

# Paper

## Reference

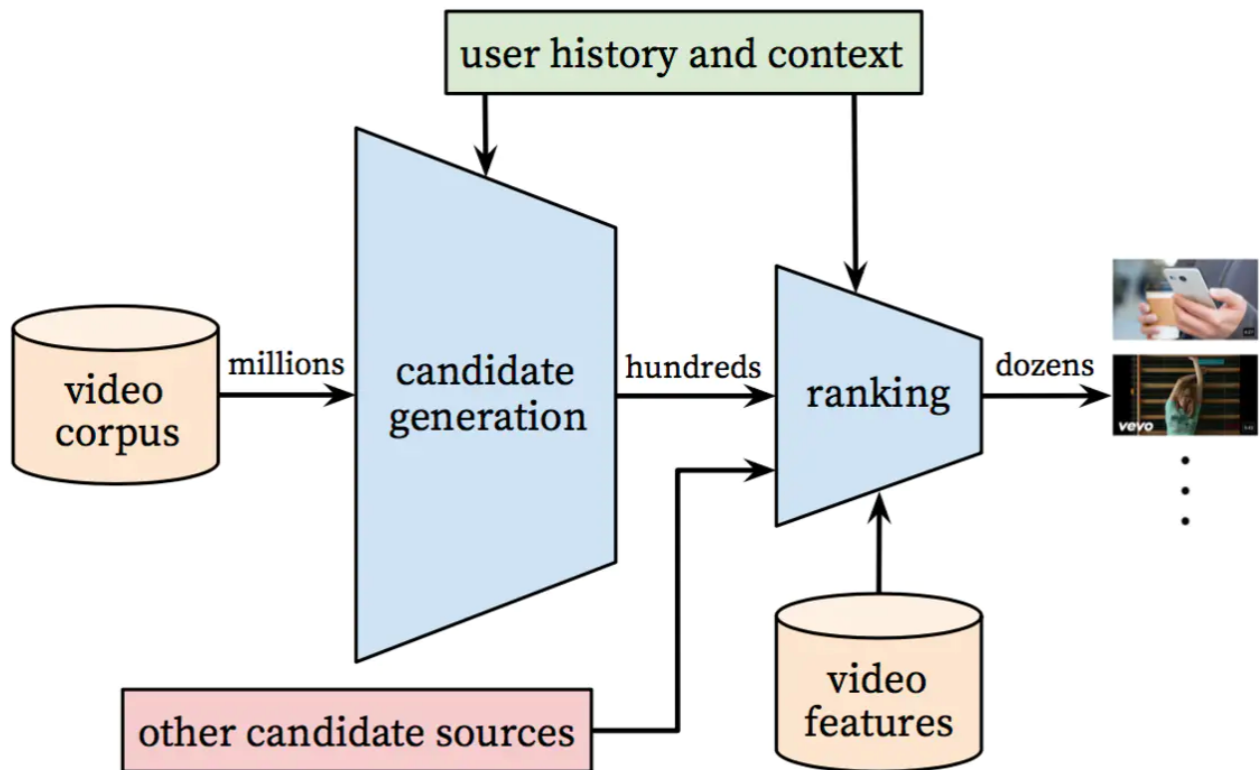
1. [Deep Neural Network for YouTube Recommendation](#)论文精读
2. [DNN for YouTube Recommendations](#) 论文总结
3. [重读Youtube深度学习推荐系统论文，字字珠玑，惊为神文-王喆](#)
4. [YouTube深度学习推荐系统的十大工程问题-王喆](#)
5. [揭开YouTube深度推荐系统模型Serving之谜-王喆](#)
- 6.

## Introduction

1. YouTube's challenge
  1. Scale
    - Highly specialized distributed learning algorithms and efficient serving systems are essential for handling YouTube's massive user base and corpus.
  2. Freshness (冷启动)
    - Trade off between 'fresh' videos and well-established videos
    - an exploration/exploitation perspective(角度)
  3. Noise
    - implicit feedback 和 content feature中都有大量的噪声
2. Scale of model
  - one billion parameters
  - hundreds of billions of examples

## System Overview

### 1. Recommendation system architecture



## 2. Composition

1. A neural network for **candidate generation** and a neural network for **ranking**
2. candidate generation network (粗排/召回)
  - provides broad personalization via *collaborative filtering* (协同过滤)
  - utilize simple features: **ID**, **search query tokens** and **demographics** (人口统计学特征)
3. ranking network (精排/排序)
  - distinguish relative importance among candidates with high recall
4. Metrics 【why not AUC】
  - offline: precision, recall, ranking loss, etc.
  - online: click-through rate (ctr), watch time and many other metrics that measure user engagement (用户参与度) .

## 3. Candidate Generation

1. previous approach
  - FM with rank loss
2. Formulation
  1. a specific video watch  $w_t$
  2. time  $t$
  3. videos  $i$
  4. from a corpus  $V$


5. user  $U$

6. context  $C$

7.  $u \in R^N$  represents a high-dimensional “embedding” of the (user, context) pair

8.  $v_j \in R^N$  represent embeddings of each candidate video

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

 Pasted image 20201013225918.png

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

3. 任务：从用户观看的历史数据和当前上下文中学习用户的embedding  $u$ ，并使用 **softmax**分类器选择最有可能被观看的video（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 among videos with a softmax classifier.）

- 第一反应：用softmax来做推荐在现在看来并不是特别合理（误）
- 后面看懂了：先使用softmax学习video embedding。线上使用时，再与DNN学到的 user embedding 一起通过K近邻计算召回的视频
- **trick**：使用比较宽泛的特征，便于用户的相似性迁移

#### 4. Efficient Extreme Multiclass

- 每次优化正样本和sample一部分负样本（一般选几千个）的交叉熵（For each example the cross-entropy loss is minimized for the true label and the sampled negative classes）。类似word2vec中Skip-Gram的负采样的方法，但word2vec是采样出来做二分类，有本质区别
- 或者使用 hierarchical softmax [F. Morin and Y. Bengio. Hierarchical probabilistic neural network language model.]，但本文在该方法上没能达到较好的精度
- Serving：选出与user embedding 最相似的N个video embeddings
  - 早期方法：hashing（??? 局部敏感哈希（Locality Sensitive Hashing））
  - 对nearest neighbor search algorithm的选择对A/B test结果不敏感
  - serving阶段问题转化为 a nearest neighbor search in **the dot product space** for which general purpose libraries can be used

#### 5. Model Architecture

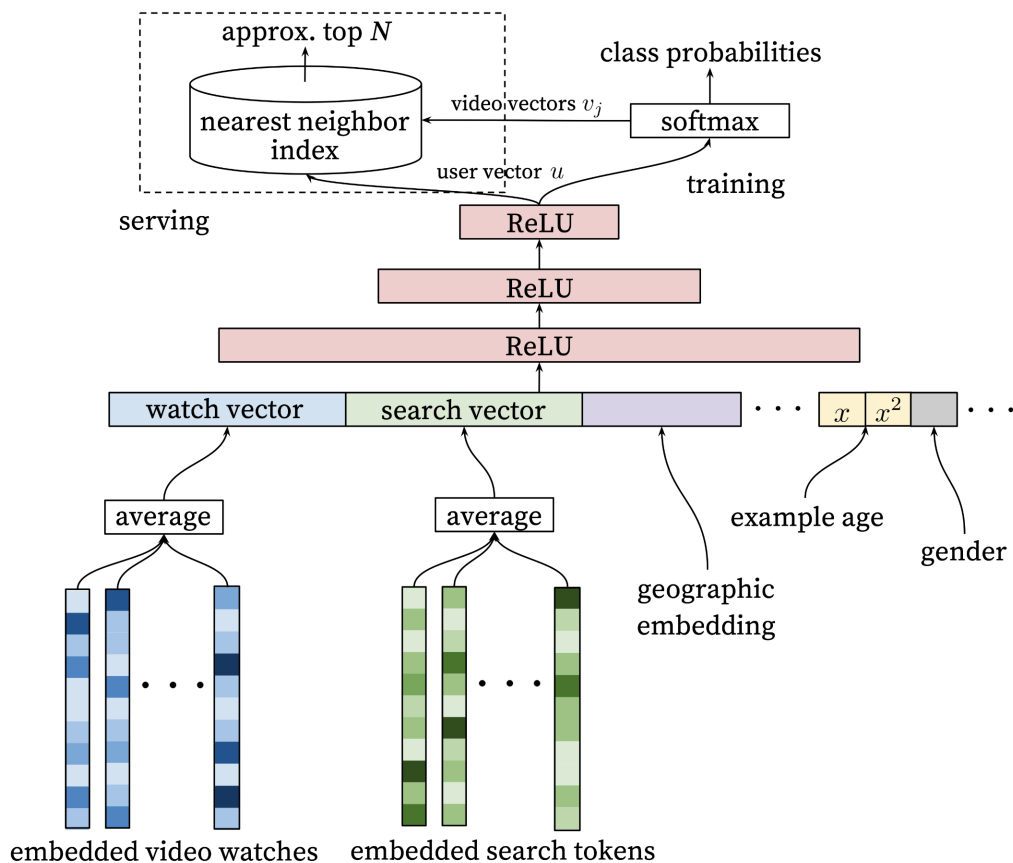


Figure 3: Deep candidate generation model architecture showing embedded sparse features concatenated with dense features. Embeddings are averaged before concatenation to transform variable sized bags of sparse IDs into fixed-width vectors suitable for input to the hidden layers. All hidden layers are fully connected. In training, a cross-entropy loss is minimized with gradient descent on the output of the sampled softmax. At serving, an approximate nearest neighbor lookup is performed to generate hundreds of candidate video recommendations.

## 6. Heterogeneous Signals

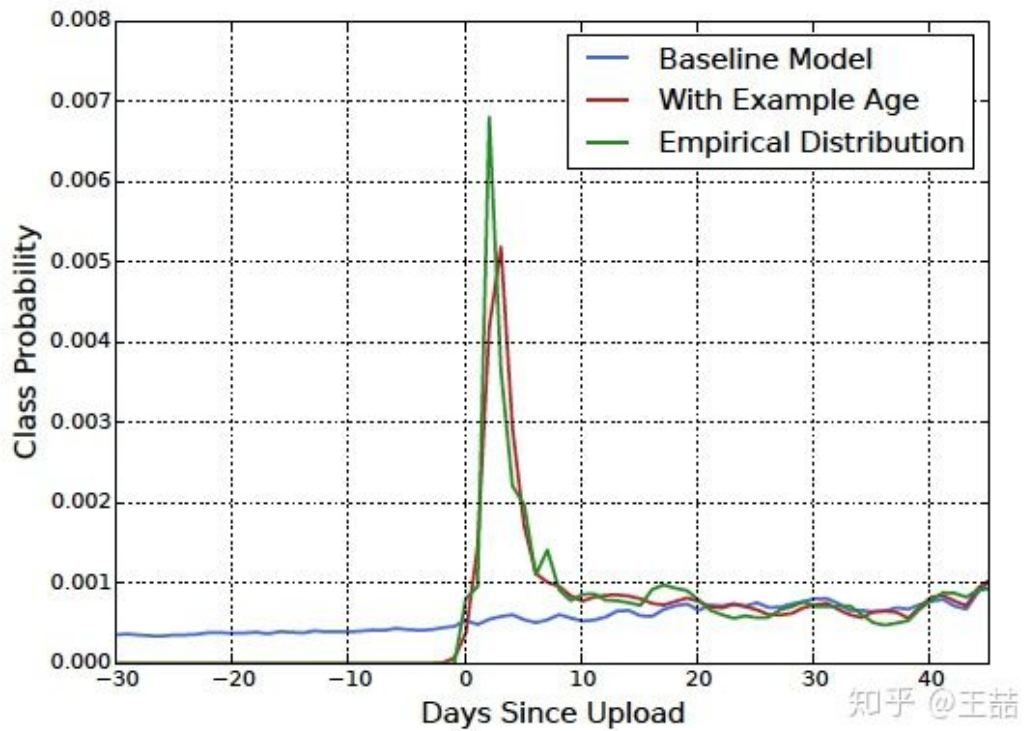
### 1. 输入三部分特征

1. User query: tokenized with **unigram or bigram**, 过embedding层后求均值, 得到query的summarized representation
2. watch videos: 过embedding层后求均值
3. demographic features (binary or continuous)
  - the user's gender and age
  - normalized to  $[0, 1]$
  - concatenate with the above embeddings
4. context infos: 设备、登录状态 (logged-in state)

### 2. 视频上传时间特征 (Example Age)

- 考虑视频上传时间对视频推荐的影响: 在相关性约束下, 越新的视频理论上更受欢迎
- 训练:  $\text{example\_age} = \text{time\_of\_training}(\text{训练时间}) - \text{time\_of\_sample\_log}$
- serving: 将example\_age置为0或者微负的值
- 意义: 例如在一段时间以前用户观看了某视频, 在训练时该age特征得取一个正值label才能预测为1; 而serving时该值为0/微负值, 所以probability偏低一些, 相当于考虑是时间对推荐倾向的影响

- 实验证明使用了example\_age特征后预测效力的分布更符合经验分布



### 3. Label and Context Selection

- 训练样本
  - 不仅考虑YouTube内推荐给用户的数据，还考虑其他嵌入页面用户观看的数据
  - 为每个用户固定训练样本数量，排除活跃用户对loss和结果的影响
- 抛弃序列信息：把watched video和query的embedding做加权
- 不对称的浏览模式：图中a方法是以历史浏览的上下文作为特征，某一次视频观看作为label；b方法是以历史浏览作为特征，下一次观看作为label。理论上看b方法更合适，因为A方法取label的方式和实际场景有gap；且b方法在A/B测表现更好。

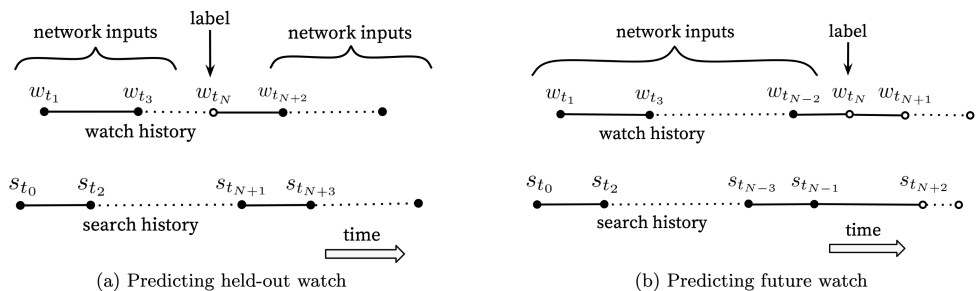


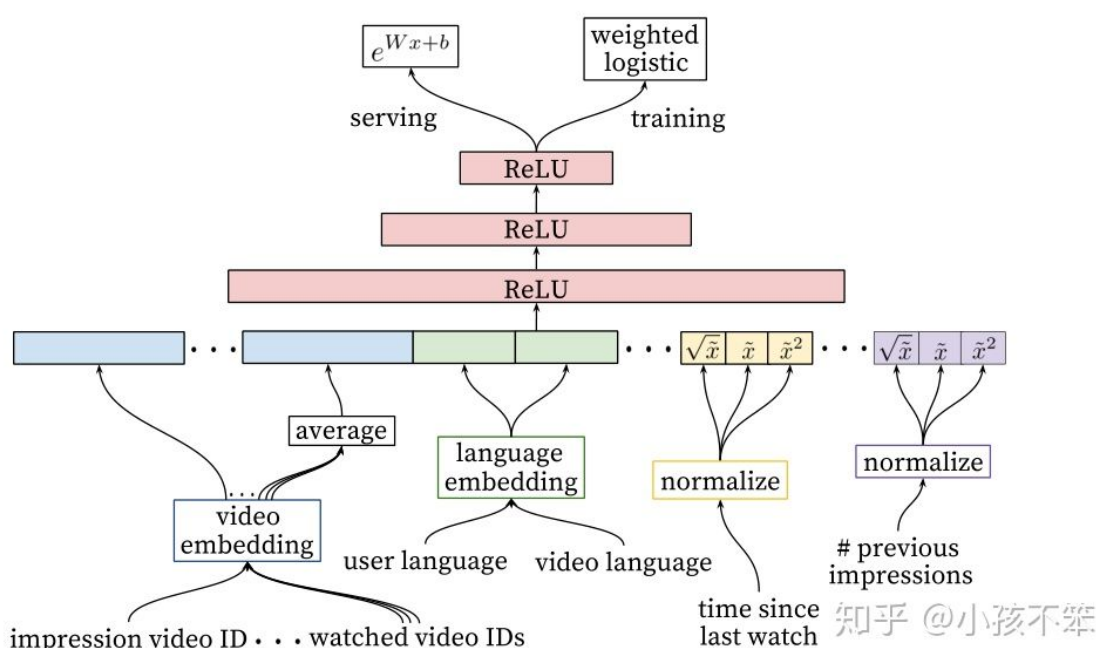
Figure 5: Choosing labels and input context to the model is challenging to evaluate offline but has a large impact on live performance. Here, solid events  $\bullet$  are input features to the network while hollow events  $\circ$  are excluded. We found predicting a future watch (5b) performed better in A/B testing. In (5b), the example age is expressed as  $t_{\max} - t_N$  where  $t_{\max}$  is the maximum observed time in the training data.

## 4. Ranking

### 1. 概述

1. 可以使用更多特征
2. emsembling different candidate sources
3. 目标函数主要基于用户观看时长(watch\_minutes\_per\_impression)，因为基于用户是否点击容易推荐“标题党”等“点击陷阱”的视频，而使用观看时长更容易捕获用

## 2. 网络结构



## 3. 特征概述

1. impression video ID embedding: 当前要计算的video的embedding
2. watched video IDs average embedding: 用户观看过的最后N个视频embedding的average pooling
3. language embedding: 用户语言的embedding和当前视频语言的embedding
4. time since last watch: 自上次观看同channel视频的时间
5. previous impressions: 该视频已经被曝光给该用户的次数。一定程度上引入了exploration的思想，避免同一个视频持续对同一用户进行无效曝光。尽量增加用户没看过的新视频的曝光可能性。

# 十问十答

1. 文中把推荐问题转换成多分类问题，在next watch的场景下，每一个备选video都会是一个分类，因此总共的分类有数百万之巨，这在使用softmax训练时无疑是低效的，这个问题Youtube是如何解决的？

- 这个问题原文的回答是这样的。简单说就是进行了负采样（negative sampling）并用importance weighting的方法对采样进行calibration。文中同样介绍了一种替代方法，hierarchical softmax，但并没有取得更好的效果。

We rely on a technique to sample negative classes from the background distribution ("candidate sampling") and then correct for this sampling via importance weighting.

2. 在candidate generation model的serving过程中，Youtube为什么不直接采用训练时的model进行预测，而是采用了一种最近邻搜索的方法？



- 这个问题的答案是一个经典的工程和学术做trade-off的结果，在model serving过程中对几百万个候选集逐一跑一遍模型的时间开销显然太大了，因此在通过candidate generation model得到user 和 video的embedding之后，通过最近邻搜索的方法的效率会高很多。我们甚至不用把任何model inference的过程搬上服务器，只需要把user embedding和video embedding存到redis或者内存中就好了。【实际使用时应该是user embedding实时生成（个人认为是因为user query可变），video embedding离线存储好】

3. 在candidate generation model的serving过程中，Youtube为什么不直接采用训练时的model进行预测，而是采用了一种最近邻搜索的方法？

- 为了拟合用户对fresh content的bias，模型引入了“Example Age”这个feature，文中其实并没有精确的定义什么是example age。按照文章的说法猜测的话，会直接把sample log距离当前的时间【指的是模型训练时的时间】作为example age。比如24小时前的日志，这个example age就是24。在做模型serving的时候，不管使用那个video，会直接把这个feature设成0。大家可以仔细想一下这个做法的细节和动机，非常有意思。【个人理解：更旧的视频在训练时age值更大，对应的，拟合目标时只需要更小的权重即可。那么在serving时age置为0了，获得的结果概率就会变得更小】
- 这个特征的用处按我理解可以这样描述：比如某个视频点击集中在7天前（比如7天前点击率10%），训练前这个时间点点击率比较低（训练前10分钟点击率3%），模型学出来之后预测的时候把Example Age置为0去预测，预测出这个视频点击率就会更接近3%。同理如果某视频以前不火，训练前突然火了，预测的时候Example Age置为0就可以预测到更高的点击率。如果不带Example Age，模型学出来的该视频点击率更接近于这个训练区间下这个视频平均点击率。

4. 在对训练集的预处理过程中，Youtube没有采用原始的用户日志，而是对每个用户提取等数量的训练样本，这是为什么？

- 每个用户采用等量样本应该是防止活跃用户带偏 model

5. Youtube为什么不采取类似RNN的Sequence model，而是完全摒弃了用户观看历史的时序特征，把用户最近的浏览历史等同看待，这不会损失有效信息吗？

- 这个原因应该是YouTube工程师的“经验之谈”，如果过多考虑时序的影响，用户的推荐结果将过多受最近观看或搜索的一个视频的影响。YouTube给出一个例子，如果用户刚搜索过“taylor swift”，你就把用户主页的推荐结果大部分变成taylor swift有关的视频，这其实是非常差的体验。为了综合考虑之前多次搜索和观看的信息，YouTube丢掉了时序信息，讲用户近期的历史纪录等同看待。
- [youtube RNN paper](#)

6. 在处理测试集的时候，Youtube为什么不采用经典的随机留一法（random holdout），而是一定要把用户最近的一次观看行为作为测试集？

- 只留最后一次观看行为做测试集主要是为了避免引入future information，产生与事实不符的数据穿越。

7. 在确定优化目标的时候，YouTube为什么不采用经典的CTR，或者播放率（Play Rate），而是采用了每次曝光预期播放时间（expected watch time per impression）作为优化目标？
- 采用观看时长应该是业务场景有关，ctr 高的观看时长不一定高，有骗点击的行为
  - 其次，观看时间也与youtube本身的收益目标一致
8. 在进行video embedding的时候，为什么要直接把大量长尾的video直接用0向量代替？
- 这又是一次工程和算法的trade-off，把大量长尾的video截断掉，主要还是为了节省online serving中宝贵的内存资源。当然从模型角度讲，低频video的embedding的准确性不佳是另一个“截断掉也不那么可惜”的理由。
  - 在实践中，直接删去低频特征可以提升效果
9. 在进行video embedding的时候，为什么要直接把大量长尾的video直接用0向量代替？
- 开方和平方是为了增加特征的非线性
10. 为什么ranking model不采用经典的logistic regression当作输出层，而是采用了weighted logistic regression？
- weighted logistic 跟第7个问题相关，因为优化目标是观看时长，weighted logistic训练，线上使用 exp 预测，近似为观看时长
  - 因为在第7问中，我们已经知道模型采用了expected watch time per impression作为优化目标，所以如果简单使用LR就无法引入正样本的watch time信息。因此采用weighted LR，将watch time作为正样本的weight，在线上serving中使用 $e(Wx+b)$ 做预测可以直接得到expected watch time的近似，完美。[揭开YouTube深度推荐系统模型Serving之谜](#)

## 自我思考

1. 为什么要设置粗排和精排？