全民K歌直播推荐算法实践

导读:随着电子商务规模的不断扩大，商品个数和种类快速增长，顾客需要花费大量的时间才能找到自己想买的商品。这种浏览大量无关的信息和产品过程无疑会使淹没在信息过载问题中的消费者不断流失。

为了解决这些问题，个性化推荐系统应运而生。个性化推荐系统是建立在海量数据挖掘基础上的一种高级商务智能平台，以帮助电子商务平台为其顾客购物提供完全个性化的决策支持和信息服务。今天会和大家分享下腾讯旗下的全民K歌的推荐系统技术及应用。

今天的介绍会围绕下面四点展开：

* 业务背景及团队介绍
* 推荐系统架构及挑战
* 召回模型算法设计
* 精排模型算法设计

**▌业务背景及团队介绍**

**1.团队介绍**

**腾讯音乐娱乐集团(TME)于2018年从腾讯拆分独立上市,涵盖四大移动音乐产品,全民K歌是其中之一,当前月活用户量超过1.5亿.现算法团队负责全民K歌各场景的推荐功能算法开发与工程实现,现在团队20+人,分布在深圳.北京两地.**

1. **业务背景**

**全民K歌的推荐场景非常多元化,涵盖内容,直播,歌房,点歌等多种形态.在K歌APP中有优质UGC推荐,关注流推荐,同城社交推荐,直播推荐,歌房推荐,点歌推荐.本团队主要负责的是其中的点播推荐.下面简单介绍一下点播推荐.**

**① 点播指标**

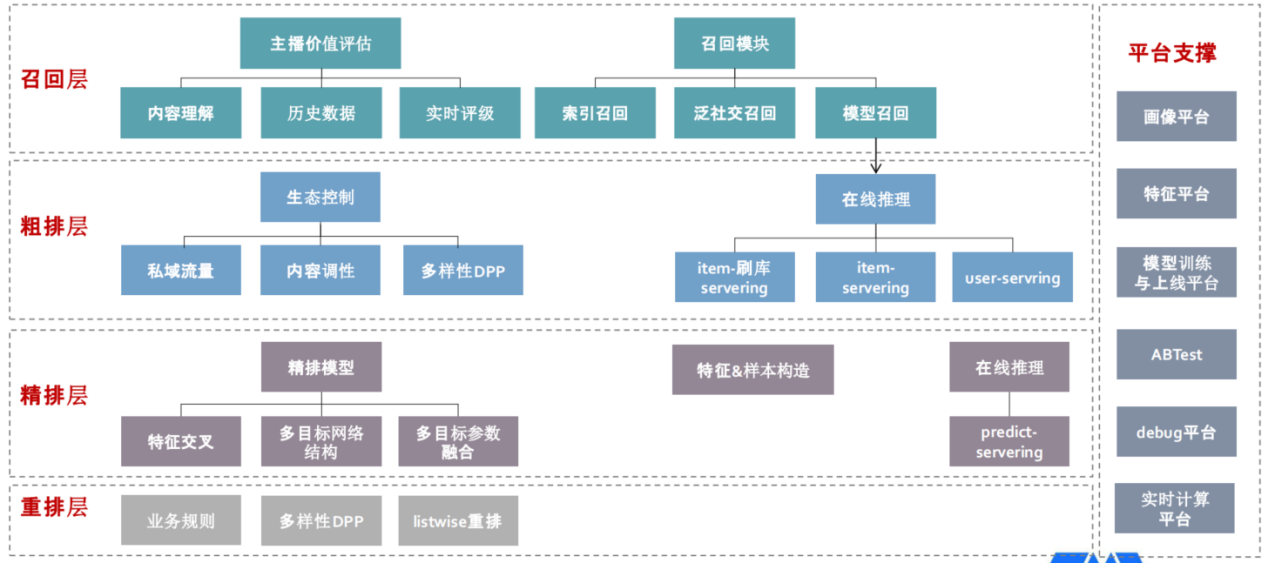
**一般来说点播指标分为四类:社区指标,增长指标,运营指标和商业指标. 关注属于社区指标;增长指标可以分为老用户场景和新用户场景.老用户场景包括:直播TAB点击率和时长.新用户场景包括:平台伴奏偏好用户,推荐FEED点击率,UGC广播点击率和社交偏好用户.运营指标主要是流量控制,包括:完成度,平稳度,细粒度.而送礼和送花属于商业指标.**

**② 点播推荐和其他推荐的比较**



**点播推荐和内容推荐的对比:它们的相同点是都有一个严格的计划经济,在一个严格的流量池下寻找用户和主播,用户和item的相关性;不同点在于直播推荐寻找的是最优的转化率模型,而内容推荐更多的是寻找多目标的融合.直播推荐不仅需要对主播进行比较严格的保量,而且需要让用户平滑地进入直播间.K歌的直播推荐对比虎牙或者YY:场景更大而全,用户的量级非常大,用户行为和画像更丰富.而虎牙或者YY它们的类目是更丰富的.**

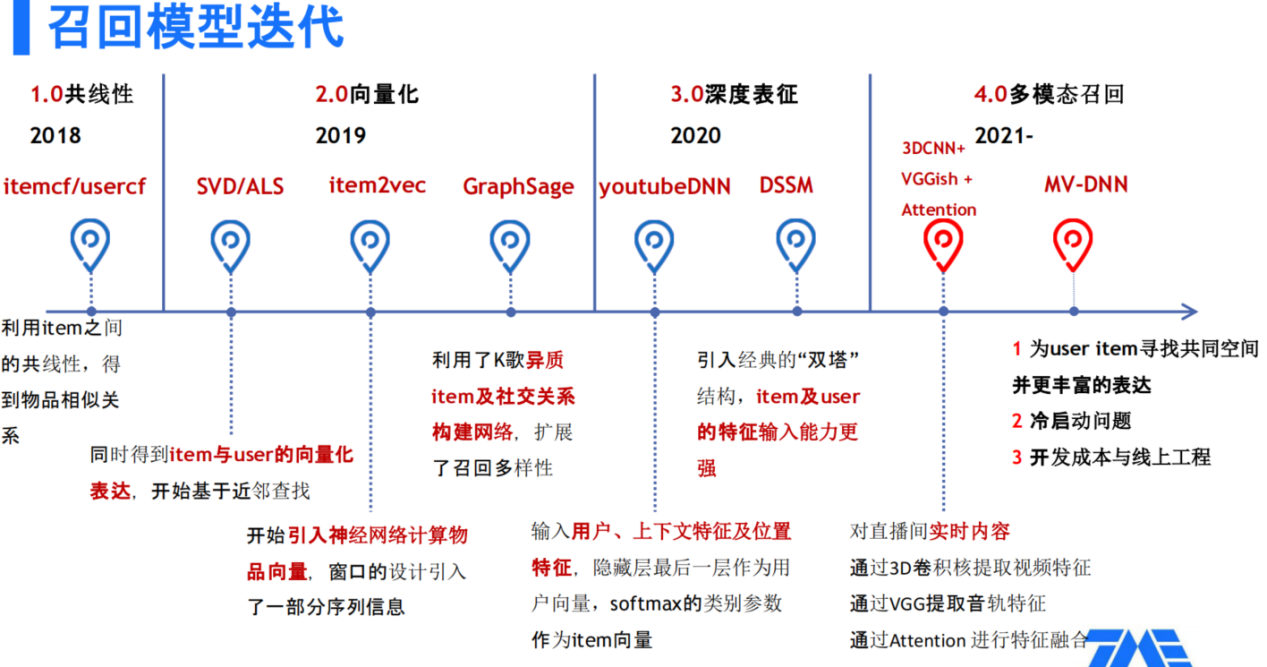
**▌推荐系统架构及挑战**



在召回这一层面,有索引召回,包括比如用户喜欢的伴奏来召回直播间.泛社交召回,还有通过好友和用户感兴趣的人所在的直播间来召回.另外,算法团队对主播有非常严格的价值评估体系,包括实时直播推流去进行视频理解和音频理解,对主播的历史和实时数据的理解,还包括一些活跃性和社交向以及衍生向的指标.在粗排层,除了需要去考虑一些模型化的问题外,还要考虑一些常亮常打赏的关系链这些数据流量,以及用户去消费直播间内容所能感受到的多样性.对于精排层,主要考虑两个方面,第一个方面是特征,针对特征怎么去做更好地交叉;另一个方面是多目标,团队考虑怎么去设计更好的多目标网络结构和更好的多目标的参数融合.在重拍层,除了对业务的研究之外,还加入了行列式这样一些模型来进行多样性重排.在除了召回,粗排,精排,重排之外,团队还依托了多个平台支撑,包括:画像,特征,训练,AB test,debug和实时计算平台.

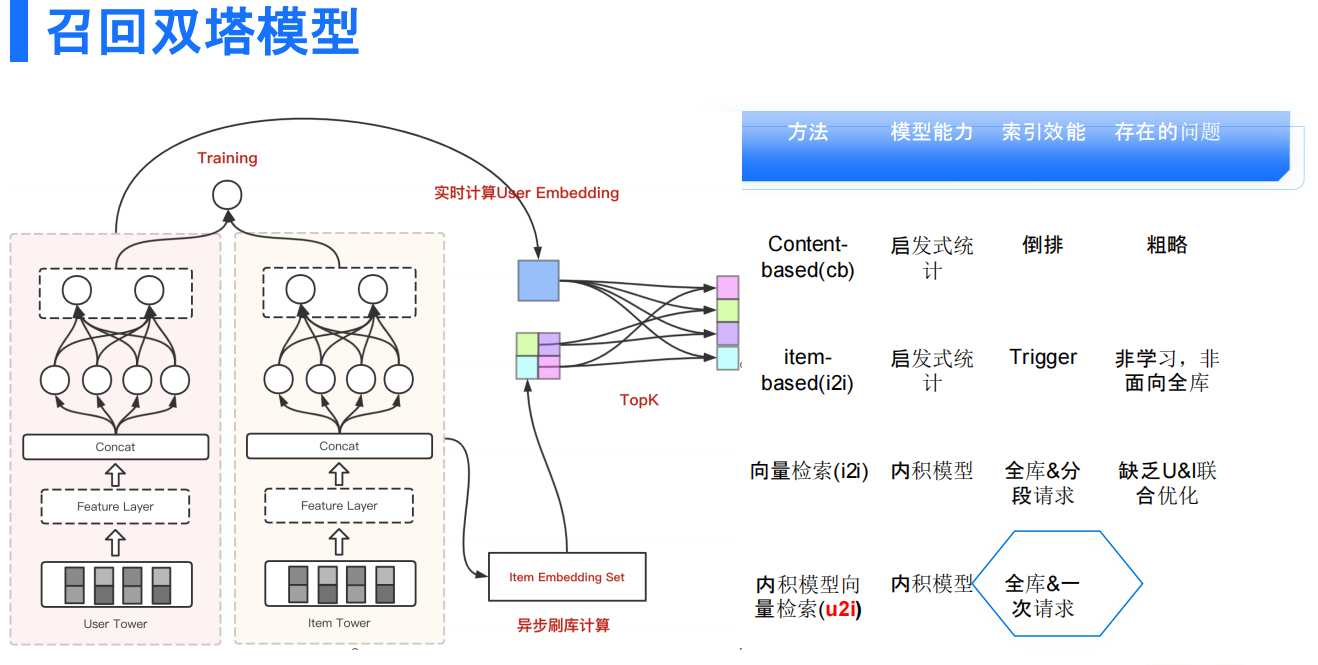
****▌**召回模型算法设计**

1. **召回模型迭代**



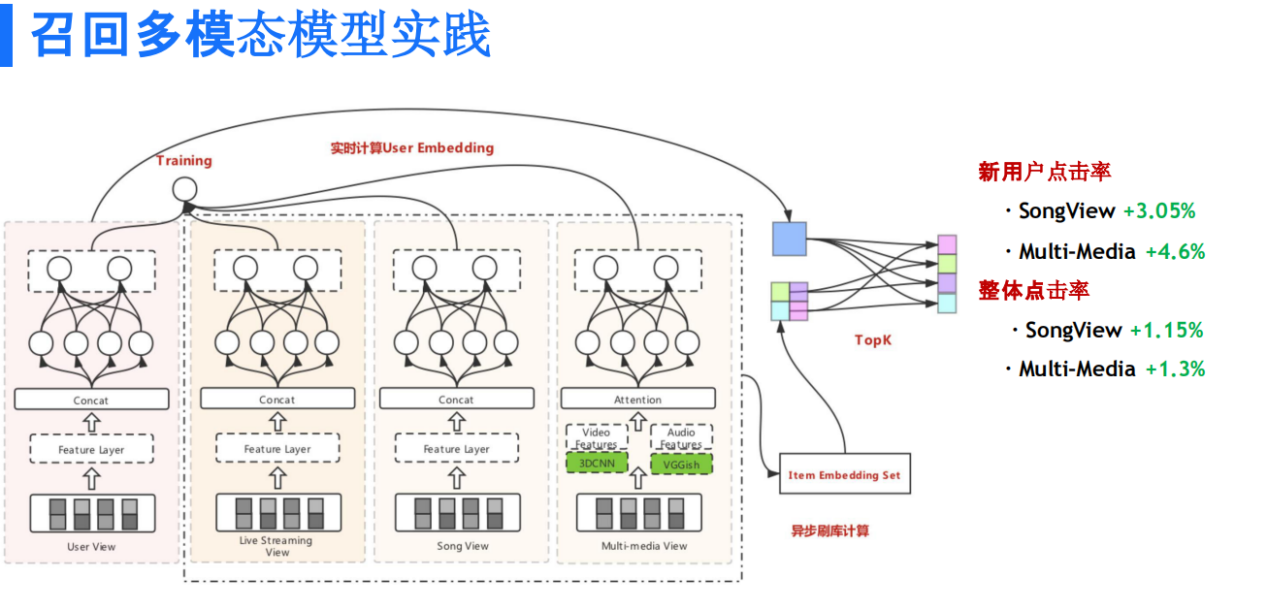
从2008年开始做推荐系统,跟随着深度模型的进展,我们的模型迭代了四个阶段,现在,工业界的baseline就是双塔模型(DSSM).总的方向就是为用户和item寻找一个共同的空间,并且能够有更丰富的表达.例如DSSM是使用了user和item的特征并且让它们能够寻找空间进行表达.在冷启动的处理上,类似DSSM的这种深度模型能更多更及时地覆盖item.在这样的baseline的基础上,我们认为DSSM其实是一个single view,还可以有更多的view来辅助主场景,比如我们加入直播间的伴奏来进行辅助训练.我们还加入一些对直播间内容的实时理解,比如我们通过对实时的直播流的3D卷积核提取视频特征,也包括提取音轨特征去得到视频的一个view来辅助我们进行训练.

1. **召回双塔模型**



那么我们重点讲讲DSSM.为什么DSSM在工业界是一个非常baseline的一个方法呢,以及为什么在这样的框架下能进行非常好的延展呢?从两个方面去进行讨论:第一个方面就是这个模型的表达能力,它的表达能力的上限是非常高的,因为它比起一些传统的方法比如说:content-based或者i2i这样的一些形式,它是有内积模型在的,它比起静态模型少了一些交叉特征,那么在这种情况下,它比起一些启发式的统计,它的上限是比较高的.我们后面还可以加入一些例如模型融合等其他方法来提升这个模型的上限.在线上的工程来看,我们只用通过一次请求,并且这样的模型能够覆盖更多的user和item,所以在模型的能力和索引的效能都能比较好的基础上,我们可以应用这样的框架去部署召回模型-双塔模型.

1. **召回多模态模型实践**



那基于这样的召回模型,我们做了怎样的优化呢?我们增加了一个SongView和多媒体的View.我们的思考点是K歌大部分的主播其实是属于唱歌类型的主播,那么大部分的直播间可能都带着SongID这样一个特征信息,那么这样一个信息是不是能辅助我们对用户进行推荐呢?尤其是K歌直播的许多新用户其实很早以前就是个K歌的老用户,他/她对这个Song其实已经有了很多的丰富行为,加了这个信息来作为一个辅助View是不是能够对新用户提升一些效率呢?而且除此之外,我们的兄弟部门QQ音乐,能对这些Song有一个比较好的训练好的embedding来为我们提供支持,加上我们的多媒体库也有很多根据伴奏的一些信息,考虑到以上诸多的优点,我们就尝试了这样一个Multi-view的 DNN,这样的效果在新用户以及整体的点击率上都能够取得一些成绩.那么进一步思考,其实直播,比如说在我们的一个打卡片的场景中,用户能够直接看到我们的打卡片,我们可以通过内容进行理解,直播的本质是不是说我们能够通过理解它的内容来给用户一个更好的推荐,尤其是对一些刚来K歌的用户的第一印象而言,比如对于一个男生来,他会看到这个女生很漂亮,这样一种情况,那么多媒体会给我们一些比较大的帮助.所以我们增加了多媒体这个辅助view能对我们的新用户和点击率有一定的帮助.

**▌精排模型算法设计**

1. **精排模型多维度迭代**

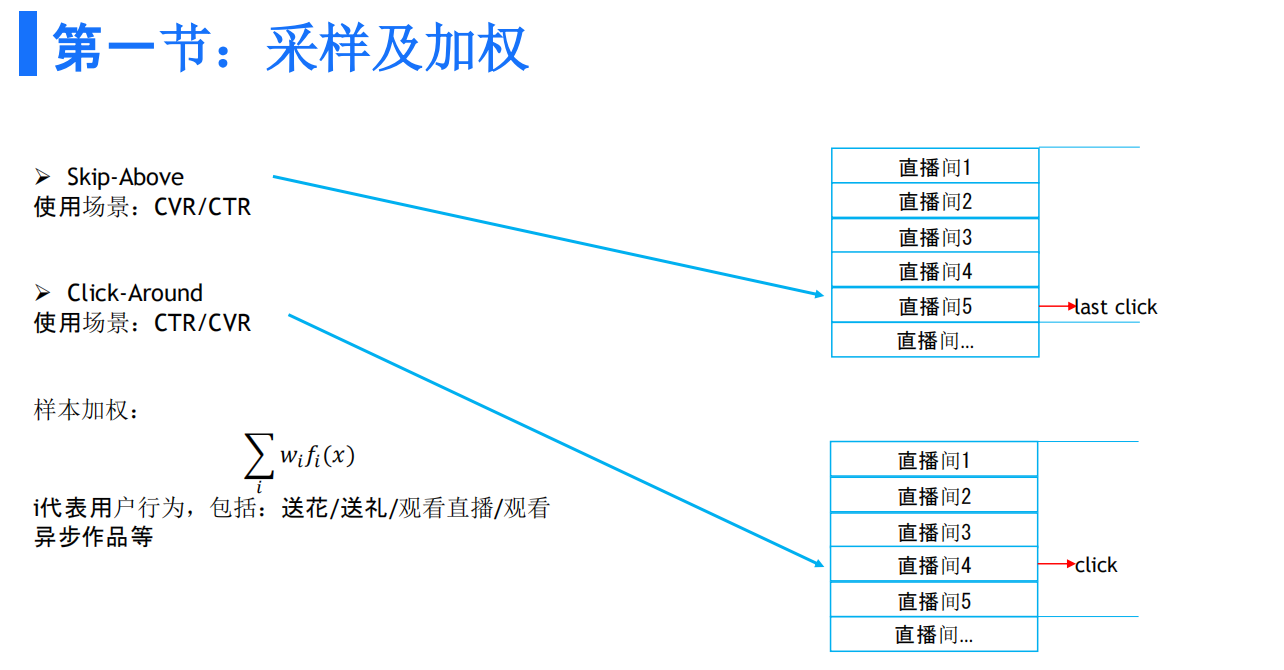
那么下一部分就是精排模型,我们的精排模型主要在三个维度进行了优化.第一个维度是特征交互的部分,在这部分我们的主要侧重点是如何进行显式的特征交叉,显式的特征交叉包括了两部分的优化,一部分是FM这样的低阶特征交叉,一部分是像Crosses和Autoint 这样的高阶特征去进行交叉,那么我们的总思想是能不能让特征的交叉更加的自由而且更加的具有物理意义.后面会详细地再介绍一下我们是如何去优化的.

那么第二个维度是多目标,我们从最早的share-bottom来看的话,逐步地用mmoe,或者是cgc,在这样的情况下,我们其实在寻找的是一个如何更加精致,更加极限的能够创造一个条件让多个任务能够相互受益,能够最大限度地去共享,学习到彼此的收益.另外一个方向是当多个任务出现一些矛盾点,或者说我们需要去权衡各个任务之间的权重时,那么在训练的过程中,我们如何去选择或者迭代,包括我们一开始去尝试的gradnorm和后来去尝试的pcgrad.

1. 特征处理,采样及加权

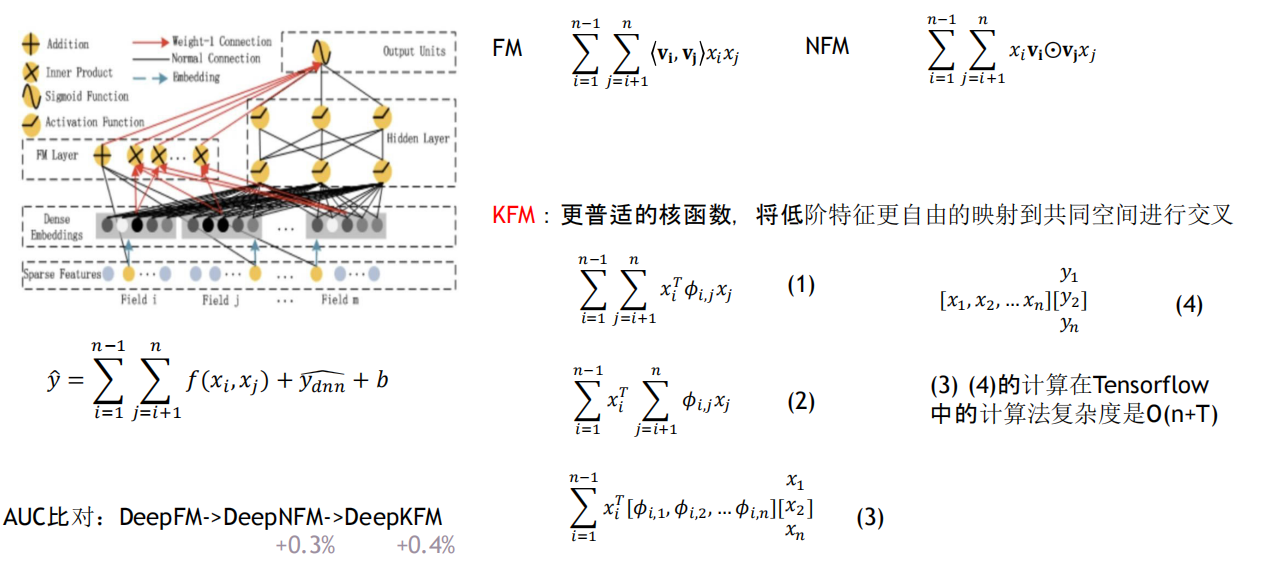


**我们首先介绍一下特征的处理.特征首先包括数值特征,类别特征,ID特征,序列特征还有embedding.举个例子,序列特征:比如说用户去过的直播间,我们会把这些直播间ID转成embedding.对这些embedding做后续的一些例如pooling或者attention这样的操作进而去训练我们的模型.多媒体团队为我们提供多媒体的embedding,第一种用法就是我们把embedding当做一个数值来喂给我们的模型;第二个用法就是我们把embedding作为一个主播的表达直接加载到我们的模型中进行进一步的运算.**

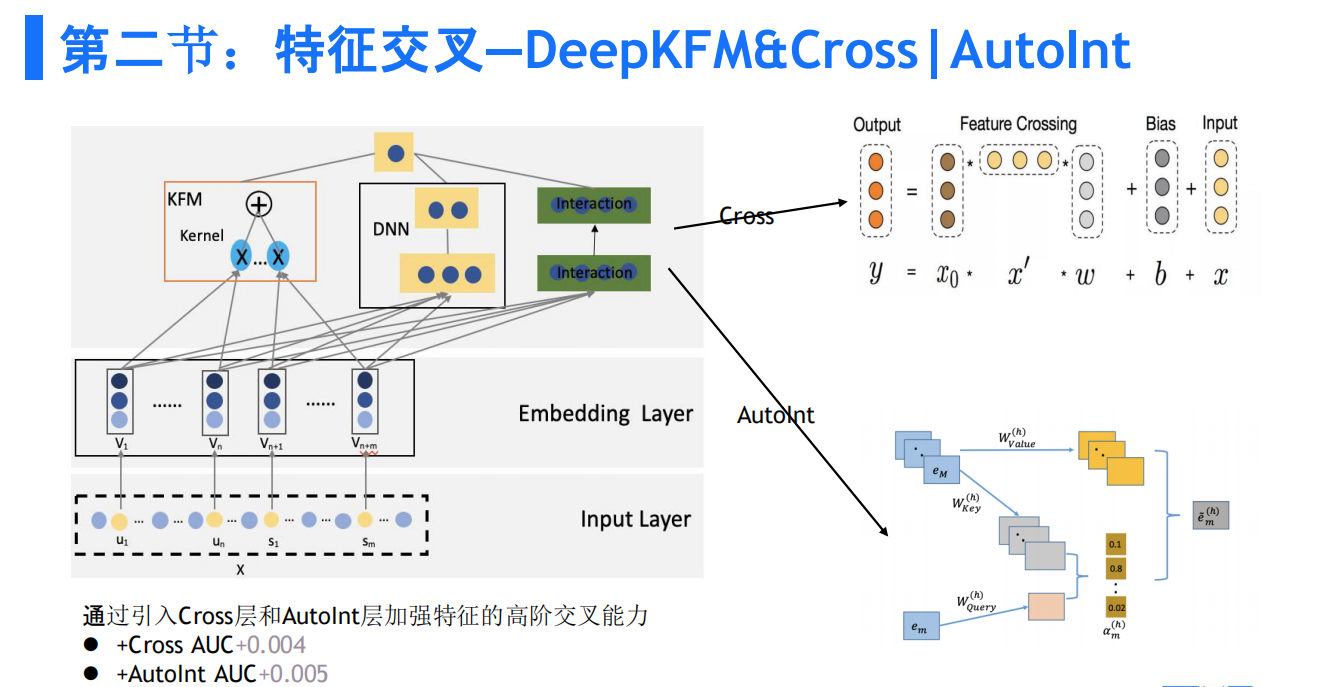


**对于双排方法,由于我们是双排的直播间click,所以我们需要有一定的采样方式,这样的采样方式一般是有两种:一种是Skip-Above,另一种是Click-Around,去采取一些有效的负样本来使我们的模型更加鲁棒,对于样本的加权,在我们的场景中如果有一些送花送礼或者是异步作品的话,会对这样的样本模型加权.**

1. **特征交叉-KFM,DeepKFM,Cross|AutoInt**

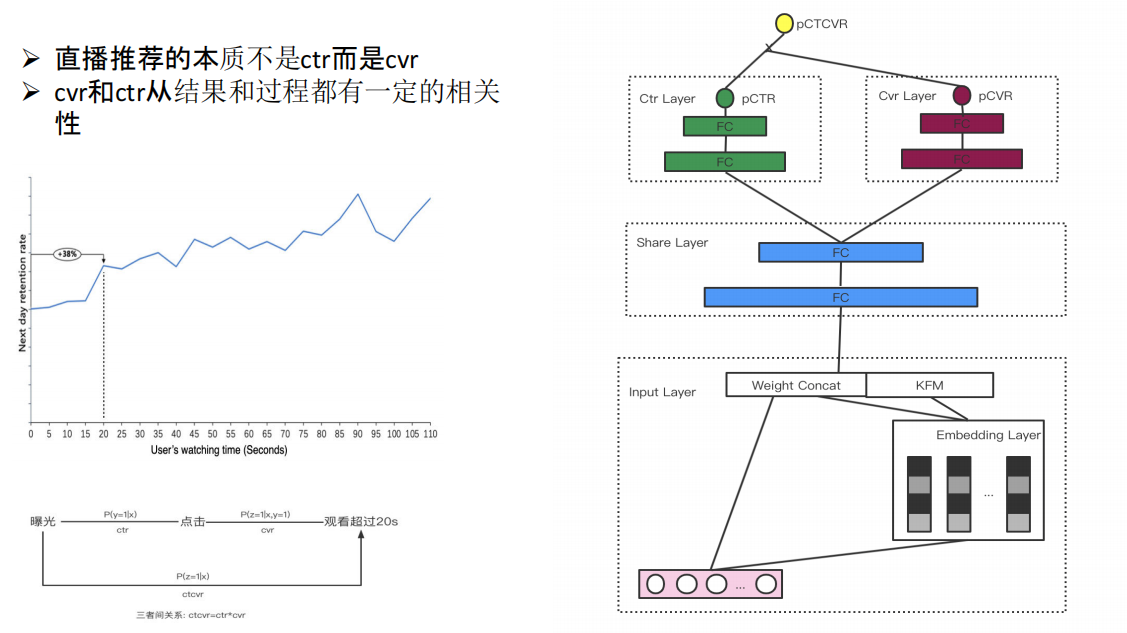


**下面来讲一下我们的低阶特征交叉.从左上角的图可以看到现在的深度神经网络普遍流行的一个并行架构,一部分是隐形交叉,这部分N侧,我们的总原则是通过一定的调整来调整这个N的宽度或者深度来保证一定的水准就可以了.主要的工作还是在特征交叉这一部分.对于现在的业界,DeepFM还是一个比较普遍的baseline.因为这样的低阶特征实际是可以通过向量的内积来表达两两特征之间的交叉.在这个基础上,有人提出了NFM,其实就是把内积换成了哈达玛积,那么我们来进一步思考:其实FM的本质就是两两特征之间的核函数就能够表达特征交叉,这里要思考一个问题:FM的一个通用的问题是什么?这个问题就是embedding长度需要设置成一样才能保证两两特征的向量能够进行点积或者哈达玛积再或者内积.那么我们怎么解决embedding不定长或者我们怎么能够设置一个更加普适的核函数来把所有的低阶特征映射到一个比较通用的空间来进行交叉?经过一系列的特征公式推导,我们发现有一个核函数能够在tensorflow中被更好的定义并且它的计算复杂度是O(n+T),T就是我们的特征的Bit级别的长度,这种情况相当于我们直接对Bit级别的特征做FM,然后定义了一个普适的核函数.这个的效果就是我们能够自由地定义每个特征的Bit size(也就是说embedding的size),从而能够达到两两特征交叉的目的.那么在这样的情况下,KFM比FM有四个千分点的提升.**

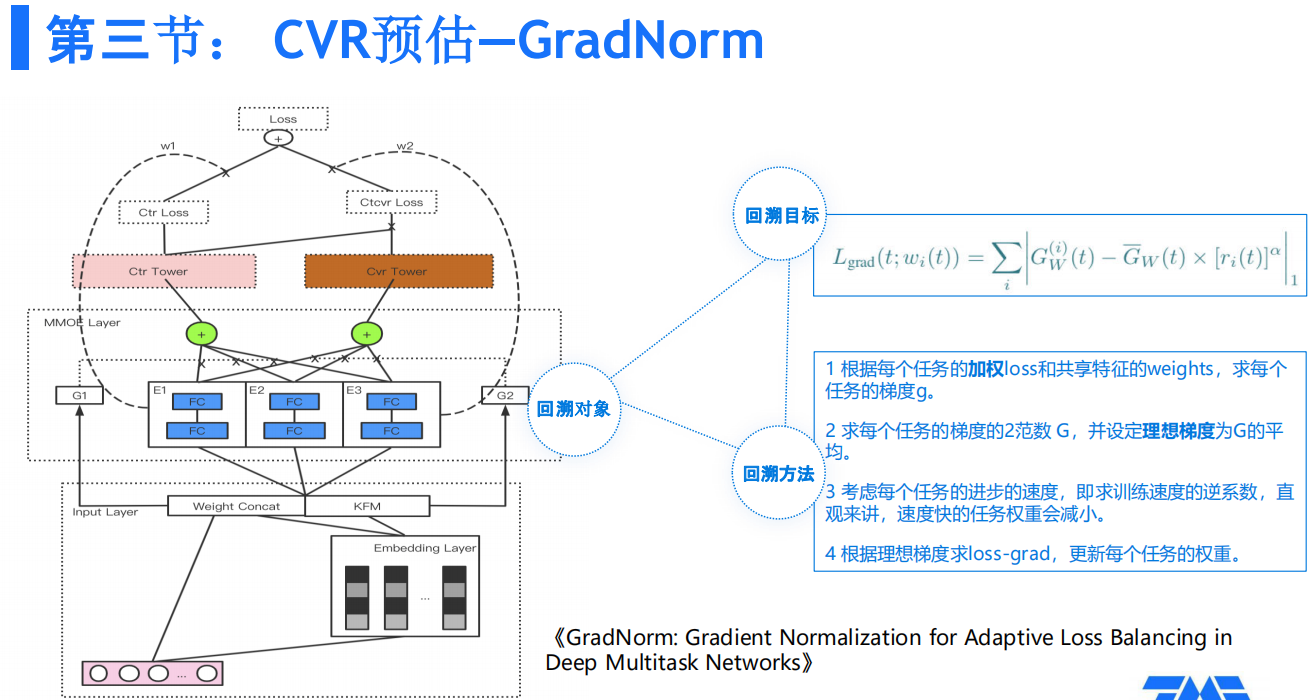


对于高阶的特征交叉,我们也是走的和业界比较通用的路线,比如Cross,只通过这样一个公式来进行层层叠加来达到高阶特征叠加的目的.然后AutoInt就是把我们的每个特征去比作一个词来做多头的self-attention进行高阶特征的交叉.这样其实在我们的场景中也是能得到一些收益的,但是基于FM之前的一些思路,我们仍然想要把每个特征设置成不同的embedding.这样的不同可以给我们带来比较大的好处,因为比如说年龄,性别这样的embedding一般都会比较小,而user embedding往往都比较大,这种情况下,其实更适合Bit级别的特征去做self-attention交叉方式.所以我们唯一的变化就是把交叉的力度变成了Bit级别.这样的AutoInt也给我们的AUC带来7个千分点的提升

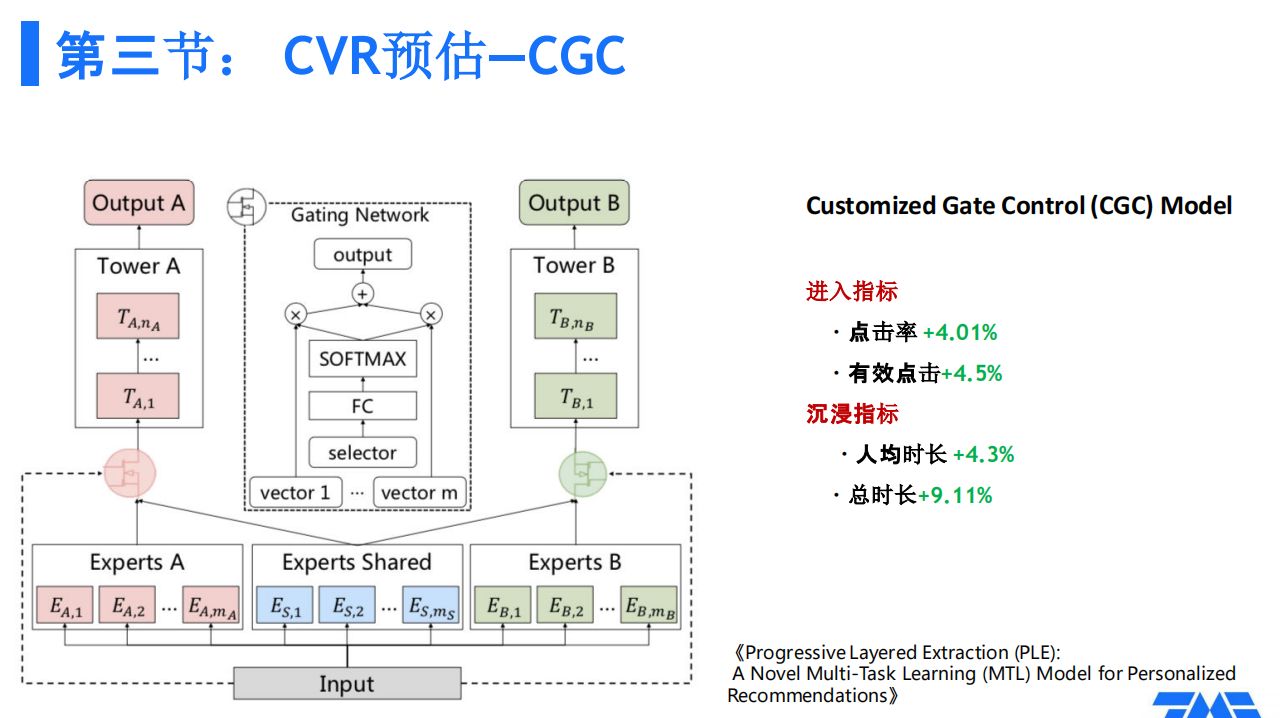
1. CVR预估--ESMM,GradNorm



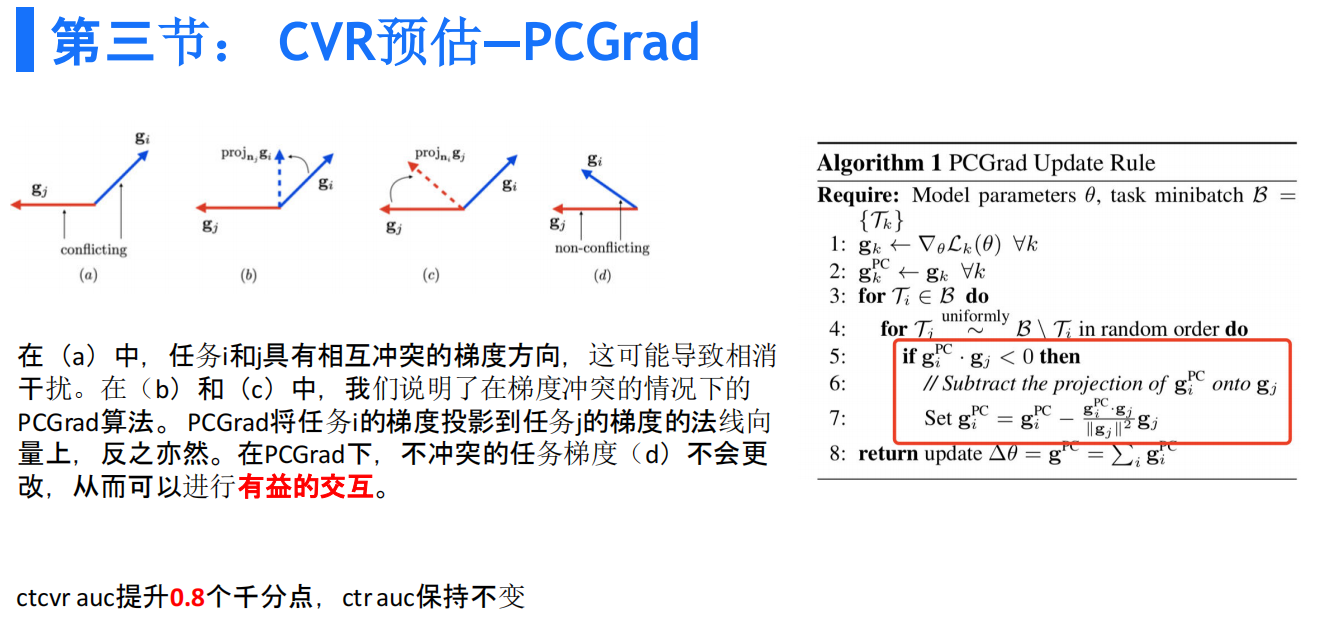
从一系列的ctr预估优化中,大家也会明白:对于直播场景,直播推荐的本质不是CTR而是CVR.因为我们需要让用户get到直播间的内容,而不是让他点进去.CTR和CVR是有一定的用户路径的,也就是说我们要先去点然后才能成为CVR.而且CVR对于我们最终的留存是有一定的相关性的.如果我们超过一定的时间,用户的直播的留存就会有一些比较显著的提升.在这样的思考下,我们也采用了一些业界比较通用的方案,就是ESMM.这样的通用方案其实解决了一些比较通用的问题.第一,就是选择的偏差,从CTR到CVR是一个选择的偏差.第二个问题就是CVR这样的问题在我们的场景里也是比较稀疏的.在最终的目标上面,我们也进行了一定的思考.首先这是我们业务需要的CVR和CTR,并且用户习惯的CTR和CVR,都可以用这两个塔来进行表示.



并且在以ESMM为baseline的基础上,我们还发现了一些问题:CVR是一个很重要的任务,而这个任务在多个任务一起学习的过程中可能会没有任何收益.原因是这个任务太过难以学习,在这个前提下,我们参考了另外一个比较通用的框架:GradNorm.在学习的过程中,如果我们手动调节CVR,CTR的权重也是可以的.但是这样的调节会有比较多的弊端:一方面是手动的调节不能做到端到端的调节,不能动态的调节,因为我们训练的周期和权重,理论上是不一样的.另外一个点就是多个任务之间的权重,包括更多的因素:训练的速度,收敛的速度,学习的梯度需要参考这样的一个框架:在训练完之后我们需要回溯,回溯的对象就是最后的一层共享的embedding,回溯目标也就是归一化的梯度,通俗一点就是希望我们的任务能够齐头并进地去进行学习.这种情况下,可以在tensorflow中打印出每个任务的权重,在训练过程中,CTR,CVR的权重会占比更加高,会给到一些倾斜,使得一些比较难的任务能得到一些补充信息来更加好的学习到CTR,CVR的任务.在线上的对比实验中,从CTR,CVR不同的打分对比来说,从alpha=0到alpha=1,我们的打分策略是CTR\*CTR或CVR的predict分数的积的alpha次方.在这样的权重条件下,我们的GradNorm都是取得了一些不错的成绩



我们还思考怎样能使GradNorm学习的更充分一些,能更好的调节两个任务的权重,那么是不是应该有一个公共,独享的专家,这里正好有一篇2020年的论文在讲CGC,它就是一个定制共享的机制,讲的是我们把MMOE的机制做得更精致一些.有一个共享专家能把我们的GradNorm学习的更充分一些,能更充分地调节出一个梯度的归一化.除此之外,还有一个Expert A和B,这样两个专家是可以有更多定制化的空间,这样共享的专家对独有的任务是有更大的收益.我们在上了CGC以后,在点击率和有效点击,包括时长方面,两个目标都会有很大提升.



事实上,GradNorm也有一定的缺点:GradNorm本身其实是做一次比较粗暴的回溯,只是从最后一层回溯最开始的一层,而且它调节的是一个权重,那么除了调节权重,我们还能做什么?是不是能够更加精确的去控制这个权重的过程.因为个人认为:训练的精髓在于梯度,如果我们能够直接影响到每个任务的梯度,那么其实就是能够对每个任务中的相互的受益做出更有效的判断,使得任务之间能够做出一些更加有意义的交互.举个例子:我们看到一篇PCGrad这样一篇论文,当两个梯度它们比较相似的时候,我们会计算cos的相似度,当两个任务的梯度是比较相似的情况下,我们会继续沿用每个人物的梯度;当两个任务梯度不太一样甚至冲突的情况下,,我们映射到一个梯度的空间,这样的话,就相当于在梯度的层面让任务进行相互的融合,下一步,我们会利用修正后的梯度来进行梯度下降这样的优化,使我们的模型训练的更加鲁棒.