Deep Neural Networks for YouTube Recommendations

Ömer Davarcı
Ömer Faruk Merey

YouTube Recommendations

Scale

Data on disk and serving latency

Freshness

Corpus constantly growing and non-stationary

Noise

Implicit feedback and unstructured metadata

User historical content is hard to predict.

System Overview

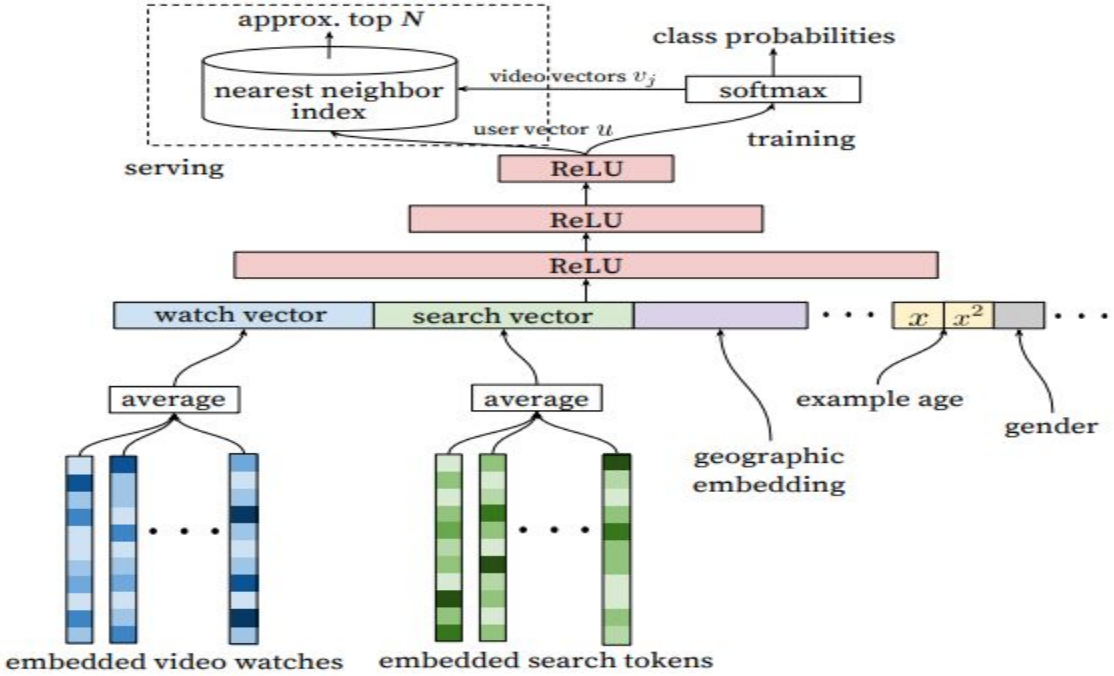
1. System consists of two neural network.
    1.1 Candidate Generation - Use filters and narrow the scope
        1.1.1 Collaborative Filtering - Similarity to other users
    1.2 Ranking - Scoring the videos and selecting the best n of them.

Candidate Generation

1. Recommendation as Classification
   1.1 Explicit and Implicit Feedback Mechanisms
2. Efficient Extreme Classification
   2.1 Candidate Sampling and Importance Weighting

Candidate Generation Architecture

Model Architecture



“Example Age” Feature

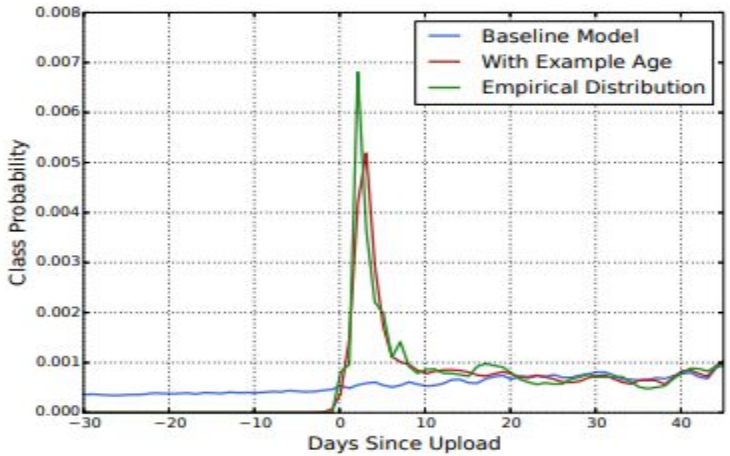
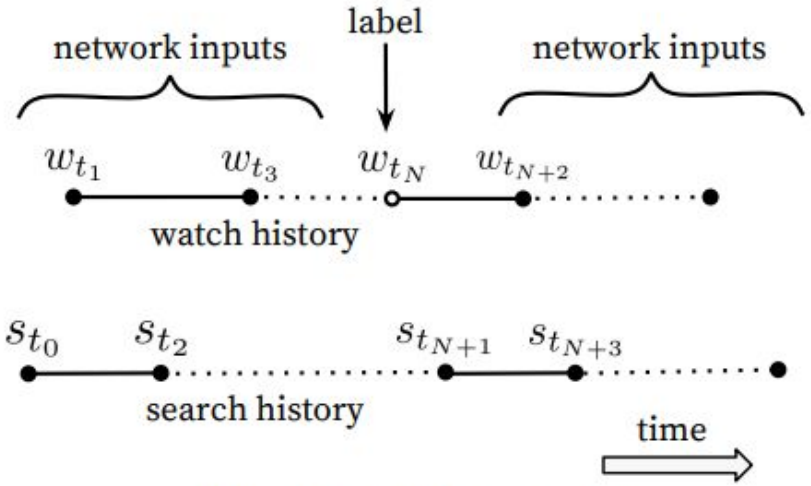


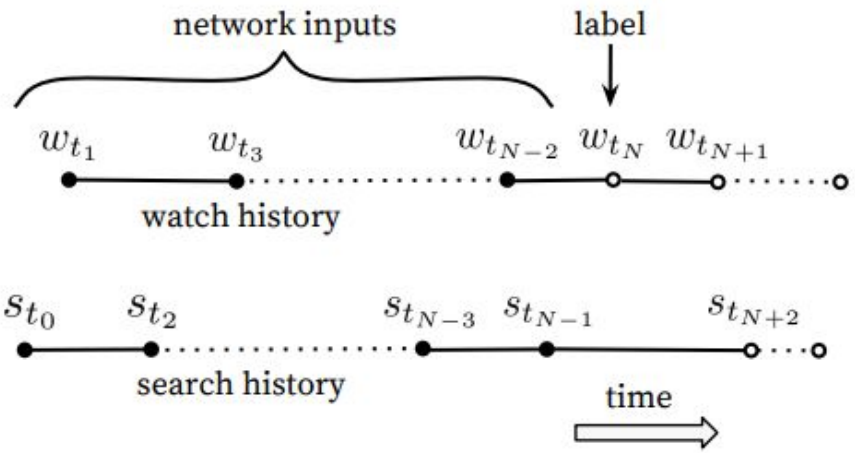
Figure 4: For a given video [26], the model trained with example age as a feature is able to accurately represent the upload time and time-dependant popularity observed in the data. Without the feature, the model would predict approximately the average likelihood over the training window.

Label and Context Selection

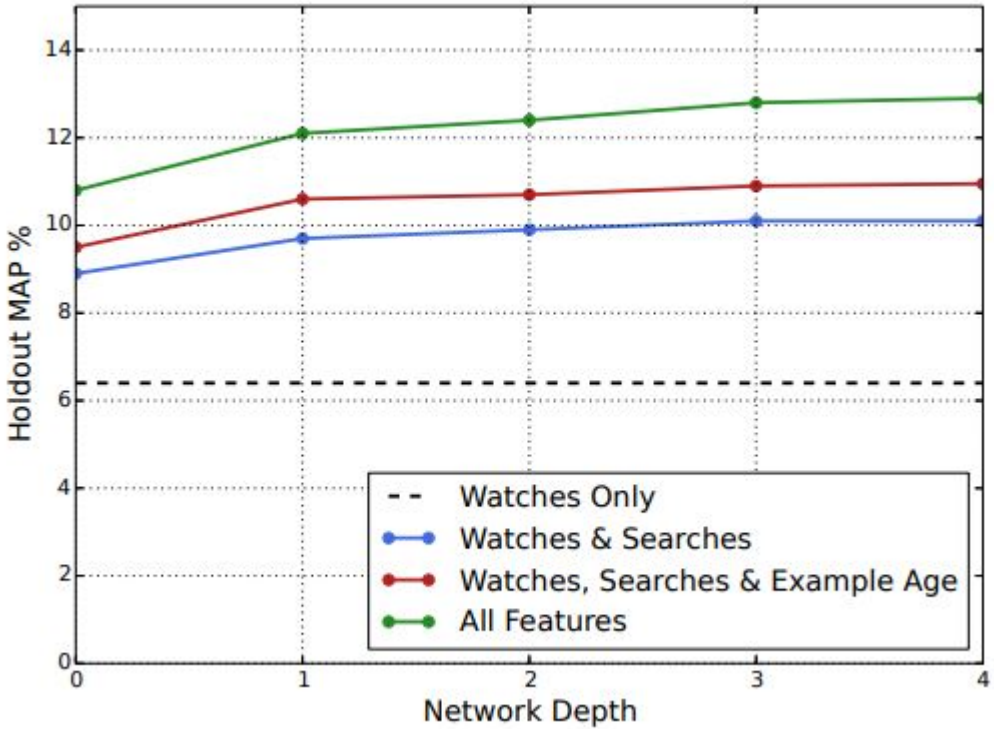
1. Episodic videos and new genres.



(a) Predicting held-out watch



(b) Predicting future watch


Holdout MAP %
Network Depth
0
2
4
6
8
10
12
14
0
1
2
3
4
Watches Only
Watches & Searches
Watches, Searches & Example Age
All Features

Ranking

Has a similar structure.

Feature presentation:

- Binary
- Univalent
- Multivalent

More tailor features.

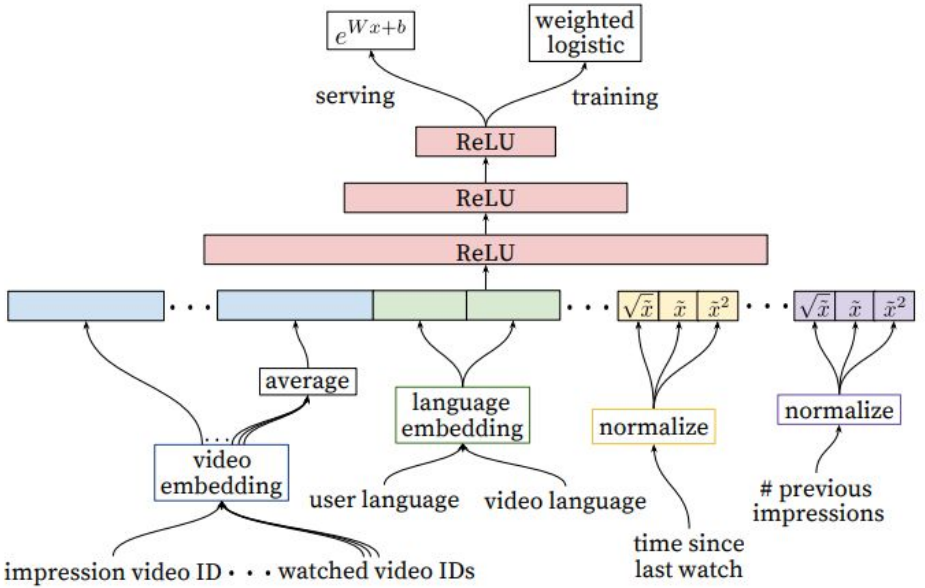


Figure 7: Deep ranking network architecture depicting embedded categorical features (both univalent and multivalent) with shared embeddings and powers of normalized continuous features. All layers are fully connected. In practice, hundreds of features are fed into the network.

Expected Watch Time

We do not use clicks of the video!

But why?... Clickbaits.

The importance of using the right data!

Training

Weighted Logistic Regression

Watch time is used for positive interactions.

Unit weight for negative interactions.

Since the number of positive interactions are much smaller than negative interactions it gives a great result.

Log-odds

Serving and Expected Watch Time

Logistic is used to score the videos.

Optimized for expected watch time from training labels.

We do not use click-baits!

Odds.

Feature Engineering

Despite promises of neural network, there is a lots of feature engineering to do.

Summarizing the temporal sequences of sparse actions

Best features are user’s interaction with the item itself.

Hidden Layers Experiment

For negative and positive interactions

Increasing hidden widths and depths

Server CPU time trade-off

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.

Conclusion

Questions

Thank you for listening