

CTR预估经典论文详解（一）——Wide&Deep模型

原创 炮屁儿 独立团、 2019-12-01

写在前面

近几年随着深度学习的蓬勃发展，CTR预估模型也逐渐从简单的广义线性模型逐渐发展为复杂的DNN模型，各大公司也都逐渐将DNN模型作为CTR预估的Base模型，所以打算开一个系列专门介绍近几年来比较经典的CTR模型论文，相信这些也是搜索、推荐、广告算法领域小伙伴们需要首先掌握的一系列经典DNN算法

一、论文背景

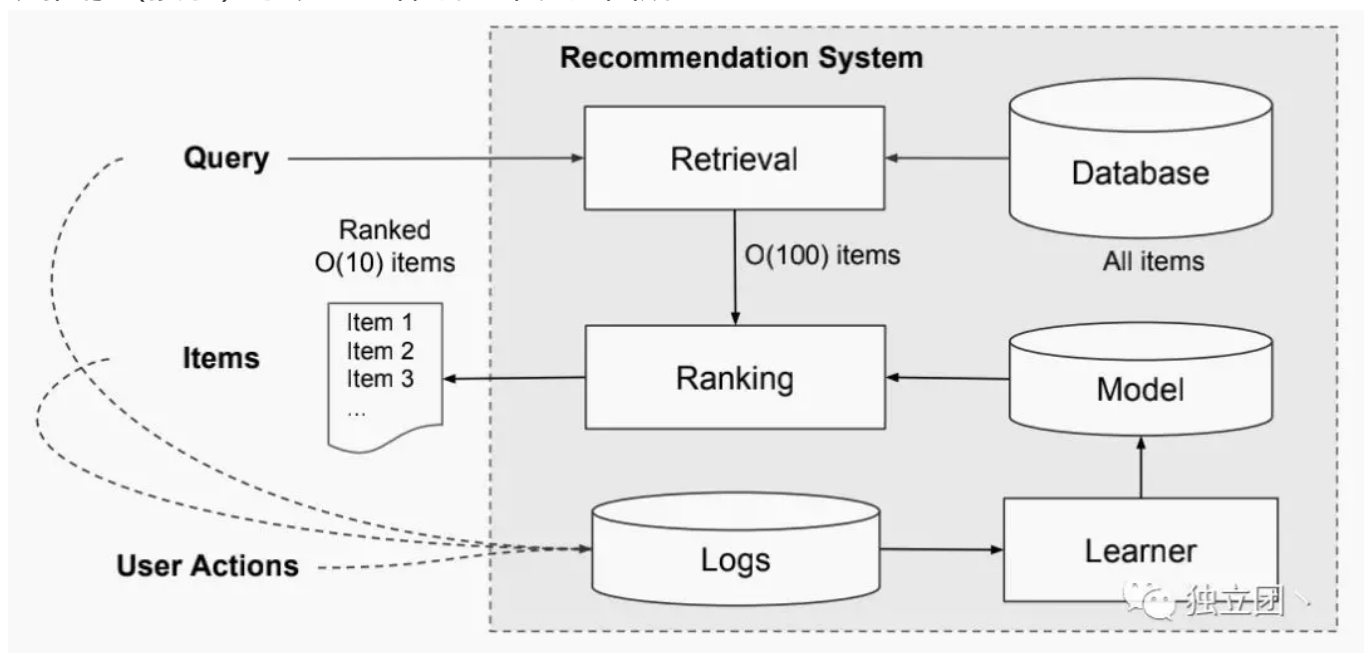
1、“wide”类型的模型利用cross-product方式进行特征交叉产生的特征具有很好的效果和很好的可解释性，但是需要较强的经验知识进行特征工程。

2、“deep”类型的模型通过将稀疏特征转换为稠密特征从而进一步挖掘高阶的特征组合，这种方式在用户-商品对比较稀疏的情况下，具备较强的泛化能力，可能给用户推荐不太相关的商品

3、为了进一步结合两者的优势，文章提出了一种全新的depp&wide网络结构

二、推荐系统整体结构

在之前的一些文章当中我们或多或少已经介绍过一些推荐系统相关的架构，总的来说推荐（搜索）系统的整体线上架构如图所示：



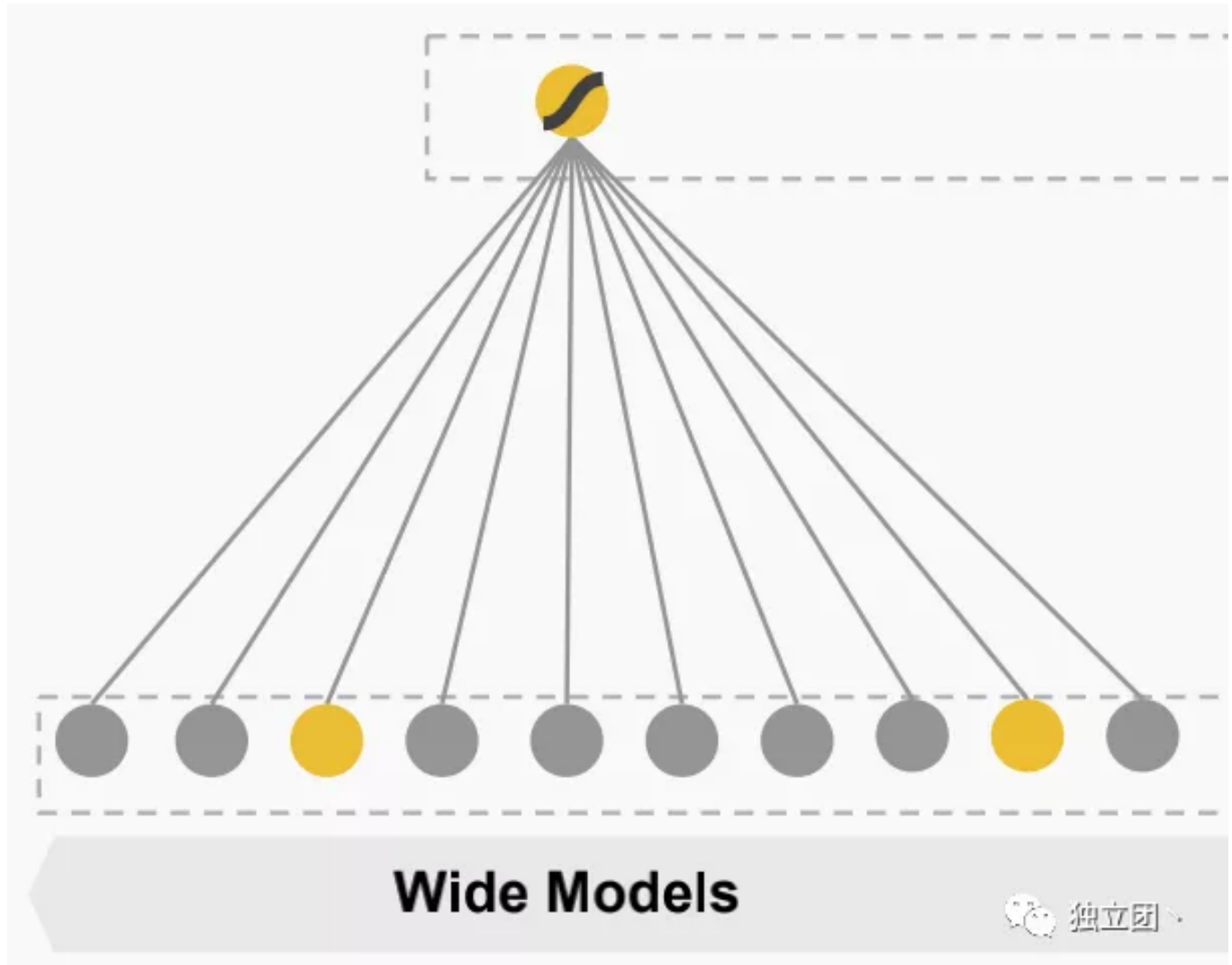
整个推荐（搜索）系统的工作方式如下：当用户发来一个请求的时候（携带着query信息以及访问的上下文信息），推荐（搜索）系统最终的任务就是给用户返回与其最匹配的信息与内容，这个寻找最匹配的过程主要包括如下两个方面：一是召回、二是排序。由于全部的候选商品可能有成千上万甚至上亿，如果我们在全部商品中直接对其进行打分排序的话，这个计算量是非常大的而且计算过程是非常耗时的，由于在用户进行访问的时候系统需要保证用户的用户体验，所以上述的过程需要我们在10-30ms内完成，不能让用户等待很久，否则用户体验会非常的差。所以召回阶段是很有必要的，

召回阶段的主要任务是从全部商品中选出一个和用户最匹配的候选集，这个候选集的规模是远小于全部商品集；筛选出候选集之后就这部分商品传递给排序模块进行排序打分，最终将得分Top的商品展现给用户。本文讲的wide&deep模型就是应用于排序模块的CTR预估模型。

三、Wide&Deep模型

1、Wide部分

wide部分主要就是一个广义线性模型（如LR），具体模型如下：



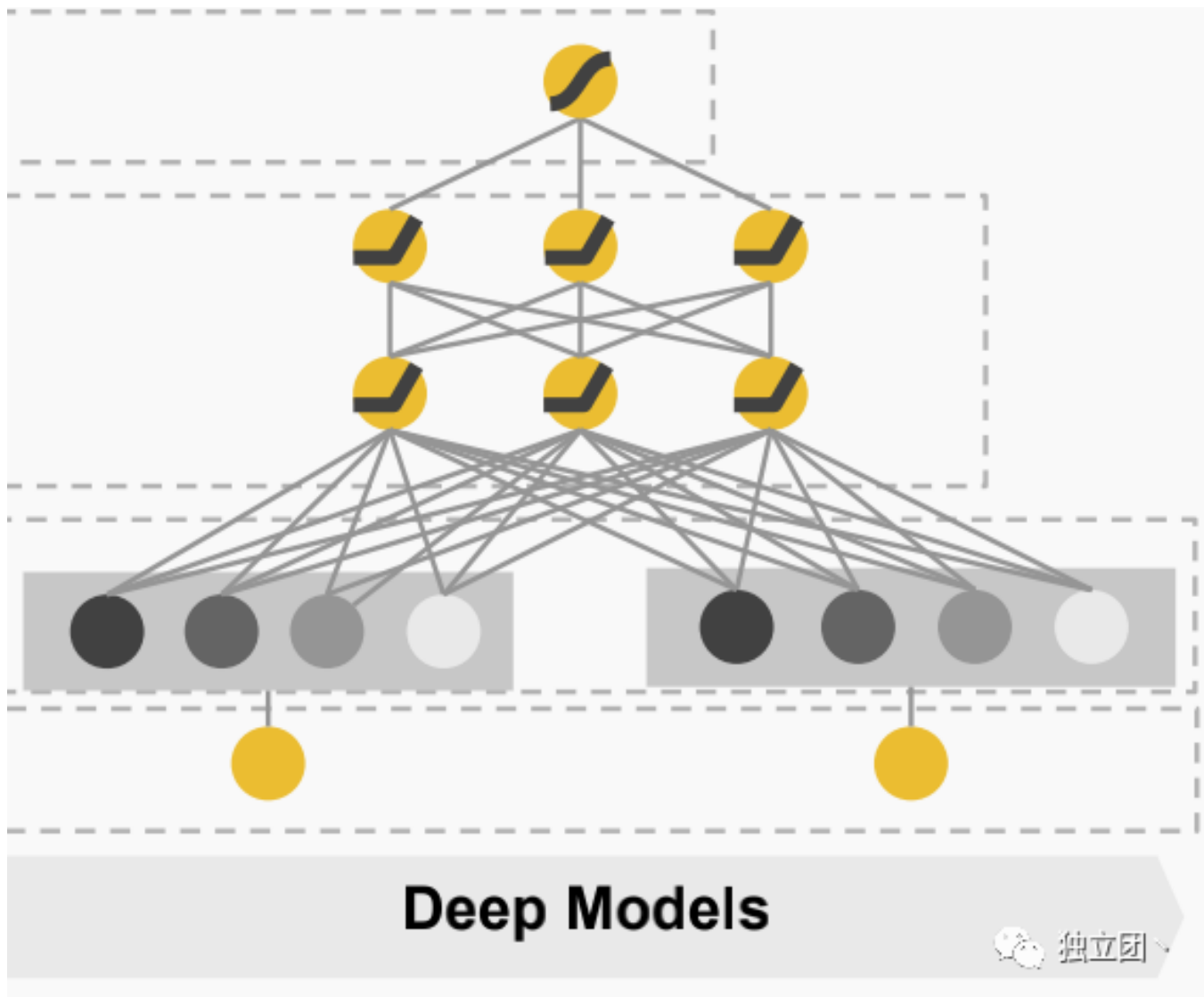
这里值得注意的是，这部分的输入特征包括两部分，一部分是原始的输入特征，另一部分是交叉特征，其中一种方式就是cross-product，具体定义如下：

$$\phi_k(\mathbf{x}) = \prod_{i=1}^d x_i^{c_{ki}} \quad c_{ki} \in \{0, 1\}$$

简单理解cross-product就是在二值特征的情况下对其进行与操作，这种特征交叉的方式给广义线性模型增加的非线性的表述，从而进一步提高了模型的泛化性能。

2、Deep部分

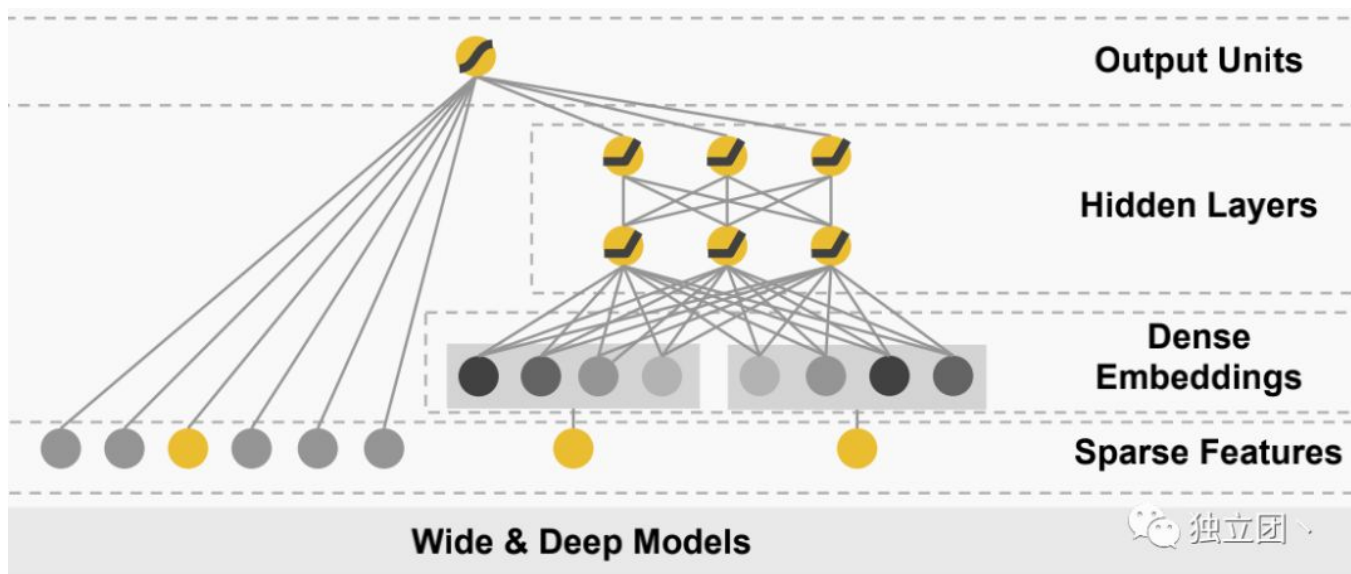
Deep部分简单理解就是Embedding+MLP这种非常普遍的结构，随着近几年深度学习在推荐系统中的大规模应用，Embedding+MLP这种网络结构基本成为了标配，具体模型结构如下所示：



从上图的网络结构可以看出，中间隐藏层的激活函数都是ReLu，最后一层的激活函数是sigmoid

3、联合训练

整体的网络结构如下图所示

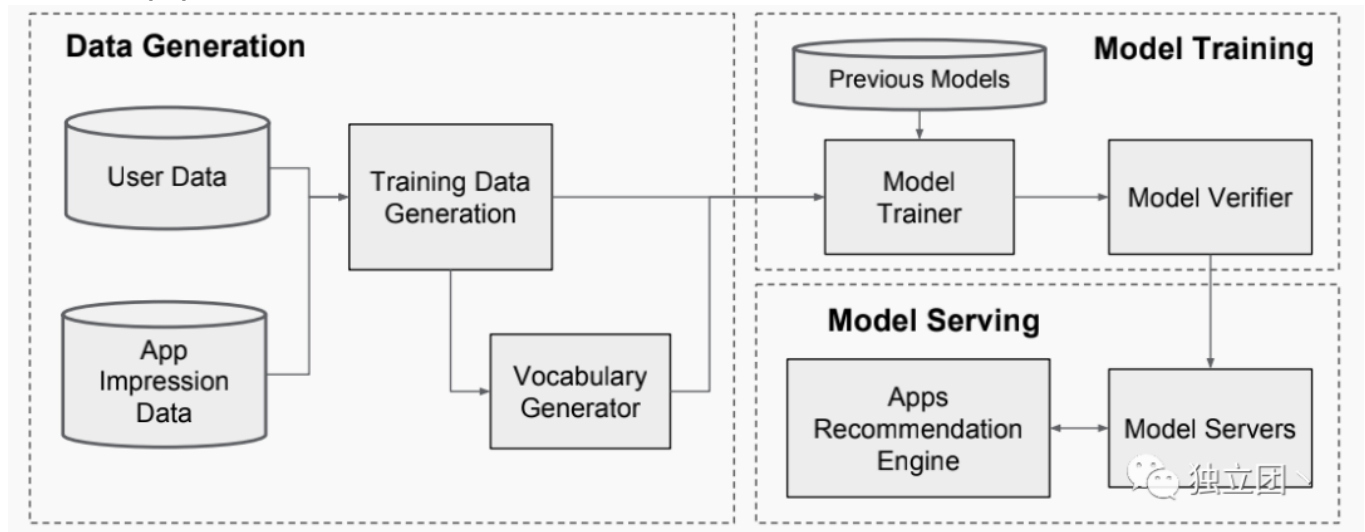


这里网络的训练方式采用的是联合训练，并不是单独的训练两部分网络，然后在预测的时候在一起使用，而是直接在训练阶段就同时训练两部分的网络参数。这种联合训

练的方式有两个好处：一方面联合训练同时训练两部分网络的参数更有利于整体的最优化；另一方面联合训练可以有效降低整个网络的大小。在训练的过程中，wide部分主要采用带L1正则的FTRL算法进行优化,deep部分采用AdaGrad进行优化。

四、系统部署与应用

整个pipeline流程如下图所示：



整个pipeline由三部分组成：数据生成、模型训练、模型部署。

1、数据生成

训练数据主要由两部分组成，一部分是用户相关的日志，另一部分是app的曝光点击数据（因为论文的业务场景是APP推荐），同时还会生成相关的字典，字典主要是将类别的信息以及相关的字符串信息按照一个固定的key进行存储，同时在该部分还会进行诸如归一化、分箱等一些数据预处理的操作。

2、模型训练

当有新的数据生成的时候，都需要重新进行模型训练以及模型的推送过程，这样是十分耗时的，而且也没法保证模型的时效性（当然可以考虑online learning的方式），为了解决这个问题，模型采用的一种解决思路就是加载上一次模型训练得到的embedding网络和线性模型的权重进行增量学习。在推送模型上线之前，需要确保模型指标正常，所以需要进行离线的一些验证。

3、模型部署

模型部署上线之后，在一个用户请求发送给系统之后，只需要进行前向预测过程从而得到最后的预测分数进行后续的排序即可，为满足线上延迟的要求，可以考虑采用分布式运行的方式来进行计算。

五、实验结果

文章对Wide&Deep模型的效果进行了线上的验证，实验结果如下：

Model	Offline AUC	Online Acquisition Gain
Wide (control)	0.726	0%
Deep	0.722	+2.9%
Wide & Deep	0.728	+3.9%

从实验结果可以看到相较于Wide模型，Wide&Deep模型可以带来接近4%的收益提升，同时AUC也有0.2%的提升；同时相较于Deep模型，Wide&Deep模型也可以带来接近1%的收益提升，AUC也有0.6%的提升。

六、结论

从现在来看Wide&Deep仍然是一种十分有效的并且可以实际进行线上部署的CTR预估模型，也有很多公司的Base模型采用的是Wide&Deep模型，可以说该模型对后来CTR深度模型的发展具有很大的指导意义，所以该论文是CTR预估模型领域非常经典的论文，推荐大家有时间能够仔细阅读几遍。

论文地址：<https://arxiv.org/abs/1606.07792>
知乎专栏：<https://zhuanlan.zhihu.com/HIT-LX>

[阅读原文](#)