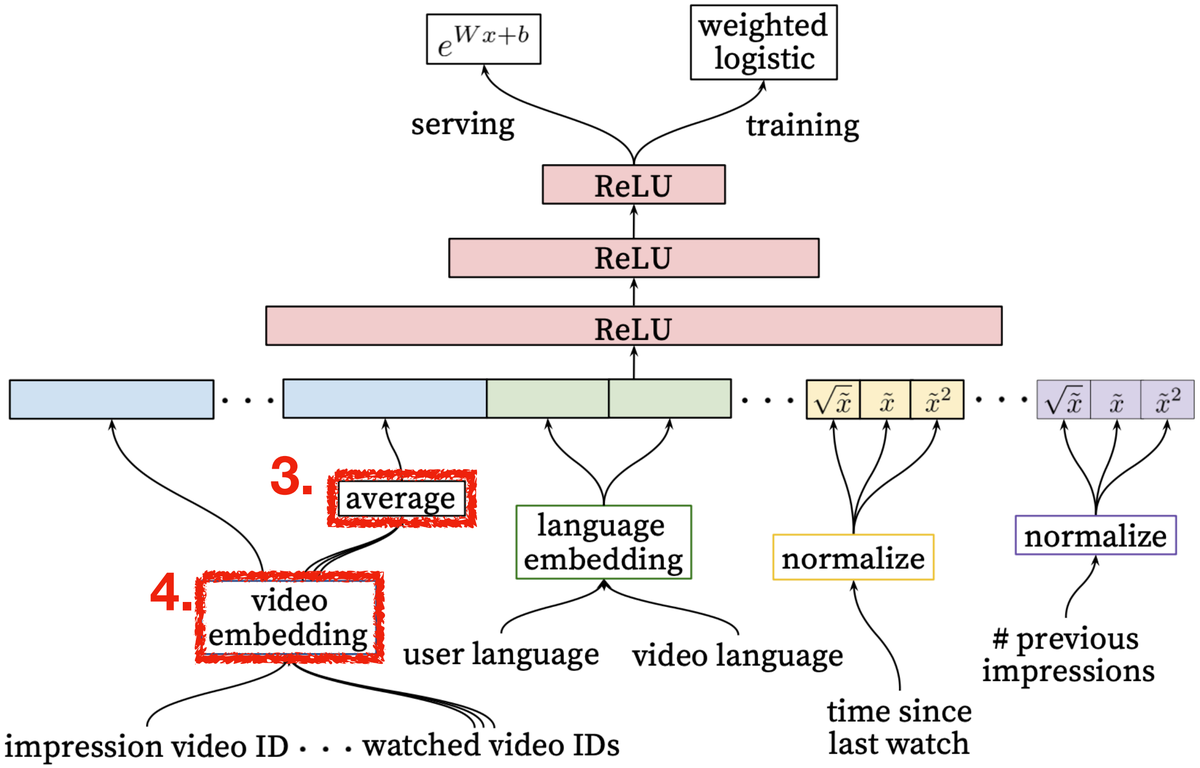


1. Dense embeddings

2. Large cardinality ID spaces are truncated based on their frequency in clicked impressions, and out-of vocabulary values are mapped to the zero embedding

② Ranking

- **Feature representation 1) Categorical features**

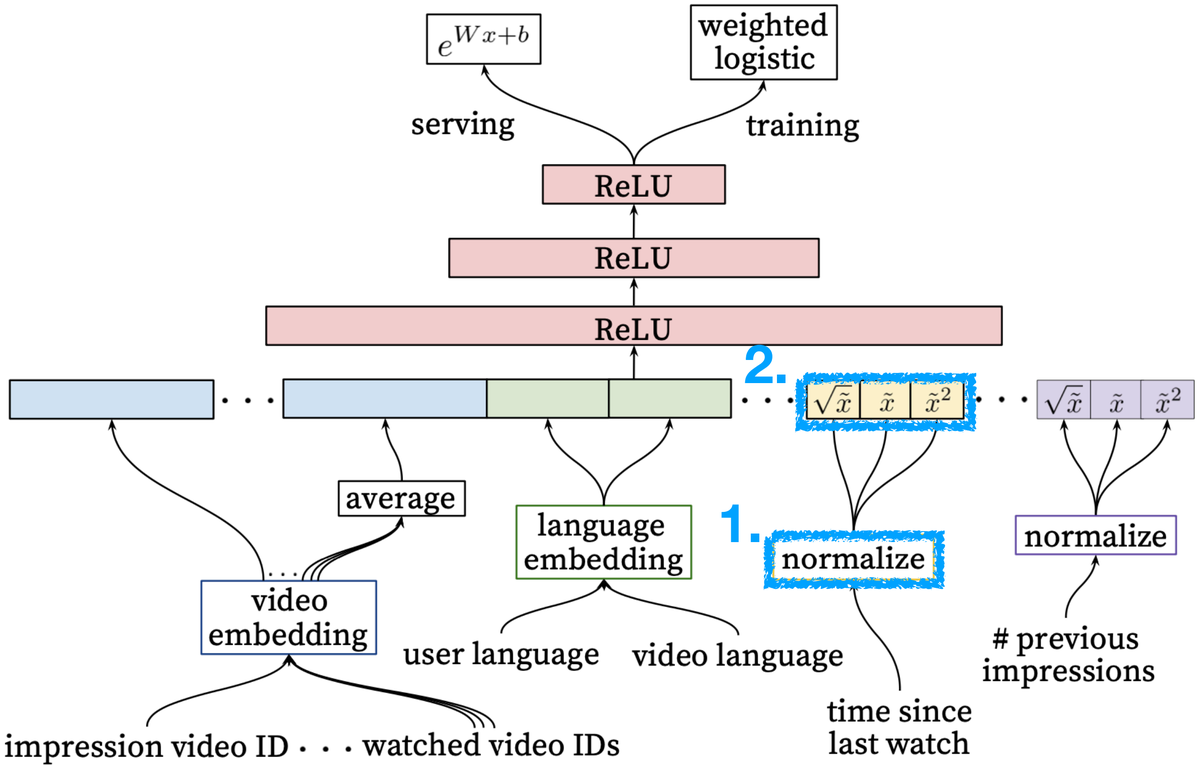


3. Multivalent categorical feature embeddings are averaged before being fed

4. ID space share embeddings but each feature is fed separately

② Ranking

- **Feature representation 2) Continuous features**



1. Normalize continuous feature x into $\tilde{x} \in [0,1)$

2. Also input $\tilde{x}^2$ and $\sqrt{\tilde{x}}$ giving the network more expressive power

② Ranking

- **Modeling expected watch time**
  - Goal is to predict expected watch time given positive / negative training examples

  - **Weighted logistic regression**

    Logistic regression under cross-entropy loss + weighted by watch time

    (Negative impressions receive unit weight)

② Ranking

- **Experiment**
  - Increasing the width and depth of hidden layers improve results

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%

Conclusions

- Deep neural network architecture for recommending YouTube videos (candidate generation → ranking)
- Outperforms previous matrix factorization approaches used at YouTube
- “Example age” helps representing the time-dependent behavior of popular videos
- Recommendation systems benefit from specialized features describing past user behavior with items
- Weighted logistic regression performed much better than predicting CTR

Thank you