深度学习在推荐系统中的应用｜排序篇

摄影师王同学 搜索与推荐Wiki 今天

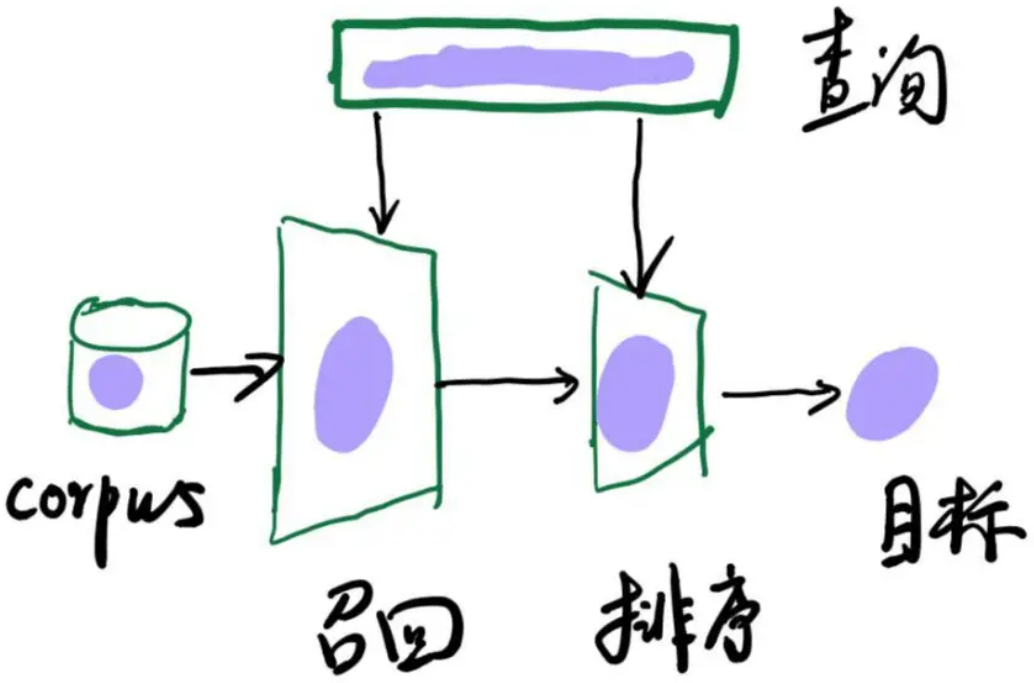
收录于话题
#推荐相关笔记 35 #精品小系列内容 28

本系列分为两篇：召回和排序。本部分介绍深度学习在推荐系统中的排序应用。

出处：https://www.zhihu.com/people/wang-jian-zhou「文末阅读原文直达出处」
作者：摄影师王同学
编辑：搜索与推荐Wiki
说明：重排原文，一起学习

1.整体架构

在现代的推荐系统，由于数据的扩张远远超过算力的增长，外加经济型的考虑，所以架构呈现出分漏斗的多阶段处理，一般整体架构图如下：



从最下游的corpus库一般是亿级别item，再经过轻量级的召回系统，item的规模缩减到千百级别，最终千百级别的item会进入排序系统得到十级别的item做为目标输出（推荐系统一般视实

际情况可能还会有业务过滤模块：不合适以及重复内容的过滤、二次精排等）。

在前边已经对召回系统的算法做了详细的介绍：深度学习在推荐系统中的使用丨召回。

本节将对排序算法在推荐系统中使用做了一个简单介绍，一般来说排序系统有以下要求。

- 1.输入：召回系统返回的少量item（item的特征）；查询，在推荐系统来说就是user（user的特征）；上下文特征（典型的如当前时间、用户所在的设备）
- 2.输出：对召回的item进行某种策略的排序，一般来说排序越靠前的和业务目标越吻合，如更容易被用户点击、购买等等（在一个良好的排序系统为了满足输出多样性或者全局收益更大，可能不是严格按照原始预测顺序排列）
- 3.工程化要求：RT、吞吐量、硬件成本等

2.建模分类

一般来说，在排序场景会将真实的业务目标通过以下几种方式建模：

之前也整理过一篇关于建模分类的文章：怎么理解基于机器学习“四大支柱”划分的学习排序方法

2.1 PointWise

基于单点打分模型，对每个参与排序的item计算得到一个Score，Score是一个相对分，可以没有物理意义，只要和业务目标越接近的item Score越高即可。单点样本参与计算，所以样本的构造简单，模型的具有较好的实时更新性，模型的计算量也相对较小。缺点就是在输出的列表没有考虑到参与排序的item之间的相关性带来的影响，也就没有考虑到整个列表序列的全局最优。

2.2 PairWise

对参与排序的item 两两比较，计算和目标优先级从而得到排序，例如参与排序的三个样本a,b,c,d，得到a>b，b>c，b>d，d>c 最终排序目标为a->b→d→c。考虑了训练样本中两两的相关性，模型的精度会有提升。但是PairWise抗噪能力偏差，对样本准确度较高构造样本成本偏高，因为无法构造出所有的pair对样本会导致模型学习有偏差。

2.3 ListWise

对参与排序的item中的每一个预测出一个在目标列表中的位置。一般来说ListWise 排序的精度最高，但是构造样本的成本最高。

在推荐场景，由于参与模型计算的item样本量较大且更新频繁，训练样本一般不会由人工构造，而是由最终用户的行为反馈来构造，PointWise模型非常适合此类构造的样本（如点击得1分，不点击得0分），而PairWise和ListWise则无法构造出准确的样本，此外推荐场景用户没有确切的意图，一般来说对排序精度相比搜索没有过高要求，所以最常用的建模方法就是PointWise，本文主要针对PointWise的模型做分享。

基于PointWise，首先需要构造正负样本对，一般来说正样本相对容易构造，用户发生过正向行为的item既可算作正样本，比如新闻推荐场景，用户发生过点击行为，但是这里要注意各种异常噪音如误点击等，负样本选取除了曝光未点击，还要混入全局随机抽样item，热门item，避免曝光样本是因为其他策略（召回）生成带来的偏置（注意采样的数据最好是客户端埋点）。

其中正样本的label记为1，负样本记为0，这样基于PointWise的模型需要处理的就是二分类问题，下边将分享一些有代表性的基于PointWise的排序模型。

3.传统模型

在很多场景中，传统的机器学习模型+设计良好的特征工程就能取得很好的效果，此外由于传统的机器学习具有可解释性，可以直接在数据上进行人工干预（比如显式调高某个item重要的特征值从而提高排序），而不用设计复杂干预系统。

可以优先尝试LR，XgBoost，若数据量不够多，可以尝试SVM等。

在传统的机器学习中，重点设计特征工程，可以快速验证特征对目标效果的提升，更多关于特征工程可以查看：https://zhuanlan.zhihu.com/p/245178672

在传统的机器学习中，也可以尝试超参数自动搜索，获取最佳的模型，下边是利用网格搜索搜索xgboost的示例代码：

```
params={
            "objective":"binary:logistic",
            "eval_metric":"error",
            "num_class":2,
        }
        grid_search_parameters={
```

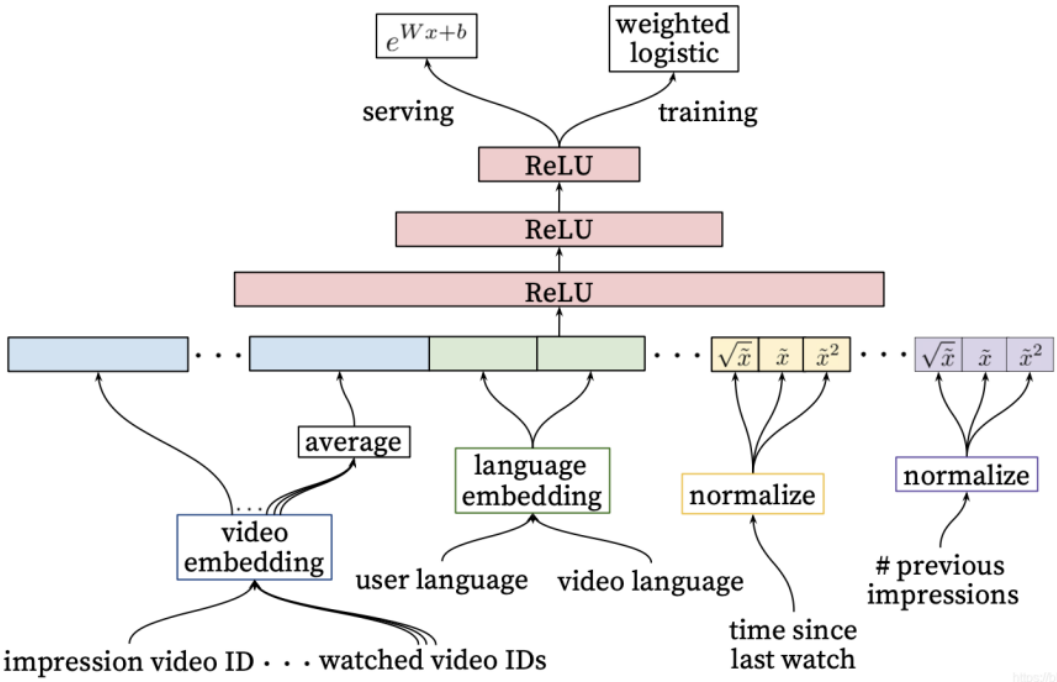
```
    'n_estimators': [160,170,180,190],
    'max_depth':[6,7,8],
    "eta":[0.1,0.15,0.2],
}
xgb_classifier=xgb.XGBClassifier(
    **params
)
optimized_xgb = GridSearchCV(estimator=xgb_classifier,
                             param_grid=grid_search_parameters,
                             cv=5,
                             n_jobs=32)
train_feature,train_label=train
optimized_xgb.fit(train_feature, train_label)
```

4.深度模型

随着2010年深度学习模型在图像、语音、自然语言上取得的成功，研究者开始将深度模型逐渐应用到推荐系统上，因为深度学习模型良好的数据拟合能力以及对原始特征不需要做太多的特征工程，现在在应用上使用越来越广泛，一些在其他任务上取得的成功的网络，如DNN、Attention、RNN、多任务学习都被移植到推荐系统的排序上，下边将重点分享具有代表性的一些网络设计。

4.1 DNN系列

4.1.1 YoutubeDNN



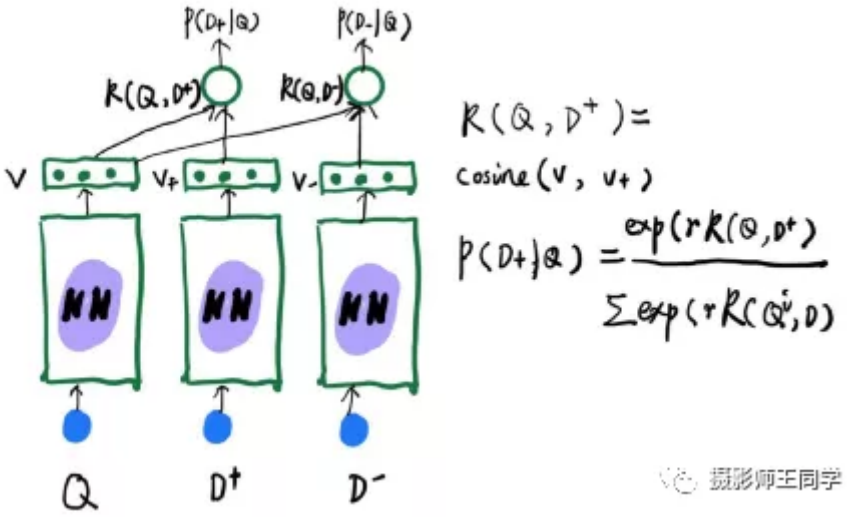
Youtube的排序模型从网络设计上看，没有特别出色的地方，标准深度全连接网络，采用weighted logistic regression 训练，模型能借鉴的主要在两个地方：

- 1.作为深度模型，依然做了较多的特征工程，比如特征中加入曝光次数、用户所在区域的语言种类、视频的语言种类，此外对特征先验的做了各种变化，如标量特征做了开根号，平方并和原始值一起做了归一化，期望可以捕获更多差异信息

- 2.在训练时采用权重回归，权重就是用户观看正样本的时长；预测时直接采用odds（发生比）近似模拟用户观看视频的时长期望

具体的解释可以查看这篇文章：结合论文看Youtube推荐系统中召回和排序的演进之路（中）篇之深度召回排序

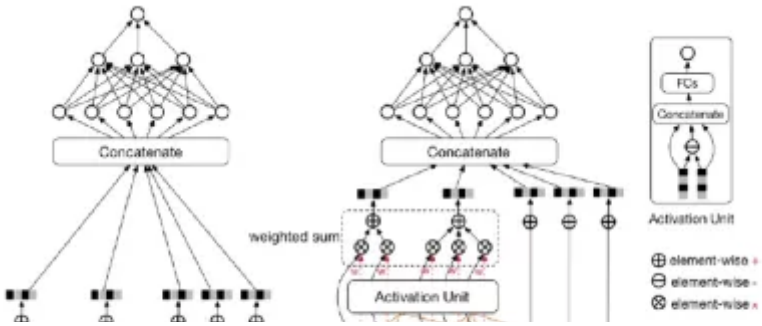
4.1.2 微软DNN

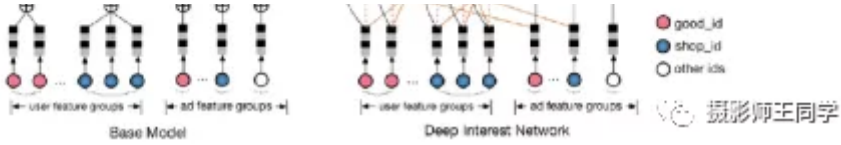


在推荐场景，Q就是用户，D是待推荐的item，同时构造一正多负的样本去训练模型从而更好的区分正负样本，在线上提供服务阶段，只需要输入Q和D得到两者cosine的相似度做为排序分（注意训练阶段多个D的网络参数是共享的）

4.2 Attention系列

4.2.1 阿里的DIN





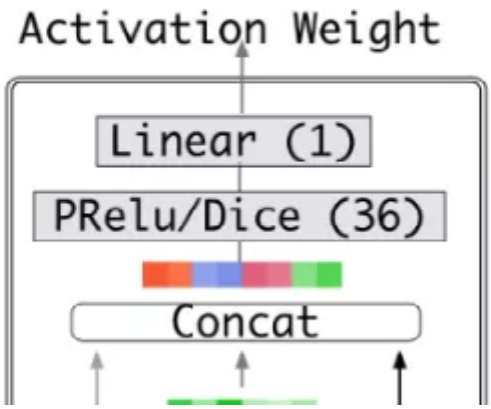
注意力机制在CV和NLP上都取得非常大的成功，在推荐上融入注意力机制的优秀模型也越来越多，比如以前分享过的微信的RAML就多阶段的融入注意力机制来提高效https://zhuanlan.zhihu.com/p/268180582 《Looklike 介绍和微信的RAML论文分享》 。

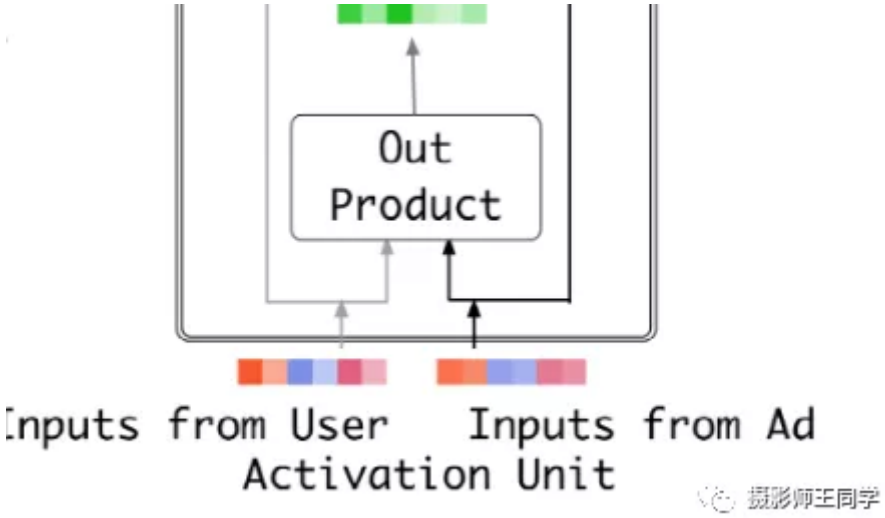
不过在注意力引入到推荐排序上，阿里云的DIN 深度兴趣网络算是开山之作，而且模型结构简单解决问题场景容易理解，所以理解这个模型的结构对把Attention引入推荐有很大的帮助。

回顾下YouTubeDNN的模型，输入的特征中最重要的就是用户发生正向行为items的embedding，在整个网络中对这些items的embedding做了pool操作，也就是对等看待所有的items，假如在下边的场景：

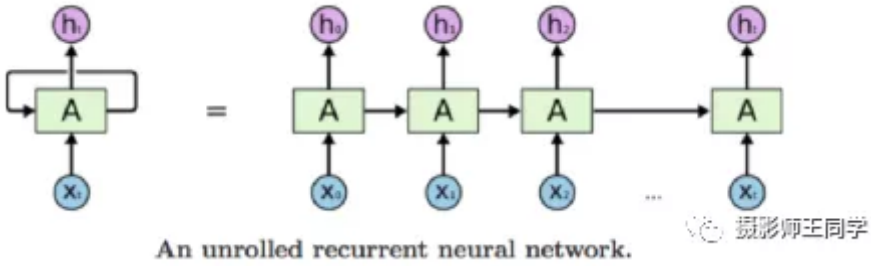
用户的行为兴趣一般是非常多样的，在计算用户对大衣的排序打分时，显然不希望用户行为中的杯子、包等和大衣关系不大item对模型起太大的作用，那么直观的思路的就是对类似不重要的item的embedding给一个比较低的权重，这个就是Attention的思路，在模型的训练过程中，自动学习到行为item和目标item的权重，然后在sum pool的时候不是以前粗暴的直接相加，而是乘以学到的Attention 权重后再相加。

以上就是DIN相比 YoutubeDNN 最大的改进，其中注意力模块的权重由以下的模型学出，用户行为中每个item和待推荐的item经过非常简单的拼接和非线性变换得到Attenion的激活权重（Attention 计算方式很多，这里只是一种，核心要点就是两个需要对齐Item的特征一并进入模型，最终得到和Item特征维度一致的权重向量做为Attention权重）





4.3 RNN系列



An unrolled recurrent neural network.

前文提到Youtube的DNN 或者阿里的DIN ，存在一个严重的问题就是用户的行为顺序在模型中没有起到任何作用，但是在实际的场景中，行为顺序会反应出用户的兴趣变化，对推荐结果产生重要影响，这里的一个改进的思路就是把用户的行为Item按照时间排序后当做一个序列，输入RNN序列网络进行学习和表示。这个时候用户的行为的特征可以由以下3种方式输入到后续的网络中：

- 1.RNN网络的最后一个时间步的输出
- 2.RNN所有时间步的输出求pool
- 3.RNN每个时间步的输出 和目标Item 计算Attention 后求Attention pool

类似的RNN思路比较直白简单，不做过多赘述。在RNN里核心分享一篇来做MSRA的《Adaptive User Modeling with Long and Short-Term Preferences for Personalized Recommendation》https://www.ijcai.org/proceedings/2019/0585.pdf

这篇论文有3大创新点：

- 1.用户的短期兴趣和长期兴趣分开建模，短期兴趣用更复杂的模型学习；长期兴趣用attention 学习权重，加权长期行为
- 2.短期兴趣和长期兴趣采用动态门融合机制
- 3.对于短期兴趣学习，考虑了不同行为的时间间隔不规则和不同行为的topic不同

其中1和2就是Attention的在不同场景的不同用法，结合论文和前边提到的Attention机制，读者自行理解。在这里核心分享和RNN相关的第3个创新点。

在前边提到的朴素的RNN的用户行为序列，存在一个比较严重的问题，在传统的RNN网络中，每个step的输入特征和前后步的信息传递上是对等的（可以简单认为就是每个时间步的时间间隔是相等的），但是在推荐系统用户行为基于这个假设没法及时捕获用户的兴趣转移，以及在信息传导上会出现严重的偏差，比如在一个视频网站，一个用户有如下行为日志：

<4月1日，观看a>，<4月2日，观看b>，<5月1日，观看c>

我们希望的是从A传递到B的信息，比从B传递到C的信息多，为了达到这个目的，论文针对原始的RNN系列的LSTM做了相应的修改，回顾标准LSTM cell的计算公式：

$$f_k = \sigma(x_k W_f + h_{k-1} U_f + b_f) \quad (2)$$
$$i_k = \sigma(x_k W_i + h_{k-1} U_i + b_i) \quad (3)$$
$$c_k = f_k \odot c_{k-1} + i_k \odot \phi(x_k W_c + h_{k-1} U_c + b_c) \quad (4)$$
$$o_k = \sigma(x_k W_o + h_{k-1} U_o + b_o) \quad (5)$$
$$h_k = o_k \odot \phi(c_k) \quad (6)$$

由遗忘门f_k、输入门i_k、输出门o_k，单元状态c_k和输出h_k 5个部分组成，可以看到参与计算的特征只是输入特征和上一步的输出特征，本论文中新使用两个基于时间学习到特征：t_k、s_{t_k}

其中：t_k代表 发生第k个行为时的时间；所以 δ_{t_k}代表的当前行为和上一个行为的时间间隔特征，s_{t_k} 代表预测商品行为和当前行为的时间间隔特征，其中时间差取了对数做了平滑。

基于上述的时间特征，新学习了两个控制门（时间特征和输入特征组合学习得到的控制门）：

$$\delta_{t_k} = \phi(W_\delta log(t_k - t_{k-1}) + b_\delta)$$
$$s_{t_k} = \phi(W_s log(t_p - t_k) + b_s)$$

这时lstm的单元状态c_k被改造成为了：

$$c_k = f_k \odot T_\delta \odot c_{k-1} + i_k \odot T_s \odot \phi(x_k W_c + h_{k-1} U_c + b_c)$$

基于上一个行为和当前行为的时间间隔的门控和遗忘门共同点积做为新的遗忘门（时间越长，遗忘的越多），基于预测商品行为和当前行为的时间间隔的门控和输入门点积做为新的输入

门。

输出门也融入这两个时间特征被改造为如下：

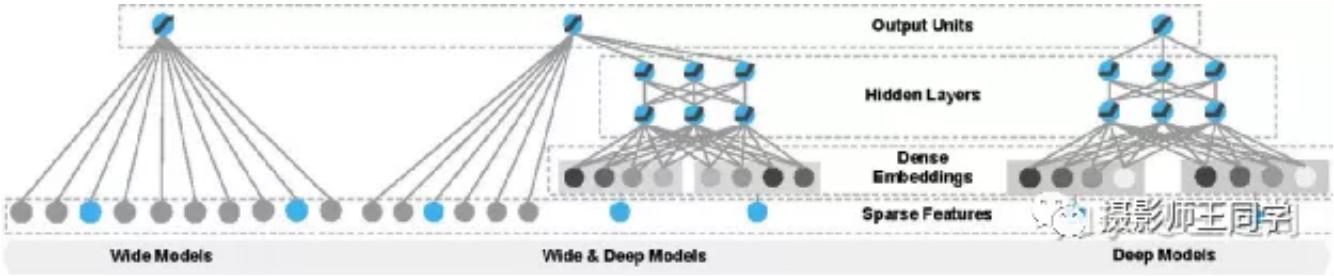
$$o_k = \sigma(x_k W_{xo} + \delta_{t_k} W_{\delta o} + s_{t_k} W_{so} + h_{k-1} V_{no} + \cdots$$

整套公式非常优雅，除了将用户的行为做为序列建模，还考虑行为间的不同时间间隔带来的影响。

4.4 融合模型

4.4.1 Wide&Deep 模型

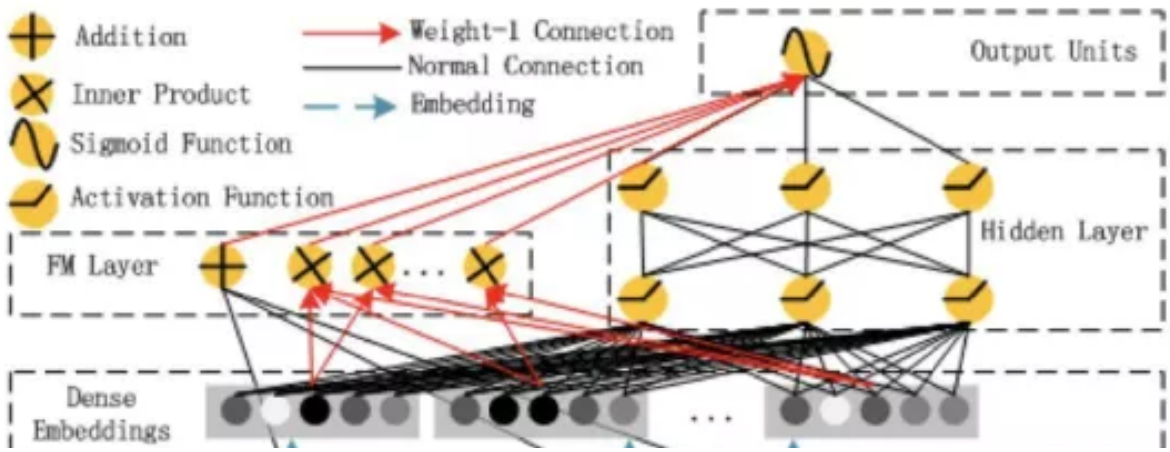
前边提到的深度模型有更好的特征捕获能力，但是很难融入人类的先验知识，此外因为缺乏可解释性也导致很难进行人工干预，工程师肯定不希望写很多逻辑分支进行人工干预，这个时候的改进思路之一就是进行传统模型和深度模型进入融合，其中最具有代表性的就是Wide&Deep模型， Wide&Deep只是一种模型融合的思路，Deep的模型可以用各种网络结构的模型。



图片

4.4.2 Deep &FM 模型

在推荐场景，有非常多的id 稀疏特征，类似的one-hot特征输入DNN会导致巨大的训练计算量，如果直接输入Wide会导致特征之间的相关性无法被捕获，这个时候可以利用FM对稀疏特征做二阶交叉，从而捕获稀疏特征的关联性，这个时候模型相比Wide&Deep ，就是用FM取代Wide层。



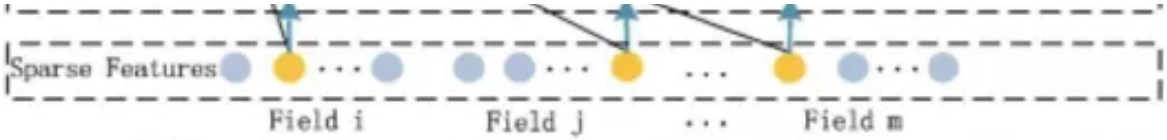


Figure 1: Wide & deep architecture of DeepFM. The wide and deep component share the same input raw feature vector, which enables DeepFM to learn low- and high-order feature interactions simultaneously from the input raw features.

图片

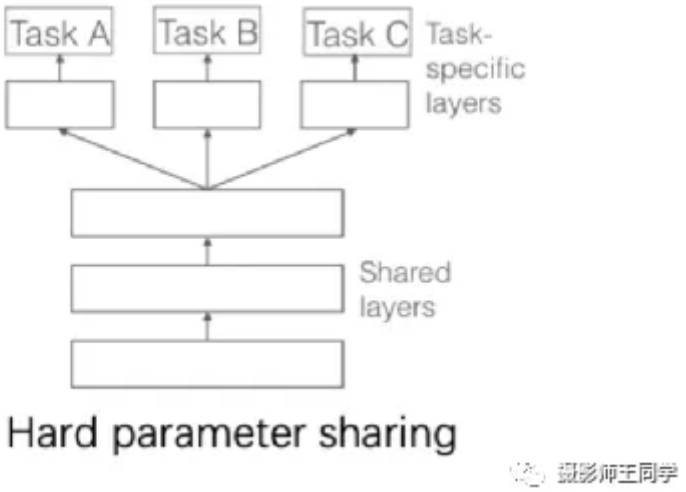
更多关于Wide&Deep和Deep&FM模型更多参考这篇解读文章：https://zhuanlan.zhihu.com/p/314074755 《Wide&Deep模型讲解附代码》

4.5 多任务模型

在深度学习中，一般来说如果多个任务有关联，采用多任务训练得到的效果相比两个任务分开训练都有提升（减少过拟合和减缓噪音数据对模型的影响）；此外一般在一个机器学习业务中可能需要多个指标去考量，如果多个模型的话训练成本和在线服务的硬件成本和在线推理的rt都会偏大，比如在一个视频场景推荐场景的打开率和观看时长，业务上需要两个指标，也希望两个指标同时优化，那么这个任务就特别适合多任务训练。

多任务的模型一般来说有两种做法

1.hard-parameters

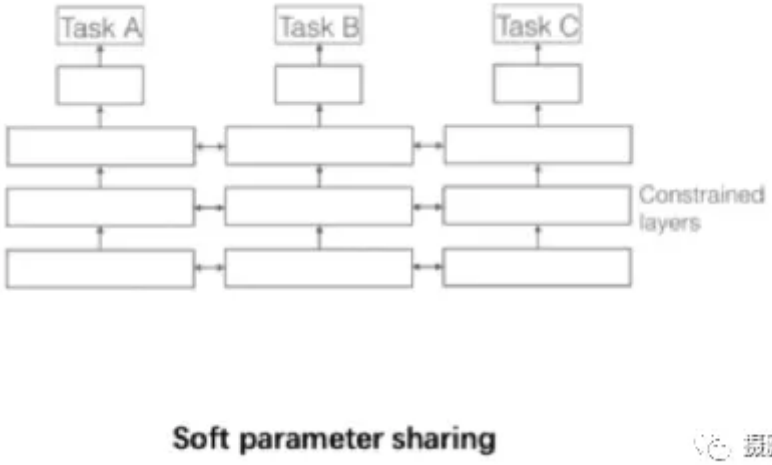


图片

这种做法一般都是多个任务共享部分相同特征提取网络，根据具体的目标再设计单独的网络分支来进行任务目标学习，这种网络设计优点：避免过拟合，提高模型性能，减少参数和硬件投入。但是这种目标也有一个较大的问题就是如果多个子任务相关性不是很强，特征提前网络表达会偏移，导致在每个任务上都表现不好。

2.soft-parameters

针对hard-parameters的子网络相关性不足会导致效果不佳，针对性的改进的多任务学习的网络如下：

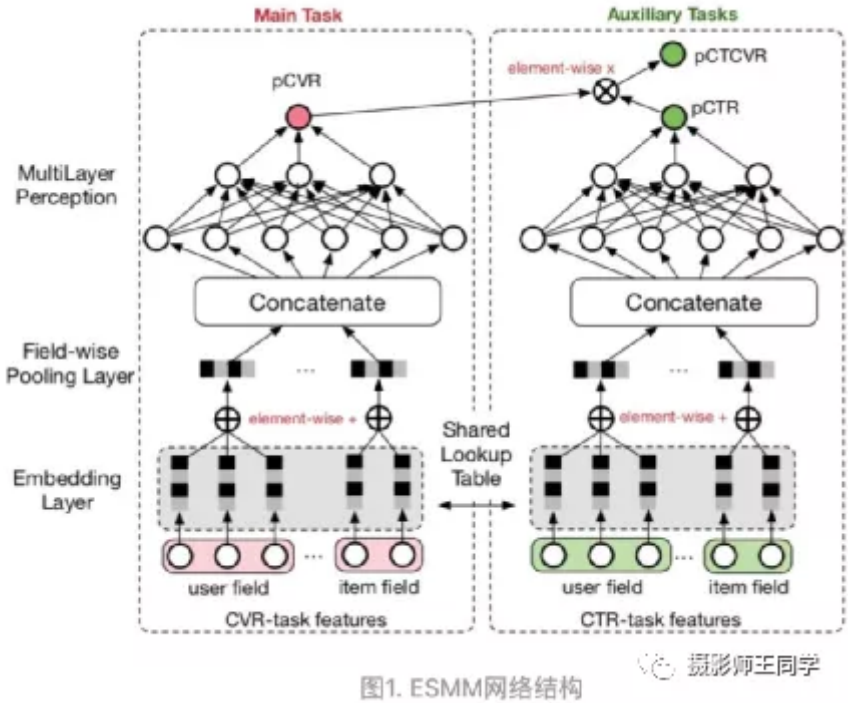


图片

主要的思路是对硬共享的底层参数，每一路上游任务通过网络再学习决定共享那些底层网络的输出以及是否要做相应的调整。

针对这两种多任务学习排序网络分别介绍两种代表性落地方案。

1.阿里的ESSM



图片

在推荐的任务中，需要评估两个指标，CVR(点击后转化率）、CTR（点击率）。

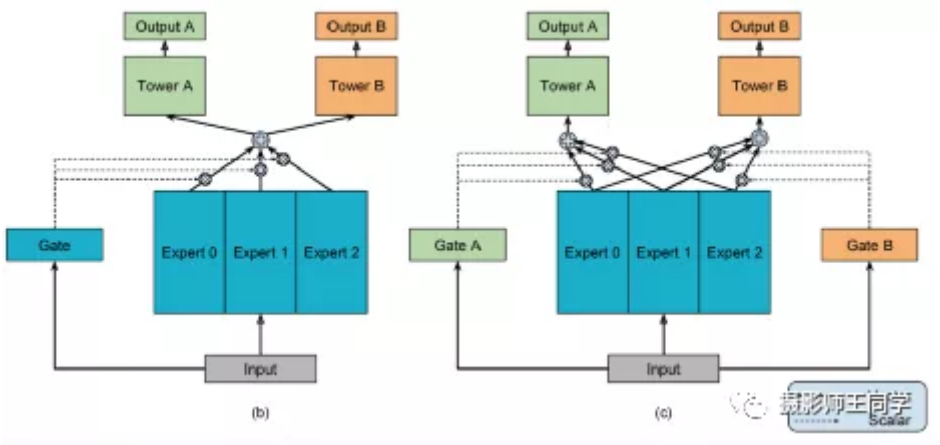
在原始的CVR预估中，参与训练样本是点击后的item，这种做法带来了一定样本偏差（点击的样本是经过模型干预的）；此外由于点击的样本量级非常少导致样本稀疏（特别是长尾的商品），所以通过ESSM来做多任务训练来同时得到CVR和CTR，为了在全量的样本中训练，ESSM引入了辅助指标CTCVR（点击后转化率），很容易得到以下公式：

$$\underbrace{p(y=1,z=1\mid \boldsymbol{x})}_{pCTCVR}=\underbrace{p(y=1\mid \boldsymbol{x})}_{pCTR}\times\underbrace{p(z=1\mid y=1,\boldsymbol{x})}_{pCVR}$$

图片

整个网络是典型的双塔网络，多任务是前边提到的hard-parameters共享了特征的Embedding，CVR和CTR任务不共享上游的DNN层。注意在网络训练中不利用CVR任务的loss，而是直接采用CTR和CTCVR的loss。

2.谷歌的MMOE网络



图片

左边B的核心可以表达为如下公式：

$$y_k = h^k(f^k(x)), f^k(x) = \sum_{i=1}^{n} g^k(x)_i f_i(x)$$

图片

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

图片

其中x是原始的输入特征，模型最核心的思想就是就是底层共享了多个子网络，但是上层不共享的网络输入不是原始共享网络的输出，而是对每一个共享网络学习一个gate参数，最终利用gate参数对共享网络的输出进行加权求和。

图c为多gate，对每一路的任务分支都学习相应的一组gate。

5.其他技术的应用

这一节简单同步下在推荐排序中会涉及的部分其他技术

5.1 生成模型

利用Seq2Seq 的思路将待排序的item作为序列以及其他特征作为输入，通过生成序列得到最终的推荐列表，由于推荐列表生成模型相比其他的序列生成模型（如机器翻译）缺乏有效的监督数据，用的相对较少，

比较代表性的有谷歌的**Seq2Slate: Re-ranking and Slate Optimization with RNNs** 。通过Seq2Seq+Attention（RNN网络采用LSTM）的指针网络生成推荐列表，loss为生成的每个step的item是否会被点击。

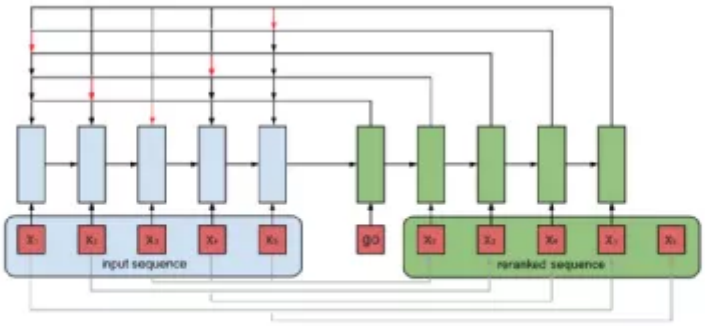


Figure 1. The seq2slate pointer network architecture for ranking

图片

以下为静态数据的评估结果：

Ranker	Yahoo			Web30k		
	MAP	NDCG@5	NDCG10	MAP	NDCG@5	NDCG@10
seq2slate	**0.67**	**0.69**	**0.75**	**0.51**	**0.53**	**0.59**
AdaRank	0.58	0.61	0.69	0.37	0.38	0.46
Coordinate Ascent	0.49	0.51	0.59	0.31	0.33	0.39
LambdaMART	0.58	0.61	0.69	0.42	0.46	0.52
ListNet	0.49	0.51	0.59	0.43	0.47	0.53
MART	0.58	0.60	0.68	0.39	0.42	0.48
Random Forests	0.54	0.57	0.65	0.36	0.39	0.45
RankBoost	0.50	0.52	0.60	0.24	0.25	0.30
RankNet	0.54	0.57	0.64	0.43	0.47	

图片

5.2 多模态特征

随着硬件处理能力的提高和深度学习的发展，输入到排序特征不仅仅限于用户行为、商品、用户离散或连续属性，越来越多的多模态特征也开始被输入到排序系统，比较典型的就是在电影和商品的推荐场景、直接将海报的图像做为特征输入，如果用了类似的多模态特征，有两种做法：

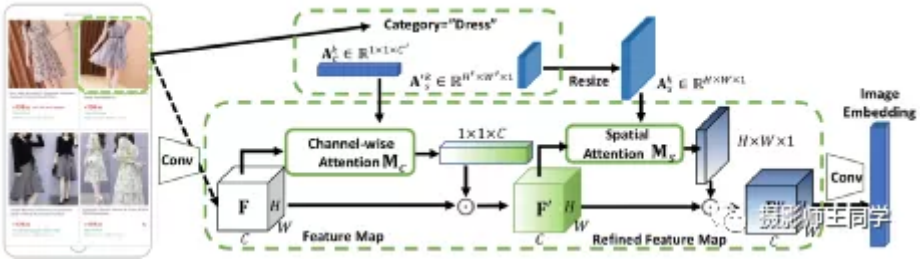
1.原始图像经过CNN网络得到的特征抽取网络和上游的排序网络联合训练

2.先预训练图像特征抽取模型，将得到特征保存，这个时候再用静态的图像特征训练上游的排序模型

由于图像特征抽取一般计算量都非常大，排序模型请求量又非常大，加上这种图像一般都是静止的，所以第二种方案用的较多。

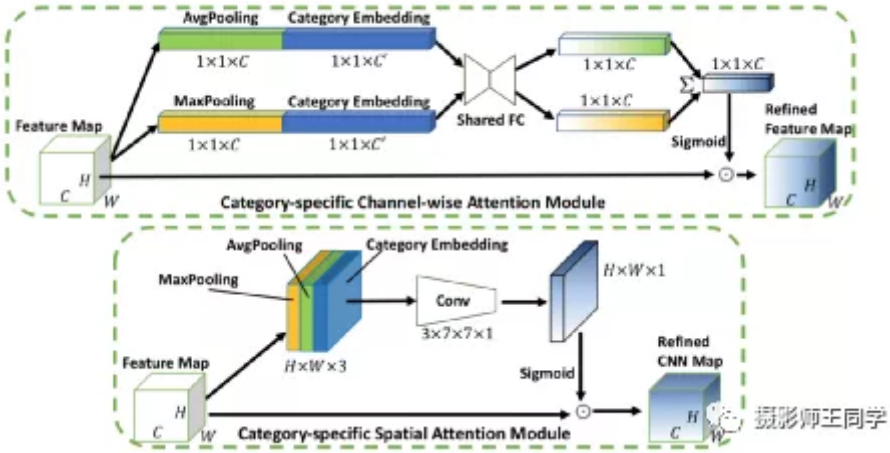
代表性的做法有 京东在kdd 2020中入选论文：**Category-Specific CNN for Visual-aware CTR Prediction at JD.com**

其中图像的的多模态特征用Resnet提前，经过Attention机制让特征更关注图像中的重要信息，其中这里的Attention机制如下：



图片

和CV中著名的CBAM Attention 基本一致，原始的图像特征分别经过channel和spatial的注意力模块，详见：https://zhuanlan.zhihu.com/p/280034277 CBAM论文解读。这里的一个改进，就是在channel和spatial的attention计算时 原始特征经过pooling后的特征还拼接了当前商品类别的embedding 。



图片

同样在视频、视频的推荐场景，越来越多的方向直接利用音频，视频等原始数据做为多模态特征加入排序模型。

5.3 模型蒸馏

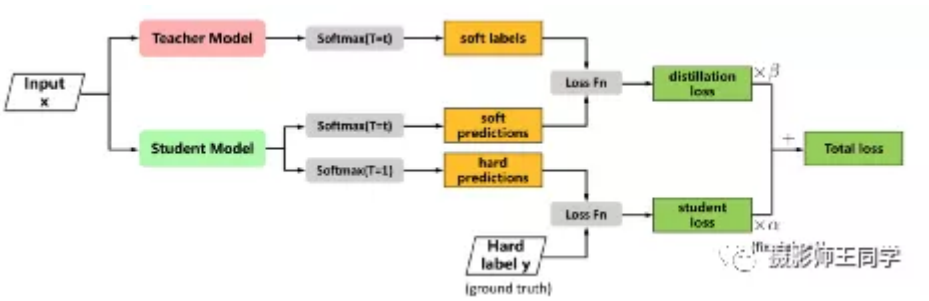
排序系统为了达到较高的准确率，在模型和特征输入都会比较大，这样就带来一些问题：计算延迟大且需要很多计算资源，而前边也说了在排序系统的请求量都是海量的且为了考虑用户体验对延迟极其敏感，为了兼顾效果和效率，往往要涉及到模型优化的工作，一般来说，模型优化有有三种做法：

1.量化，用更轻量级的参数单位如float16，in8代替原始的float32 精度参数；

2.剪枝，对于计算图中不重要的计算边直接拿掉，减少计算量；

3.蒸馏，通过大模型（高准确率模型）带动小模型来训练，让小模型的准确率逼近大模型；

1和2都是偏工程问题，这里分享3的做法，朴素的模型蒸馏示意图如下：



图片

首先先训练大模型（Teacher Model），Loss就是标准Loss，如建模目标是二分类的话就是交叉熵，等待Teacher Model训练完成后；开始训练Student Model，在整个Student Model训练过程，Teacher Model的参数冻结不更新，训练Student Model时 Loss由两部分组成：

- 1.Student 预测的结果和真值（hard label)的交叉熵
- 2.Student 预测的结果和 Teacher预测的logits softmax(soft label） 的交叉熵

除过这种朴素的模型蒸馏，在推荐系统中还可以进行优势特征蒸馏，也就是在Teacher Model可以加入一些计算量比较大但是能提升效果的优势特征（比如上文提到的图像特征），而Student模型不用这些特征从而减少计算量。

6.问题解答

1.Q：相较于大曝光量（样本量），转化率尚可（正负样本比例）的item，lgb为什么会为低曝光量（样本量），高转化（正负样本比例）的item给出更高的分数？采样数据的置信度如何在排序模型中体现？针对腰部、尾部等数据量较少的item，应该如何消解这个样本量偏置？

A：首先这个问题简单分析，应该用了转化率做为特征，而这个特征的权重非常高，而样本中低曝光的样本转化率很高，类似的的解决方案是采用贝叶斯平滑或威尔逊区间平滑技术平滑特征。

2.Q:当排序模型的优化目标是ctr，是否会更容易推出高ctr的item（而不考虑样本量的置信度），是否需要根据item的样本量对目标函数做修正？

A: 目标函数是ctr，模型就会去去优化ctr。样本量的置信度如果是指样本数量，可以通过前边说的特征工程来来解决，如果业务目标不光想优化ctr，比如还需要cvr或者gmv，可以设计多任务训练去解决。

3.Q:在排序模型中，某个item的样本量和这个item的正负样本比例在模型中的作用是如何体现的？

A:问题退化成简单问题，样本中两个item A和B ，样本的输入特征只有A和B的id特征。正样本A出现998次，负样本0次，输入特征为A id时，模型全部判断为点击，模型准确率有99.8，所以单个item的正负样本要均衡。

假如有1002个样本，A的正负样本各500次，B的正负样本各1次，那么模型对B训练不充分，导致预测将不准。

4.Q:在全域（包含了多个场景的曝光、点击数据）的样本下训练，最终在某一具体场景下进行部署验证，这种训练集数据和真实测试数据可能存在的抽样偏置是否会影响到最终指标？

A：如果多个子域关联性不大会影响，模型是去拟合样本，一般来说模型没法完全拟合样本，如果这个具体场景在样本中占比小导致拟合较小就会出现影响，

5. Q:训练集和测试集的分布如何判断是否存在偏差？何种程度的偏差是可以被接受的？如果偏差已经影响到效果的话应该如何消解？

A：参考前边特征工程中的异常值判断和处理：https://zhuanlan.zhihu.com/p/245178672。一般来数据（特征）决定了模型效果的上限。

6.Q:当排序模型效果不佳时，如何将问题归因于样本、特征或者是算法？

A：首先样本特征要保证准确，样本数量要充分（具体数量和算法模型以及场景问题密切相关），如果前这两者没有问题，再具体问题分析：

1.训练时的模型效果很好，但是测试效果不佳，主要检查过拟合的问题

2.训练和测试时效果都很好，但是线上效果不好，检查特征穿越或者关键特征训练阶段和线上预测不一致的情况。

3.训练时loss都很难收敛，检查模型的问题。

7.Q 推荐系统是否也可以采用预训练技术

A:可以，一种是预训练的模型整体迁移到其他推荐场景，比如从视频推荐迁移到音乐推荐，一般要注意：1迁移的场景需要尽量一致；2.模型的输入要尽量一致，避免关键输入特征的缺失；3.建模的目标要一致比如都是点击率预估。

上边这种迁移要求比较苛刻，更多的预训练是特征预训练，如5.3 提到的可以先预训练多模态的图像、视频特征，通过各种其他技术预训练用户、item的Embedding。

—完—