

# YouTube基于多任务学习的视频排序推荐系统

gongyouliu 搜索与推荐Wiki 2020-12-23

点击标题下「[搜索与推荐Wiki](#)」可快速关注

## ▼ 相关推荐 ▼

1、Sentence2Vec & GloVe 算法原理、推导与实现

2、近100页的《常见的五种神经网络》汇总电子书

3、Doc2vec的算法原理、代码实现及应用启发

4、3W字长文带你轻松入门视觉Transformer

作者在《[深度学习在推荐系统中的应用](#)》这篇文章中对YouTube在2016年提出的一个两阶段的深度学习推荐模型进行了详细介绍，该模型是深度学习在推荐系统中的应用最经典的模型之一。后面（去年）YouTube又提出了一个针对推荐排序阶段的多任务深度学习模型，这即是这篇文章我们要介绍的。

在本篇文章中，我们从问题背景及算法方案简介、排序算法模型整体框架、排序算法模型核心模块、排序模型核心亮点解读、建模过程中的挑战、模型未来优化的方向等6个方面来介绍这篇非常有学习参考价值的文章。跟16年那篇文章一样，这篇文章也是难得的讲解工业级推荐系统算法工程实践的佳作。希望读者读完后可以从中受到启发，能够更好地从工程实践、问题建模、优化目标等角度来思考怎么构建推荐排序模型，从而更好地学习推荐系统算法设计的哲学。

## 一、问题背景及算法方案简介

该论文提出了一个在YouTube视频分享平台上基于当前观看视频预测用户下一部要观看的视频(具体产品形态见下面图1)的大型多目标推荐排序系统(见参考文献1)。该论文聚焦在推荐排序阶段，通过构建一个复杂的大容量模型从候选集中挑选出用户最可能看的视频作为最终推荐结果。该系统面临许多真实业务场景中的挑战，主要体现如下：

1. 面临多个不同甚至相互冲突的的排序目标；比如，我们希望用户观看某个视频，并且还希望他能够给视频高评分并分享给他的好友。
2. 用户对推荐结果隐式反馈产生的选择偏差；比如用户可能会观看一个评分比较高(因此排在推荐列表的前面)的视频但是用户本身不是特别喜欢这个视频。

为了解决如上两个现实问题，论文通过软参数共享技术(soft-parameter sharing techniques，比如 Multi-gate Mixture-of-Experts，简称MMoE，见参考文献3)来优化多个排序目标。另外，通过引入 wide & deep框架(参考文献4)，该模型增加一个整合了偏差信息(位置偏差等)的浅层塔结构来解决选择偏差问题。最终通过在YouTube上做在线AB测试验证，发现该模型确实可以显著提升多种排序目标的推荐质量。

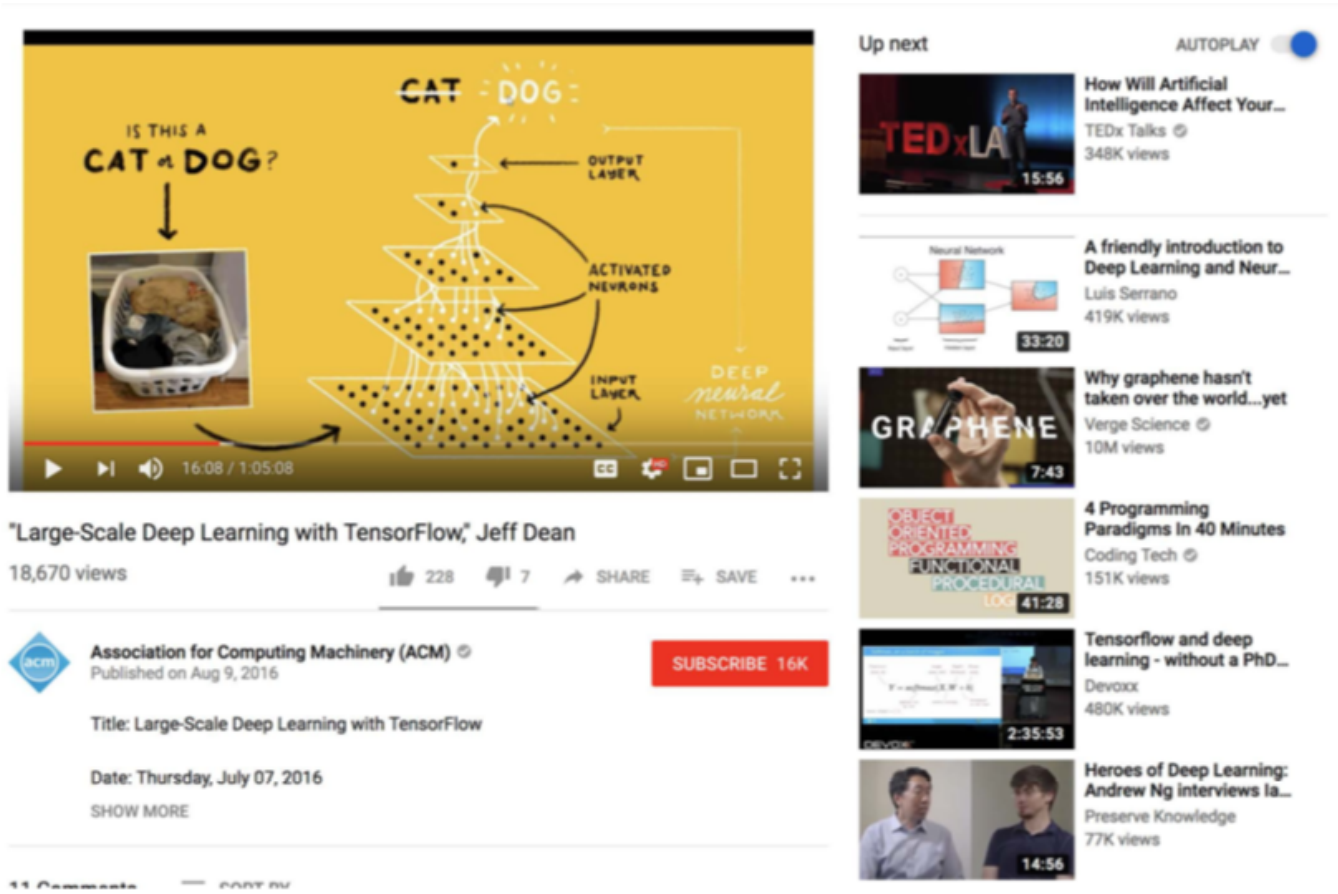


图1：YouTube推荐下一部要观看的视频(用户看完左上角视频会连播右边一系列视频，即是所谓的连播推荐)

## 二、排序算法模型整体框架

在上一节我们介绍了该论文的背景知识，在这一节，我们来介绍具体的算法模型，我们介绍该算法怎么通过引入不同的模块来解决上一节提到的两个问题的。

该算法模型的架构如下图2，该模型将最终需要优化的目标分为如下两大类：

### 1. 用户参与度(engagement objective)目标

比如用户的点击、用户对推荐视频的接触程度(视频观看比例等)等属于参与度指标。

2. 用户满意度(satisfaction objective)目标

比如用户喜欢一个视频(图1中的“顶一下”按钮)、用户给推荐的视频评分等都属于满意度目标。

为了学习和估计多样的用户行为，该模型基于wide & deep模型架构做了拓展，采用MMoE(参考文献3)技术来自动学习多个可能存在冲突的目标的共享模型参数。下图中间偏右的Mixture-of-Experts模块从多种输入数据源中学习数据不同维度的特征，每个维度可以类比为某个方面的“专家”(Experts)，再通过使用多个门控网络(Gating Networks)模块(见下图中间部分)，不同的学习目标可以选择不同的或者共享的“专家”来最终优化多目标任务。

为了减少从有偏差的训练数据中产生的模型偏差(比如位置偏差)，该模型加入了一个浅层塔(shallow tower)模块(见下图左边蓝色的shollow tower模块)，这个模块可以类比wide & deep模型中的wide部分。该模块通过整合推荐的视频的位置信息等可能产生偏差的信息，最终作为主模型的偏差项(浅塔模型输出是一个具体的数值)整合进主模型进行学习，而在推断时，可以将位置参数设置为pos=1来消除位置产生的影响。

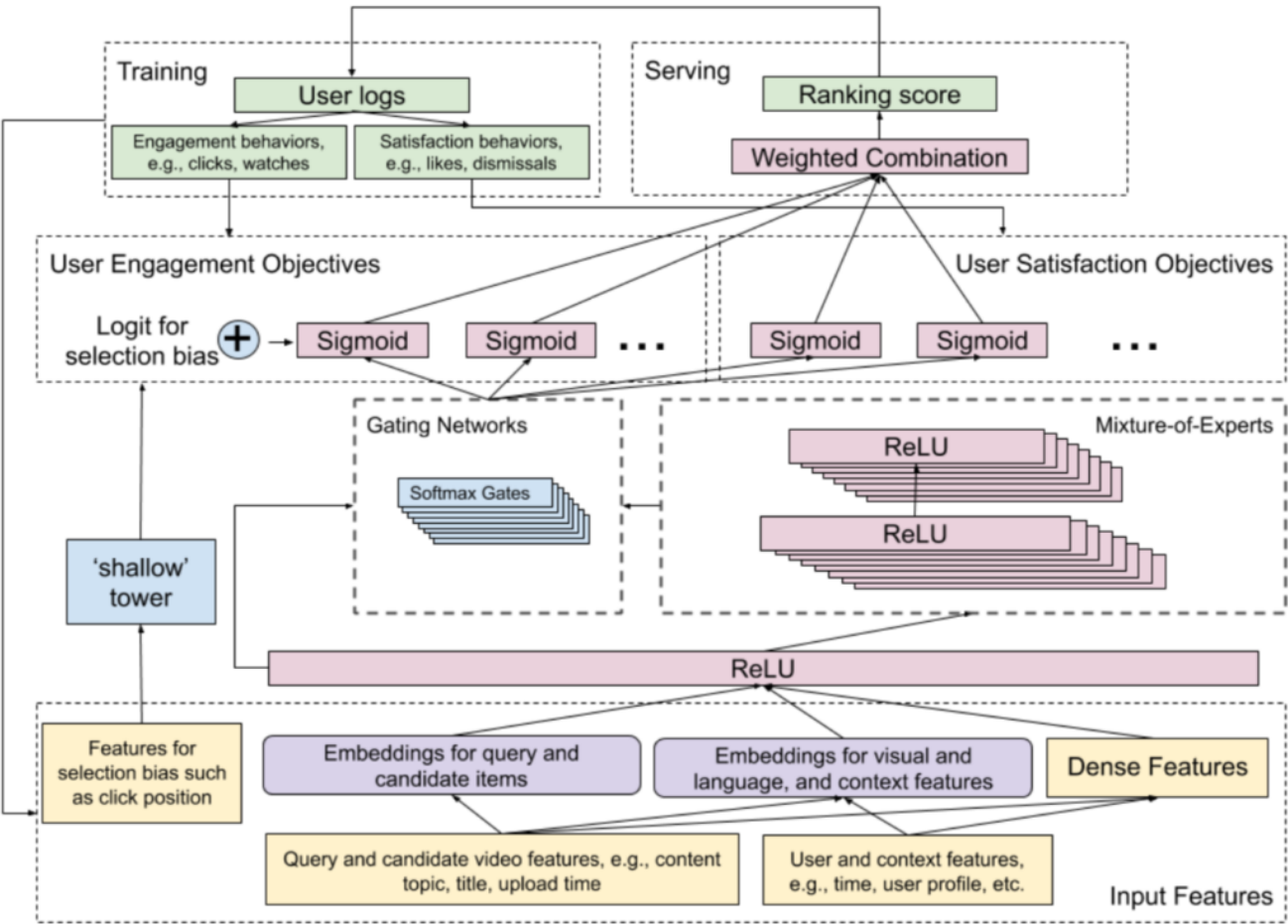


图2：多任务排序模型架构

该模型通过部署到YouTube平台中进行AB测试，相对原来的模型，在用户参与度和用户喜好度等指标上都有较大提升。

通过上面的简单介绍，总结来说，这篇论文的主要贡献有如下几点：

1. 引入了一个端到端的推荐排序模型；
2. 将排序模型抽象为一个多任务学习问题，并且利用MMoE技术来优化多目标学习；
3. 通过采用wide & deep架构，引入浅层塔结构避免位置偏差；
4. 通过在YouTube上部署AB测试，证实了该模型可以大大提升预测效果；

### 三、排序算法模型核心模块

我们在第二节对论文中算法模型的整体框架进行了简单介绍，知道了该模型的架构、解决的问题以及该论文的主要贡献，下面我们对模型的细节进行更深入的解读，让读者更好地理解该模型的精妙之处。我们从如下4个维度来讲解模型的实现细节。

#### 1. 模型系统概述

该论文中的排序系统学习两类用户反馈：用户参与行为(如点击、播放等)和用户满意度行为(喜欢、不喜欢等)。对每个待排序的候选视频，排序模型利用候选视频的特征、query(如用户播放历史或者用户人口统计学特征等)、上下文(如观看时的时间等)作为输入来预测多种用户行为(见下面图3)。

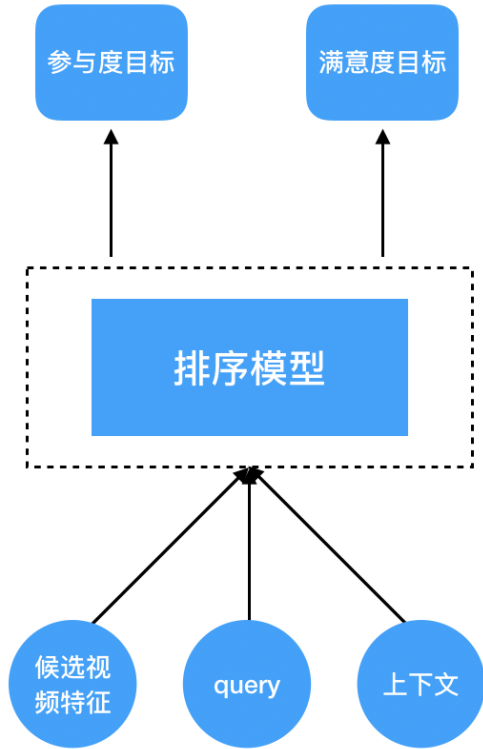


图3：排序模型的输入与目标

该论文采用LTR(learning to rank)框架，将预测问题建模为分类或者回归问题。具体是利用point-wise方法，而不是pair-wise或者list-wise LTR方法，主要是因为point-wise更简单，可以有效地拓展到大规模的候选集排序中，尽管pair-wise和list-wise方法可以增加预测结果的多样性。

## 2. 排序目标定义

排序模型利用用户的行为作为训练的label。在1中提到排序模型预测参与性目标与满意性目标。对于参与度目标(如点击与播放)，将预测行为建模为二分类任务(如点击) 和回归任务(如用户播放时长)两大类。类似地，对于满意度目标，也可以建模为二分类任务和回归任务两大类，比如，喜欢某个视频可以看成是二分类任务，对视频进行评分可以看成是回归任务。对于二分类任务，可以采用交叉熵(cross entropy)损失函数，而对于回归任务，可以采用平方损失函数。

一旦多任务排序的目标和建模问题类型确定了，我们可以训练多任务排序模型来进行最终的排序。对任意一个候选视频，我们通过多个目标预测模型的输入获得一个综合得分(通过多个预测模型得分的加权乘积，权重通过手动调整，需要权衡用户参与度和满意度目标)。

## 3. 利用MMoE技术来对任务的相关性与冲突性建模

多目标排序系统的学习一般使用一个共用的底层模型(见下面图4左边的网络结构)，但是硬编码的参数共享往往对学习多目标是有副作用的，特别是当这些不同学习任务相关性很低时。为了避免这种情况，

这篇论文采用了MMoE技术，MMoE采用软参数共享结构，可以消除多任务目标中的存在冲突的目标。

MMoE采用Mixture-of-Experts(MoE)结构应对多任务学习，让多个任务共享这些“专家”，同时，每个任务配备一个训练好的门控网络(gating network)来协调不同“专家”所起的作用。相比于共享的底层模型结构，MMoE需要更少的参数。MMoE的核心思想是对每个学习任务将共享的ReLU层替换为MoE层并加上门控网络。

推荐排序系统通过在共享的隐含层之上增加“专家”层(见下面图4的右边网络结构)，让最终的模型可以从输入中学习模块化的信息。在输入层或者隐含层之上增加“专家”层，最终可以更好地对多模态的特征空间进行建模。但是直接加在输入层之上会增加模型训练和推断的时间开销(因为输入层维度相比隐含层包含更多的参数，参数越多，训练效率往往越低)，因此该论文采用“专家”层直接加在隐含层之上的方法。

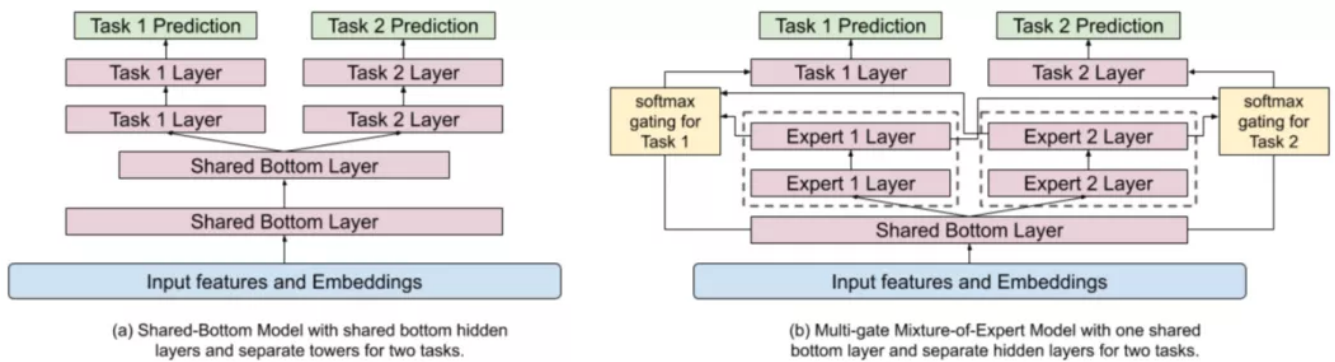


图4：利用MMoE来取代共享的网络层

该模型的“专家”网络相当于参考文献3中的利用ReLU作为激活函数的多层感知机。对于任务  $k$  和预测值  $y_k$  及最后一个隐含层  $h^k$ 。包含  $n$  个“专家”的MMoE层在任务  $k$  上的输出为：  $f^k(x)$ 。可以用如下方程来表示：

$$y_k = h^k(f^k(x)),$$

$$\text{where } f^k(x) = \sum_{i=1}^n g_{(i)}^k(x) f_i(x)$$

这里  $\mathbf{x} \in \mathbb{R}^d$  是共享隐含层的低维嵌入向量， $g^k$  是任务  $k$  的门控网络， $g^k(x) \in \mathbb{R}^n$ ， $g_{(i)}^k(x)$  是第  $i$  个分量，而  $f_i(x)$  是第  $i$  个“专家”。门控网络是输入通过简单线性变换后再经过softmax变换，即



$$g^k(x) = \text{softmax}(W_{g^k}x),$$

这里， $W_{g^k}$ 是线性变换的自由参数。“专家”的数量可以设置为很多个，但是论文构建的模型中只用少量的“专家”，期望“专家”可以被更多的任务共享，这样做主要是期望提升模型的训练效率(“专家”数越多参数越大)。

#### 4. 通过整合浅层塔结构剔除位置等带来的偏差

隐式反馈数据被广泛用于排序学习模型的训练中。从用户行为日志中可以获得大量的隐式反馈信息，因此可以利用复杂的神经网络模型来学习排序特征。但是，很多隐式反馈是从推荐排序系统的产品交互中产生的(我们通过给用户推荐视频，用户点击推荐的视频产生隐式反馈行为)，因此是带有偏差的。位置偏差(推荐的视频在推荐列表中所排的次序)及其他类型的偏差被大量研究者在排序学习模型中所证实。

在我们的排序系统中，这里的查询请求是当前用户正在观看的视频(如果将推荐问题看成一个搜索问题，那么当前在看的视频可以等价于搜索的查询关键词)，候选视频是相关的视频，用户通常倾向于点击并观看显示在列表顶部的视频，而不管这些视频的真实用户效用(与观看视频的相关性、用户的偏好)如何(这是一种先入为主的效应，因此会产生位置偏差)。我们的目标是从排序模型中消除这种位置偏差。在我们的训练数据中或在模型训练过程中通过建模减少选择偏差可以提升模型质量。

我们的模型采用跟wide & deep类似的结构，将模型预测分解为两个部分：来自主塔的用户效用(user-utility)部分和来自浅层塔的偏差部分。具体地说，我们训练一个包含选择偏差(如位置偏差)特征的浅塔模型，然后将其输出添加到主模型的Logit中，如下图5所示。在训练中，使用所有视频的位置信息，采用drop-out的思路，随机丢弃10%的位置信息，以防止我们的模型过度依赖位置特征。在模型推断期间，将位置特征设置为缺失值(这样所以视频的位置信息都不考虑了，从而剔除了位置偏差)。同时我们将位置特征与设备特征交叉(是因为在不同类型的设备上观察到不同的位置偏差)。

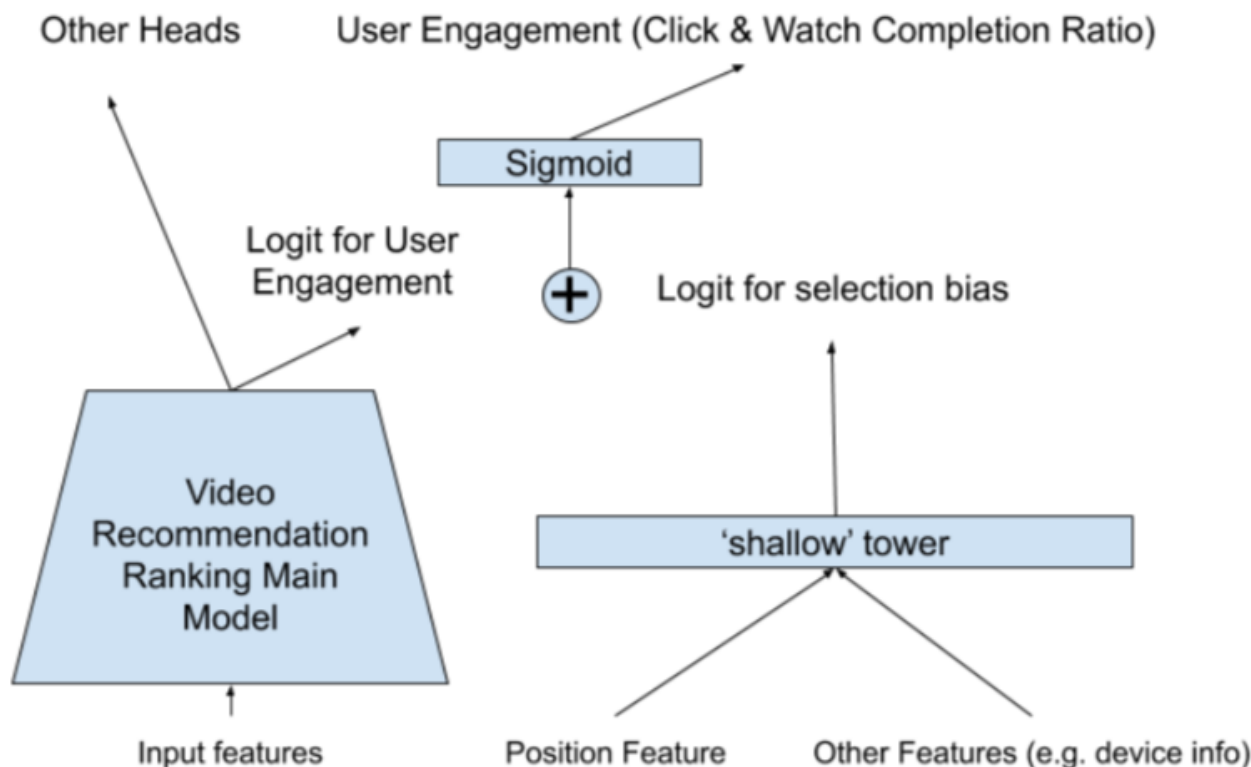


图5：增加浅塔层便于学习模型的偏差(如位置偏差)

## 四、排序模型核心亮点解读

通过上面的详细讲解，我们介绍完了这篇非常有工程参考价值的推荐排序学习模型。在这一节我们来梳理一下该模型的核心亮点，这些点都是值得我们参考借鉴的。

### 1. 通过多任务学习来进行排序

该模型通过将用户的行为分为参与度行为与满意度行为两大类，作为独立的任务，通过机器学习中的多任务学习框架来建模。参与度行为更多的是隐式反馈，而满意度行为更多的用户的主动反馈，在同一个模型中整合这两类反馈可以充分利用已知信息，提升模型的精准度。使用多种类型的隐式反馈信号和多个排序目标，我们获得了更多调控手段，更好地平衡模型预测结果和用户真实兴趣之间的关系。

### 2. 通过MMoE技术来平衡多目标

有了多任务框架，那么怎么来学习和平衡这两类任务呢？MMoE框架通过在不同任务之间共享“专家”的方式来更好地平衡多任务之间的冲突和相关关系，可以更好地学习多任务模型。这个思想是非常朴素的，不同的“专家”有不同的优点和特长，不同任务通过在不同“专家”上学习到不同的权重参数分布(可以



简单理解为选择适合自己的“专家”), 最终达到更好地刻画每个任务的目的, 最终提升整体模型的效果。

分布式训练一般可能会导致模型发散。在MMoE模型中, 采用softmax激活函数的门控网络会导致“专家”分布不平衡的问题, 最终导致很多“专家”的效用为0, 该论文作者在实验中发现大概20%的概率会出现这种情况, 作者采用drop-out的方式来解决了该问题, 具体就是采用10%的概率随机设置“专家”的效用为0, 并且重新归一化softmax的输出。

### 3. 通过整合位置信息来消除偏差

采用wide & deep的双塔结构, 在浅层塔中通过整合位置信息, 让模型学到位置偏差, 最终在模型推断时剔除掉位置偏差, 从而让模型有更好的表现。位置偏差等选择偏差是推荐算法工程师在构建推荐模型中很少会考虑的(但是一般是会意识到这个问题的)的一类影响因子。该模型提供了一种消除位置偏差的解决方案和思路。

### 4. 按照时间序将数据灌入模型训练

作者按照样本产生的时间顺序依次训练模型。先训练之前的数据, 当有新的样本产生时再将新样本加入训练中。通过这样做, 模型可以适应最新的数据。这对于许多实际的推荐应用来说是非常关键的, 在这些实际应用中, 数据分布和用户模式会随着时间的推移而动态变化。

## 五、模型构建过程中面临的挑战

在构建YouTube排序推荐模型过程中会面临很多问题与挑战, 这里我们简要整理一下, 本篇论文对这些问题与挑战给出了相应的解决方案, 在前面都有所讲解。这些问题及经验教训值得我们多思考。

### 1. 构建基于神经网络的推荐排序算法面临的问题

许多推荐系统研究扩展了最初为传统机器学习应用设计的模型体系结构, 如自然语言处理的multi-headed attention和计算机视觉的cnn算法。然而, 许多适合于特定领域的表示学习的模型体系结构并不直接适用于YouTube的推荐需求。这是由于:

#### (1) 多模态特征空间

YouTube排序模型依赖多种数据源的特征, 这些特征包括类别特征、文本、图像等等, 从混合特征中进行学习是非常有挑战的。

## (2) 可拓展性及多目标排序

很多模型结构被设计来建模某种信息，它们在提升某个目标的同时可能损害另外一个目标，通过构建复杂的模型结构来处理多目标又往往无法拓展到大规模数据场景。

## (3) 包含噪音及局部稀疏的数据

排序系统需要训练视频和查询的嵌入向量。然而，大多数稀疏特性遵循幂律分布，并且在用户反馈上有较大的差异。比如，在同一条件下，用户可能会播放或者不播放一个视频仅仅由于推荐视频的视频海报图清晰度上的一点点差异，而这些差异又很难被算法捕捉到，因此，对于长尾内容训练出好的嵌入表示是非常困难的。

## (4) 利用mini-batch随机梯度下降法来分布式训练

当数据量大，模型表达能力强(模型复杂，参数多)时，必须要用分布式技术进行训练，这本身在工程上就是一件非常有挑战的事情(需要搭建分布式计算平台来支持模型训练，还要实现分布式算法)。论文作者基于TensorFlow利用TPU来训练模型，利用TFX(见参考文献7)来提供生产服务。

### 1. 效果与效率的平衡

在工业级推荐系统中，效率除了影响请求时间外，也会影响用户体验。过于复杂的模型会显著增加生成视频推荐结果的延迟，从而降低用户满意度和在线指标。因此，我们通常更热衷于使用更简单、更直接的模型。

## 2. 训练数据中的偏差

除了位置偏差外，模型可能还存在其他种类的偏差，这些偏差可能是未知的。自动化地学习和捕获这些偏差本身就是一个非常复杂的问题。

## 3. 模型评估面临的挑战

由于排序模型使用隐式反馈，线下评估好的模型上线后不一定真实的线上指标就好，甚至有时真实线上指标效果会更差。一般来说，可以选择一个简单的模型，这样可以更好地推广到线上，产生相对较好的结果，也就是说，简单的模型能更好地保证线下和线上指标的一致性。

# 六、未来优化的方向

论文作者在文章最后总结了几个对该模型进行优化的方向，具体包含如下3点：

## 1. 探索新模型架构，在稳定性、可训练性、强表达能力等三方面达到平衡

MMoE技术通过灵活选择哪些“专家”可以共享来提升多任务排序模型的性能，参考文献5提供了另一种架构，除了可以保证模型性能外，还能提升模型稳定性，是值得研究的一个方向。

## 2. 理解和识别偏差

为了对已知和未知的偏差进行建模，探索从训练数据中自动识别潜在偏差并学习如何减少这些偏差的模型架构和目标函数是一个值得研究的方向。

## 3. 模型压缩

为了降低请求延迟，可以使用不同类型的模型压缩技术(见参考文献6)，减少模型大小，从而提升处理效率。

# 总结

该论文通过构建多任务学习模型，利用MMoE技术来平衡多个任务之间的相关或者冲突关系从而更好地平衡多个目标。同时将位置偏差信息整合进模型，消除偏差影响，最终达到非常好的线上效果。

从这篇文章出发，我们可以进行更深入的思考。比如视频一般是包含贴片广告的，增加贴片广告(从而为公司创造广告收益)和用户体验之间也是一对冲突的目标，是否可以采用多任务学习的方式来寻求广告收益和用户体验的平衡呢？除了位置偏差外，视频海报的颜色、清晰度是否会产生偏差呢？怎么通过将这些因素引入模型解决相关的偏差呢？这些都是值得探索的问题，相信本文给我提供了很好的思路，为我们指明了探索的方向。

# 参考文献

1. [2019 YouTube] Recommending What Video to Watch Next- A Multitask Ranking System
2. [2016 YouTube] Deep Neural Networks for YouTube Recommendations
3. [2018] Modeling task relationships in multi-task learning with multi-gate mixture-of- experts
4. [2016 Google] Wide & Deep Learning for Recommender Systems
5. [2019] SNR: Sub-Network Routing for Flexible Parameter Sharing in Multi-task Learning
6. [2018] Ranking distillation: Learning compact ranking models with high performance for recommender system