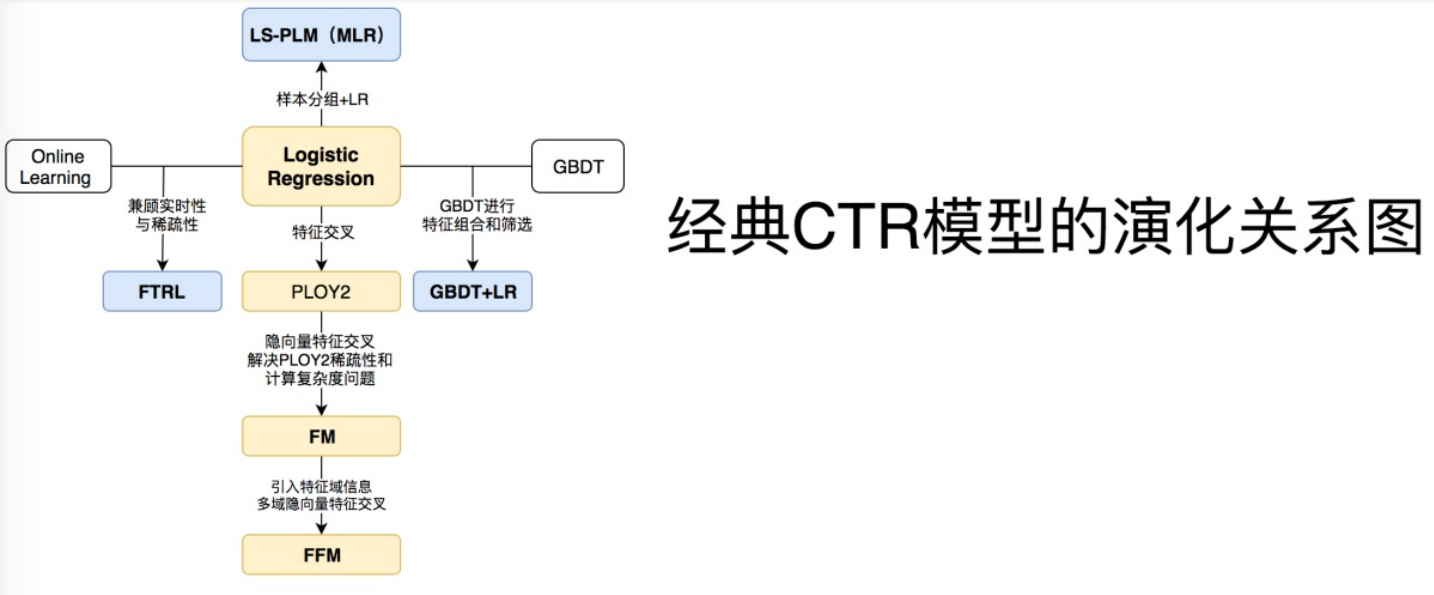


前深度学习时代CTR预估模型的演化之路

王喆
机器学习、深度学习（Deep Learning） 话题的优秀回答者

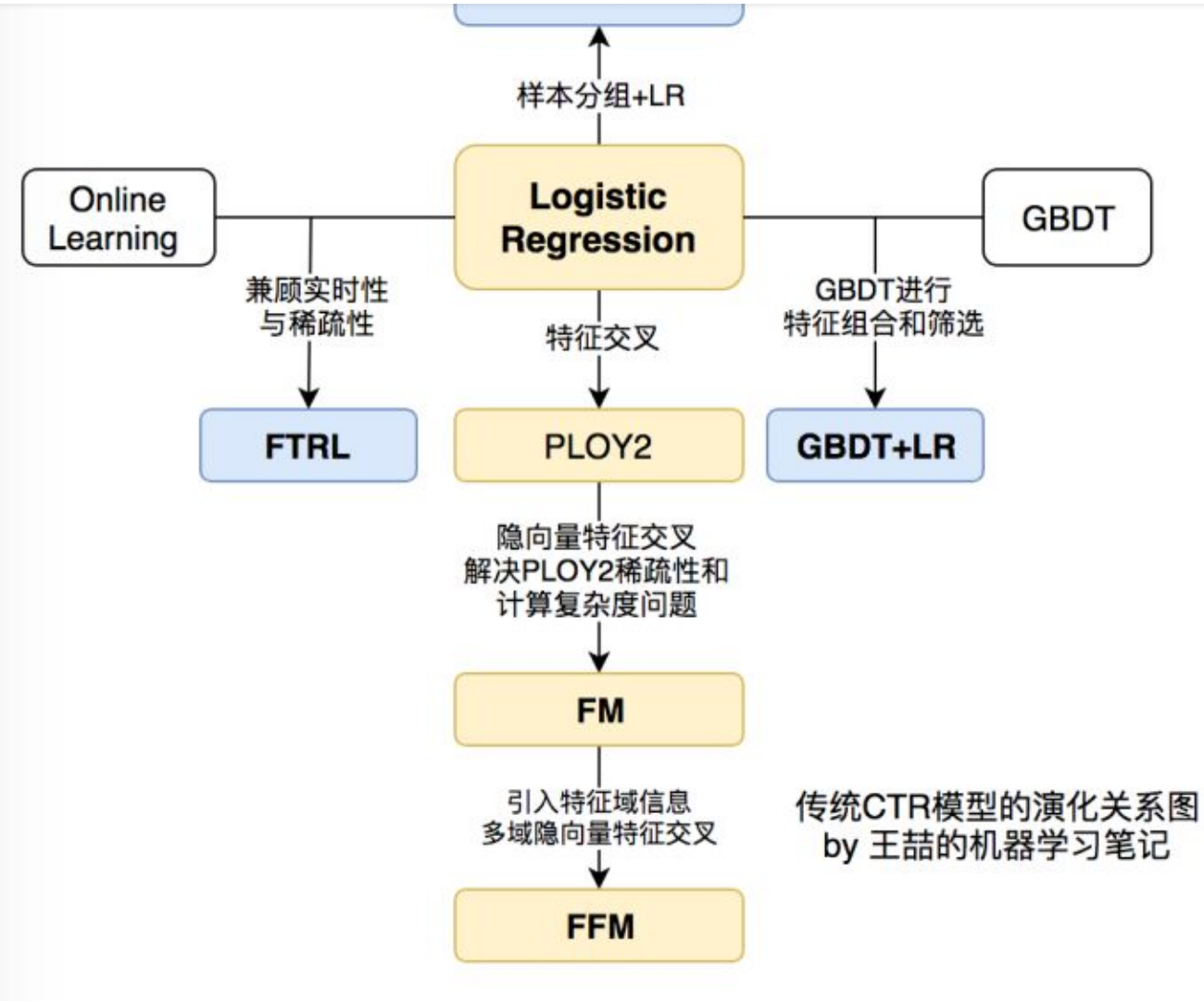
这里是**「王喆的机器学习笔记」**的第十一篇文章，有一段时间没有更新文章了，因为受InfoQ的约稿，梳理了一下子从传统机器学习时代到深度学习时代**所有经典CTR模型的演化关系和模型特点**。我们这篇文章先从前深度学习时代开始，帮大家梳理传统CTR模型的知识体系。

在互联网永不停歇的增长需求的驱动下，CTR预估模型（以下简称CTR模型）的发展也可谓一日千里，从2010年之前千篇一律的**逻辑回归**（Logistic Regression，LR），进化到**因子分解机**（Factorization Machine，FM）、**梯度提升树**（Gradient Boosting Decision Tree，GBDT），再到2015年之后深度学习的百花齐放，各种模型架构层出不穷。

我想所有从业者谈起深度学习CTR预估模型都有一种莫名的兴奋，但在这之前，认真的回顾前深度学习时代的CTR模型仍是非常必要的。原因有两点：

1. 即使是深度学习空前流行的今天，**LR、FM等传统CTR模型仍然凭借其可解释性强、轻量级的训练部署要求、便于在线学习等不可替代的优势，拥有大量适用的应用场景**。模型的应用不分新旧贵贱，熟悉每种模型的优缺点，能够灵活运用和改进不同的算法模型是算法工程师的基本要求。
2. **传统CTR模型是深度学习CTR模型的基础**。深度神经网络（Deep Nerual Network，DNN）从一个神经元生发而来，而LR正是单一神经元的经典结构；此外，影响力很大的FNN，DeepFM，NFM等深度学习模型更是与传统的FM模型有着千丝万缕的联系；更不要说各种梯度下降方法的一脉相承。所以说传统CTR模型是深度学习模型的地基和入口。

下面，我们用传统CTR模型演化的关系图来正式开始技术部分的内容。



传统CTR模型的演化关系图

看到上面的关系图，有经验的同学可能已经对各模型的细节和特点如数家珍了。中间位置的LR模型向四个方向的延伸分别代表了传统CTR模型演化的四个方向。

1. 向下为了解决特征交叉的问题，演化出**PLOY2，FM，FFM**等模型；
2. 向右为了使用模型化、自动化的手段解决之前特征工程的难题，Facebook将LR与GBDT进行结合，提出了**GBDT+LR**组合模型；
3. 向左Google从online learning的角度解决模型时效性的问题，提出了**FTRL**；
4. 向上阿里基于样本分组的思路增加模型的非线性，提出了**LS-PLM（MLR）**模型。

一、LR——CTR模型的核心和基础

位于正中央的是当之无愧的Logistic Regression。仍记得2012年我刚进入计算广告这个行业的时候，各大中小公司的主流CTR模型无一例外全都是LR模型。LR模型的流行是有三方面原因的，一是数学形式和含义上的支撑；二是人类的直觉和可解释性的原因；三是工程化的需要。

1. 逻辑回归的数学基础

典的掷偏心硬币问题，因此CTR模型的因变量显然应该服从伯努利分布。所以采用LR作为CTR 模型是符合“点击”这一事件的物理意义的。

与之相比较，线性回归（Linear Regression）作为广义线性模型的另一个特例，其假设是因变量y服从高斯分布，这明显不是点击这类二分类问题的数学假设。

在了解LR的数学理论基础后，其数学形式就不再是空中楼阁了，具体的形式如下：

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

其中x是输入向量，θ 是我们要学习的参数向量。结合CTR模型的问题来说，x就是输入的特征向量，h(x)就是我们最终希望得到的点击率。

2. 人类的直觉和可解释性

直观来讲，LR模型目标函数的形式就是各特征的加权和，再施以sigmoid函数。忽略其数学基础（虽然这是其模型成立的本质支撑），仅靠人类的直觉认知也可以一定程度上得出使用LR作为CTR模型的合理性。

使用各特征的加权和是为了综合不同特征对CTR的影响，而由于不同特征的重要程度不一样，所以为不同特征指定不同的权重来代表不同特征的重要程度。最后要套上sigmoid函数，正是希望其值能够映射到0-1之间，使其符合CTR的物理意义。

LR如此符合人类的直觉认知显然有其他的好处，就是模型具有极强的可解释性，算法工程师们可以轻易的解释哪些特征比较重要，在CTR模型的预测有偏差的时候，也可以轻易找到哪些因素影响了最后的结果，如果你有跟运营、产品一起工作的经验的话，更会知道可解释性强是一个模型多么优秀的“品质”。

3.工程化的需要

在互联网公司每天动辄TB级别的数据面前，模型的训练开销就异常重要了。在GPU尚未流行开来的2012年之前，LR模型也凭借其易于并行化、模型简单、训练开销小等特点占据着工程领域的主

知乎 首发于 王喆的机器学习笔记

二、POLY2——特征交叉的开始

但LR的表达能力毕竟是非常初级的。由于LR仅使用单一特征，无法利用高维信息，在“辛普森悖论”现象的存在下，只用单一特征进行判断，甚至会得出错误的结论。

针对这个问题，当时的算法工程师们经常采用手动组合特征，再通过各种分析手段筛选特征的方法。但这个方法无疑是残忍的，完全不符合“**懒惰是程序员的美德**”这一金科玉律。更遗憾的是，人类的经验往往有局限性，程序员的时间和精力也无法支撑其找到最优的特征组合。因此采用PLOY2模型进行特征的“暴力”组合成为了可行的选择。

$$\phi_{\mathrm{Poly2}}(\boldsymbol{w},\boldsymbol{x}) = \sum_{j_1=1}^{n}\sum_{j_2=j_1+1}^{n} w_{h(j_1,j_2)}x_{j_1}x_{j_2}$$

在上面POLY2二阶部分的目标函数中（上式省略一阶部分和sigmoid函数的部分），我们可以看到POLY2对所有特征进行了两两交叉，并对所有的特征组合赋予了权重 $w_{h(j_1,j_2)}$ 。POLY2通过暴力组合特征的方式一定程度上解决了特征组合的问题。并且由于本质上仍是线性模型，其训练方法与LR并无区别，便于工程上的兼容。

但POLY2这一模型同时存在着两个巨大的缺陷：

1. 由于在处理互联网数据时，经常采用one-hot的方法处理id类数据，致使特征向量极度稀疏，POLY2进行无选择的特征交叉使原本就非常稀疏的特征向量更加稀疏，使得大部分交叉特征的权重缺乏有效的数据进行训练，无法收敛。
2. 权重参数的数量由n直接上升到n^2，极大增加了训练复杂度。

三、FM——隐向量特征交叉

为了解决POLY2模型的缺陷，2010年德国康斯坦茨大学的Steffen Rendle提出了FM（Factorization Machine）。

$$\phi_{\mathrm{FM}}(\boldsymbol{w},\boldsymbol{x}) = \sum_{j_1=1}^{n}\sum_{j_2=j_1+1}^{n} (\boldsymbol{w}_{j_1}\cdot\boldsymbol{w}_{j_2})x_{j_1}x_{j_2}$$

vector），在特征交叉时，使用两个特征隐向量的内积作为交叉特征的权重。

通过引入特征隐向量的方式，直接把原先n^2级别的权重数量减低到了n*k（k为隐向量维度，n>>k）。在训练过程中，又可以通过转换目标函数形式的方法，使FM的训练复杂度进一步降低到n*k级别。相比POLY2极大降低训练开销。

隐向量的引入还使得FM比POLY2能够更好的解决数据稀疏性的问题。举例来说，我们有两个特征，分别是channel和brand，一个训练样本的feature组合是(ESPN, Adidas)，在POLY2中，只有当ESPN和Adidas同时出现在一个训练样本中时，模型才能学到这个组合特征对应的权重。而在FM中，ESPN的隐向量也可以通过(ESPN, Gucci)这个样本学到，Adidas的隐向量也可以通过(NBC, Adidas)学到，这大大降低了模型对于数据稀疏性的要求。甚至对于一个从未出现过的特征组合(NBC, Gucci)，由于模型之前已经分别学习过NBC和Gucci的隐向量，FM也具备了计算该特征组合权重的能力，这是POLY2无法实现的。也许FM相比POLY2丢失了某些信息的记忆能力，但是泛化能力大大提高，这对于互联网的数据特点是非常重要的。

工程方面，FM同样可以用梯度下降进行学习的特点使其不失实时性和灵活性。相比之后深度学习模型复杂的网络结构，FM比较容易实现的inference过程也使其没有serving的难题。因此FM在2012-2014年前后逐渐成为业界CTR模型的重要选择。

四、FFM——引入特征域概念

2015年，基于FM提出的FFM（Field-aware Factorization Machine，简称FFM）在多项CTR预估大赛中一举夺魁，并随后被Criteo、美团等公司深度应用在CTR预估，推荐系统领域。相比FM模型，FFM模型主要引入了Field-aware这一概念，使模型的表达能力更强。

$$\phi_{\mathrm{FFM}}(\boldsymbol{w},\boldsymbol{x}) = \sum_{j_1=1}^{n}\sum_{j_2=j_1+1}^{n}(\boldsymbol{w}_{j_1,f_2}\cdot\boldsymbol{w}_{j_2,f_1})x_{j_1}x_{j_2}$$

上式是FFM的目标函数的二阶部分。其与FM目标函数的区别就在于隐向量由原来的 $\boldsymbol{w}_{j_1}$ 变成了 $\boldsymbol{w}_{j_1,f_2}$ ，这就意味着每个特征对应的不是一个隐向量，而是对应着不同域的一组隐向量，当 $\boldsymbol{w}_{j_1}$ 特征与 $\boldsymbol{w}_{j_2}$ 特征进行交叉时， $\boldsymbol{x}_{j_1}$ 特征会从一组隐向量中挑出与特征 $\boldsymbol{x}_{j_2}$ 的域f2对应的隐向量 $\boldsymbol{w}_{j_1,f_2}$ 进行交叉。同理特征 $\boldsymbol{x}_{j_2}$ 也会用与 $\boldsymbol{x}_{j_1}$ 的域f1对应的隐向量进行交叉。

这里再次强调一下，上面所说的“域”就代表着特征域，域内的特征一般会采用one-hot编码形成one-hot特征向量。

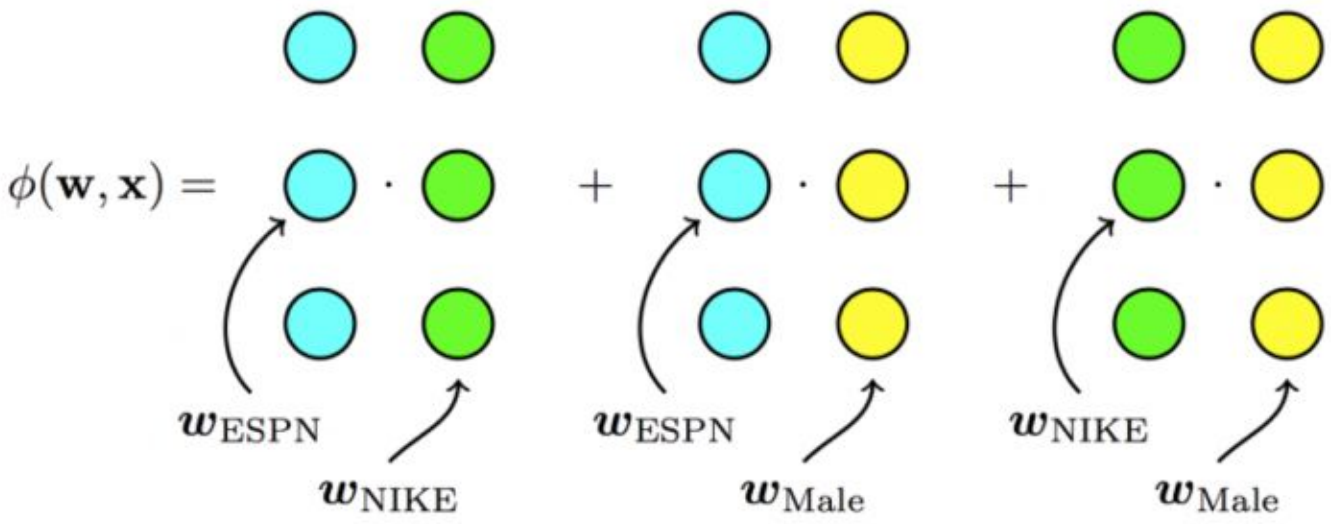
FFM模型学习每个特征在f个域上的k维隐向量，交叉特征的权重由特征在对方特征域上的隐向积得到，权重数量共n*k*f个。在训练方面，由于FFM的二次项并不能够像FM那样简化，因此其复

相比FM，FFM由于引入了field这一概念，为模型引入了更多有价值信息，使模型表达能力更强，但与此同时，FFM的计算复杂度上升到kn^2，远远大于FM的k*n。

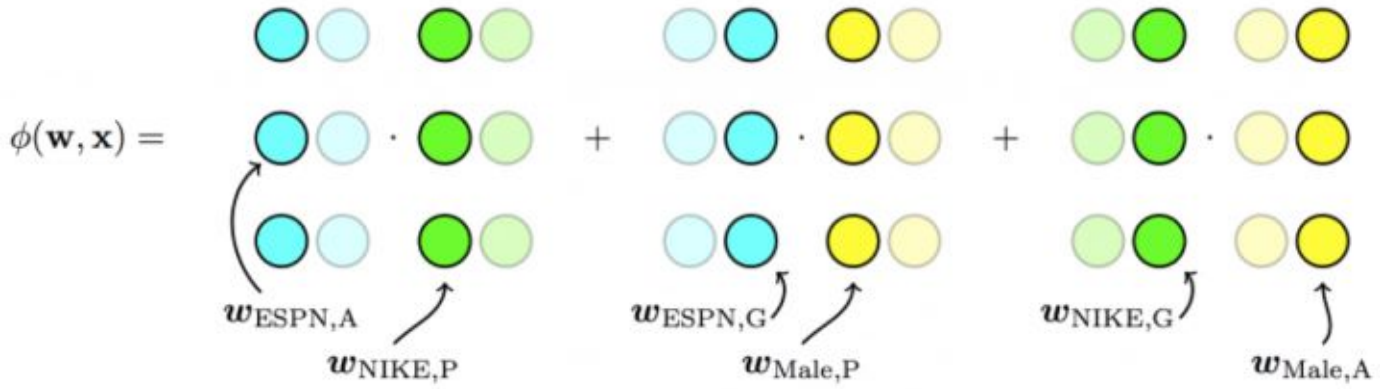
CTR模型特征交叉方向的演化

以上模型实际上是CTR模型朝着特征交叉的方向演化的过程，我们再用图示方法回顾一下从POLY2到FM，再到FFM进行特征交叉方法的不同。

POLY2模型直接学习每个交叉特征的权重，权重数量共n^2个。



FM模型学习每个特征的k维隐向量，交叉特征由相应特征隐向量的内积得到，权重数量共n*k个。

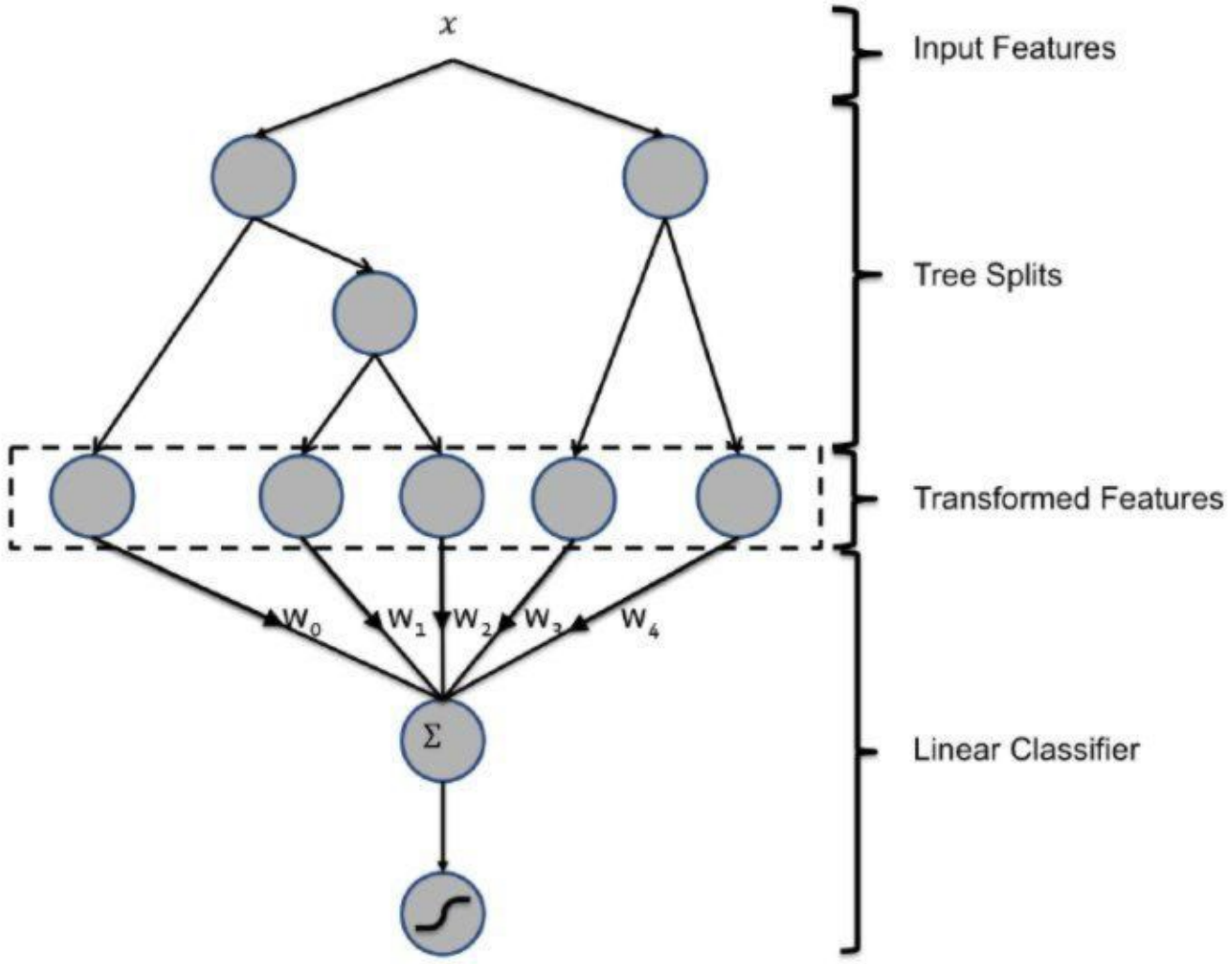


知乎 首发于 王喆的机器学习笔记

五、GBDT+LR——特征工程模型化的开端

FFM模型采用引入特征域的方式增强了模型的表达能力，但无论如何，FFM只能够做二阶的特征交叉，如果要继续提高特征交叉的维度，不可避免的会发生组合爆炸和计算复杂度过高的情况。那么有没有其他的方法可以有效的处理高维特征组合和筛选的问题？2014年，Facebook提出了基于GBDT+LR组合模型的解决方案。

简而言之，Facebook提出了一种利用GBDT自动进行特征筛选和组合，进而生成新的离散特征向量，再把该特征向量当作LR模型输入，预估CTR的模型结构。

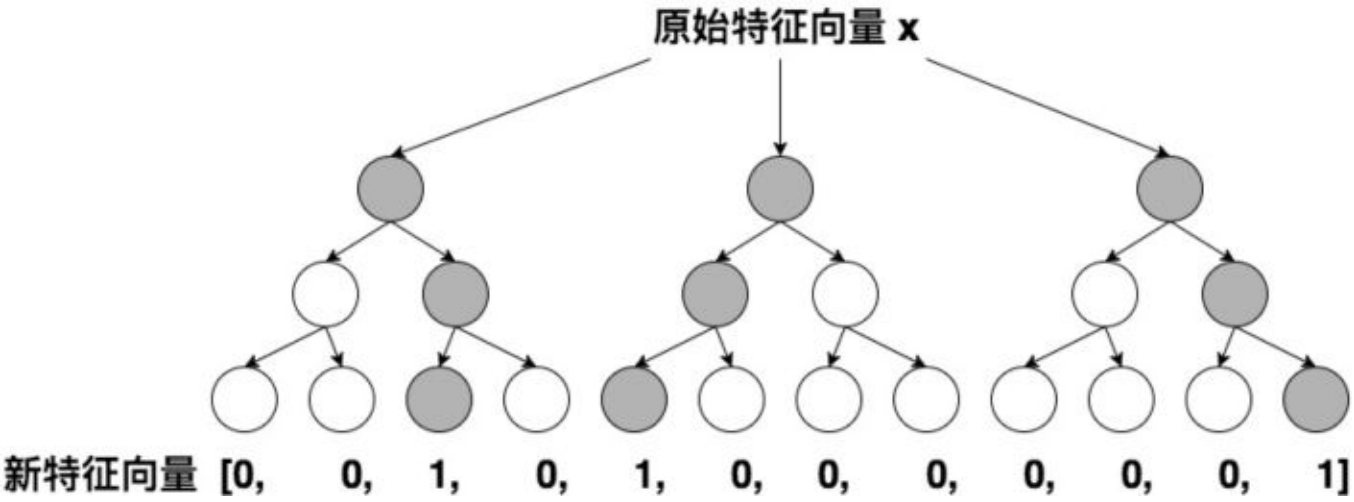


GBDT+LR的模型结构

需要强调的是，用GBDT构建特征工程，和利用LR预估CTR两步是独立训练的。所以自然不存在如何将LR的梯度回传到GBDT这类复杂的问题，而利用LR预估CTR的过程前面已经有所介绍，在此不再赘述，下面着重讲解如何利用GBDT构建新的特征向量。

特征选择的过程，而多层节点的结构自然进行了有效的特征组合，也就非常高效的解决了过去非常棘手的特征选择和特征组合的问题。

利用训练集训练好GBDT模型之后，就可以利用该模型完成从原始特征向量到新的离散型特征向量的转化。具体过程是这样的，一个训练样本在输入GBDT的某一子树后，会根据每个节点的规则最终落入某一叶子节点，那么我们把该叶子节点置为1，其他叶子节点置为0，所有叶子节点组成的向量即形成了该棵树的特征向量，把GBDT所有子树的特征向量连接起来，即形成了后续LR输入的特征向量。



GBDT生成特征向量的过程

举例来说，如上图所示，GBDT由三颗子树构成，每个子树有4个叶子节点，一个训练样本进来后，先后落入“子树1”的第3个叶节点中，那么特征向量就是[0,0,1,0]，“子树2”的第1个叶节点，特征向量为[1,0,0,0]，“子树3”的第4个叶节点，特征向量为[0,0,0,1]，最后连接所有特征向量，形成最终的特征向量[0,0,1,0,1,0,0,0,0,0,0,1]。

由于决策树的结构特点，事实上，决策树的深度就决定了特征交叉的维度。如果决策树的深度为4，通过三次节点分裂，最终的叶节点实际上是进行了3阶特征组合后的结果，如此强的特征组合能力显然是FM系的模型不具备的。但由于GBDT容易产生过拟合，以及GBDT这种特征转换方式实际上丢失了大量特征的数值信息，因此我们不能简单说GBDT由于特征交叉的能力更强，效果就比FFM好，在模型的选择和调试上，永远都是多种因素综合作用的结果。

GBDT+LR比FM重要的意义在于，**它大大推进了特征工程模型化这一重要趋势**，某种意义上来说，之后深度学习的各类网络结构，以及embedding技术的应用，都是这一趋势的延续。

在之前所有的模型演化过程中，实际上是从特征工程这一角度来推演的。接下来，我们从时效性这个角度出发，看一看模型的更新频率是如何影响模型效果的。

模型实效性实验

在模型更新这个问题上，我们的直觉是模型的训练时间和serving时间之间的间隔越短，模型的效果越好，为了证明这一点，facebook的工程师还是做了一组实效性的实验（如上图），在结束模型的训练之后，观察了其后6天的模型loss（这里采用normalized entropy作为loss）。可以看出，模型的loss在第0天之后就有所上升，特别是第2天过后显著上升。因此daily update的模型相比weekly update的模型效果肯定是有大幅提升的。

如果说日更新的模型比周更新的模型的效果提升显著，我们有没有方法实时引入模型的效果反馈数据，做到模型的实时更新从而进一步提升CTR模型的效果呢？Google 2013年应用的FTRL给了我们答案。

六、FTRL——天下武功，唯快不破

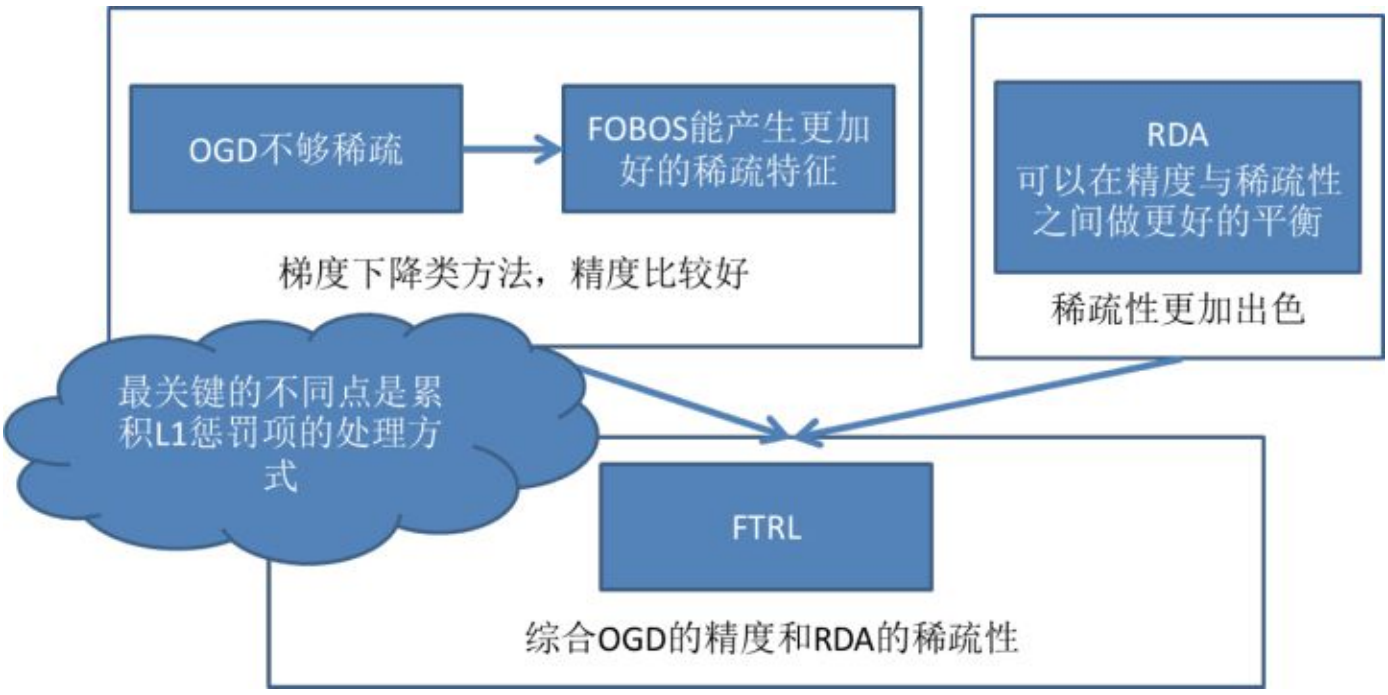
FTRL的全称是Follow-the-regularized-Leader，是一种在线实时训练模型的方法，Google在2010年提出了FTRL的思路，2013年实现了FTRL的工程化，之后快速成为online learning的主流方法。与模型演化图中的其他模型不同，FTRL本质上是模型的训练方法。虽然Google的工程化方案是针对LR模型的，但理论上FTRL可以应用在FM，NN等任何通过梯度下降训练的模型上。

为了更清楚的认识FTRL，这里对梯度下降方法做一个简要的介绍，从训练样本的规模角度来说，梯度下降可以分为：batch，mini-batch，SGD（随机梯度下降）三种，batch方法每次都使用全量训练样本计算本次迭代的梯度方向，mini-batch使用一小部分样本进行迭代，而SGD每次只利用

但SGD对于互利网广告和推荐的场景来说，有比较大的缺陷，就是难以产生稀疏解。为什么稀疏解对于CTR模型如此重要呢？

之前我们已经多次强调，由于one hot等id类特征处理方法导致广告和推荐场景下的样本特征向量极度稀疏，维度极高，动辄达到百万、千万量级。为了不割裂特征选择和模型训练两个步骤，如果能够在保证精度的前提下尽可能多的让模型的参数权重为0，那么我们就可以自动过滤掉这些权重为0的特征，生成一个“轻量级”的模型。“轻量级”的模型不仅会使样本部署的成本大大降低，而且可以极大降低模型inference的计算延迟。这就是模型稀疏性的重要之处。

而SGD由于每次迭代只选取一个样本，梯度下降的方向虽然总体朝向全局最优解，但微观上的运动的过程呈现布朗运动的形式，这就导致SGD会使几乎所有特征的权重非零。即使加入L1正则化项，由于CPU浮点运算的结果很难精确的得到0的结果，也不会完全解决SGD稀疏性差的问题。就是在这样的前提下，FTRL几乎完美地解决了模型精度和模型稀疏性兼顾的训练问题。



FTRL的发展过程

但FTRL的提出也并不是一蹴而就的。如上图所示，FTRL的提出经历了下面几个关键的过程：

1. **从最近简单的SGD到OGD**（online gradient descent），OGD通过引入L1正则化简单解决稀疏性问题；
2. **从OGD到截断梯度法**，通过暴力截断小数值梯度的方法保证模型的稀疏性，但损失了梯度下降的效率和精度；
3. **FOBOS**（Forward-Backward Splitting），google和伯克利对OGD做进一步改进，09年提出了保证精度并兼顾稀疏性的FOBOS方法；
4. **RDA**：微软抛弃了梯度下降这条路，独辟蹊径提出了正则对偶平均来进行online learning的

一，提出并应用FTRL，使FOBOS和RDA均成为了FTRL在特定条件下的特殊形式。
FTRL的算法细节对于初学者来说仍然是晦涩的，建议非专业的同学仅了解其特点和应用场景即可。对算法的数学形式和实现细节感兴趣的同学，我强烈推荐微博 冯扬 写的“在线最优化求解”一文，希望能够帮助大家进一步熟悉FTRL的技术细节。

七、LS-PLM——阿里曾经的主流CTR模型

下面我们从样本pattern本身来入手，介绍阿里的的LS-PLM（Large Scale Piece-wise Linear Model），它的另一个更广为人知的名字是MLR（Mixed Logistic Regression）。MLR模型虽然在2017年才公之于众，但其早在2012年就是阿里主流的CTR模型，并且在深度学习模型提出之前长时间应用于阿里的各类广告场景。

本质上，MLR可以看做是对LR的自然推广，它**在LR的基础上采用分而治之的思路**，先对样本进行分片，再在样本分片中应用LR进行CTR预估。在LR的基础上加入聚类的思想，其动机其实来源于对计算广告领域样本特点的观察 。

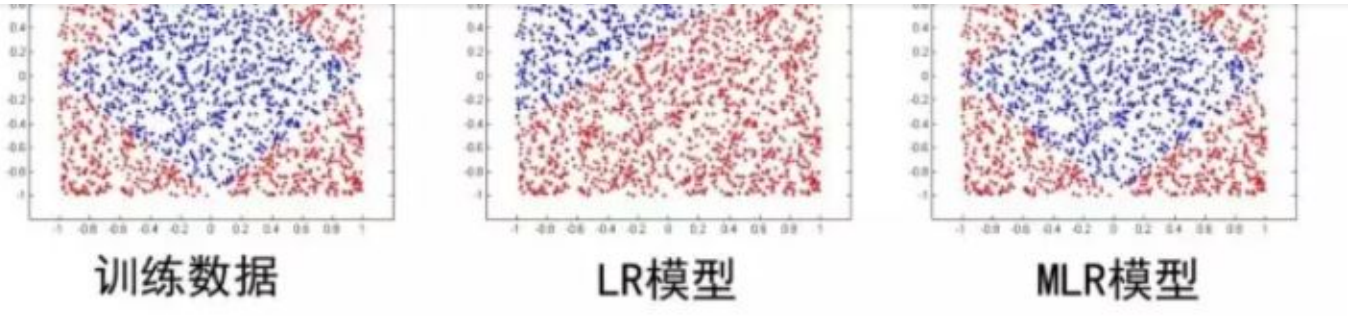
举例来说，如果CTR模型要预估的是女性受众点击女装广告的CTR，显然我们并不希望把男性用户点击数码类产品的样本数据也考虑进来，因为这样的样本不仅对于女性购买女装这样的广告场景毫无相关性，甚至会在模型训练过程中扰乱相关特征的权重。为了让CTR模型对不同用户群体，不用用户场景更有针对性，其实理想的方法是先对全量样本进行聚类，再对每个分类施以LR模型进行CTR预估。MLR的实现思路就是由该动机产生的。

$$f(x) = \sum_{i=1}^{m} \pi_i(x) \cdot \eta_i(x) = \sum_{i=1}^{m} \frac{e^{\mu_i \cdot x}}{\sum_{j=1}^{m} e^{\mu_j \cdot x}} \cdot \frac{1}{1 + e^{-w_i \cdot x}}$$

MLR目标函数的数学形式如上式，首先用聚类函数π对样本进行分类（这里的π采用了softmax函数，对样本进行多分类），再用LR模型计算样本在分片中具体的CTR，然后将二者进行相乘后加和。

其中超参数分片数m可以较好地平衡模型的拟合与推广能力。当m=1时MLR就退化为普通的LR，m越大模型的拟合能力越强，但是模型参数规模随m线性增长，相应所需的训练样本也随之增长。在实践中，阿里给出了m的经验值为12。

下图中MLR模型用4个分片可以完美地拟合出数据中的菱形分类面。



MLR算法适合于工业级的广告、推荐等大规模稀疏数据场景问题。主要是由于表达能力强、稀疏性高等两个优势：

1. **端到端的非线性学习**：从模型端自动挖掘数据中蕴藏的非线性模式，省去了大量的人工特征设计，这使得MLR算法可以端到端地完成训练，在不同场景中的迁移和应用非常轻松。
2. **稀疏性**：MLR在建模时引入了L1和L2,1范数，可以使得最终训练出来的模型具有较高的稀疏度，模型的学习和在线预测性能更好。

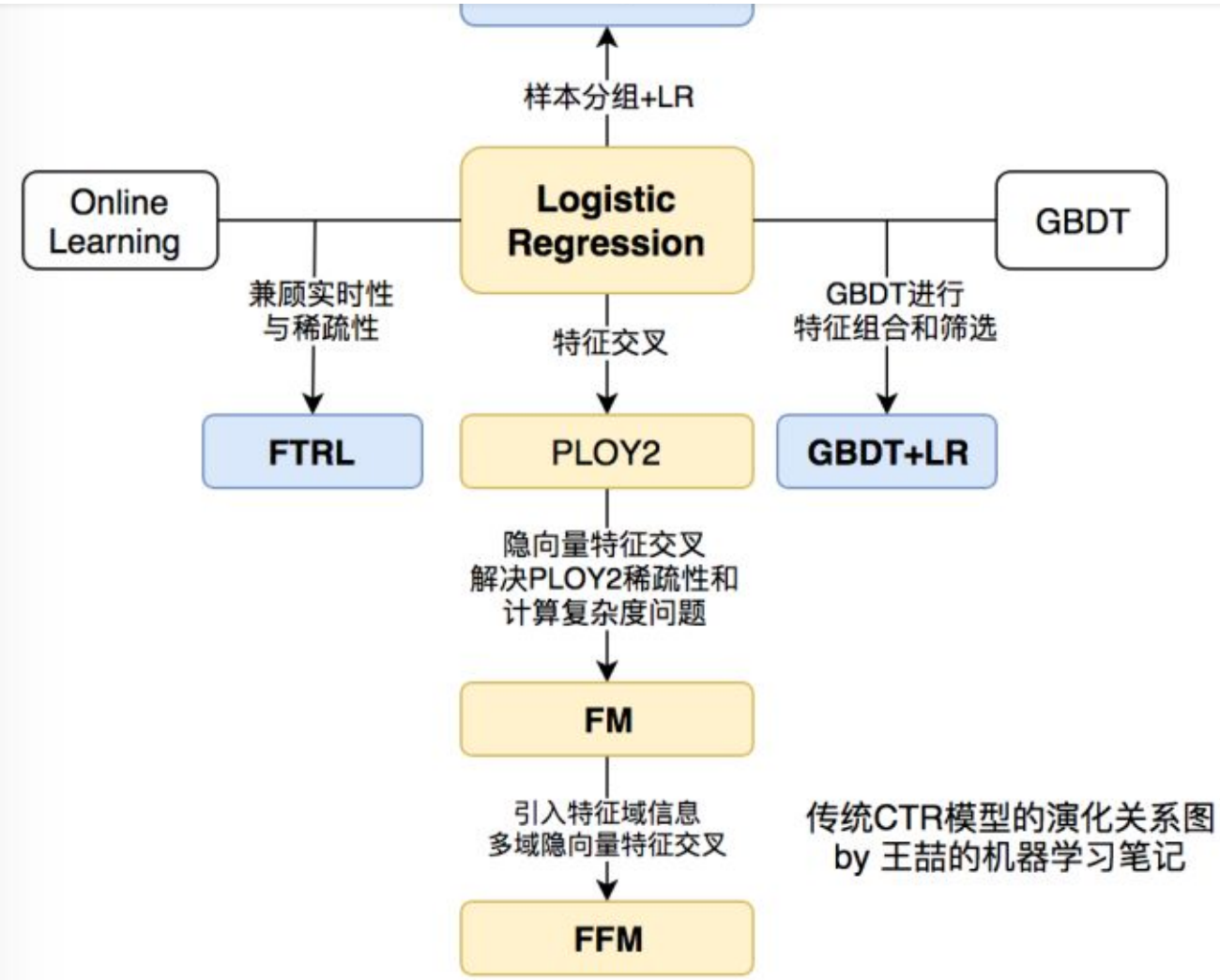
如果我们用深度学习的眼光来看待MLR这个模型，其在结构上已经很接近由输入层、单隐层、输出层组成的神经网络。所以某种意义上说，MLR也在用自己的方式逐渐逼近深度学习的大门了。

深度学习CTR模型的前夜

2010年FM被提出，特征交叉的概念被引入CTR模型；2012年MLR在阿里大规模应用，其结构十分接近三层神经网络；2014年Facebook用GBDT处理特征，揭开了特征工程模型化的篇章。这些概念都将在深度学习CTR模型中继续应用，持续发光。

另一边，Alex Krizhevsky 2012年提出了引爆整个深度学习浪潮的AlexNet，深度学习的大幕正式拉开，其应用逐渐从图像扩展到语音，再到NLP领域，推荐和广告也必然会紧随其后，投入深度学习的大潮之中。

2016年，随着FNN，Deep&Wide，Deep crossing等一大批优秀的CTR模型框架的提出，深度学习CTR模型逐渐席卷了推荐和广告领域，成为新一代CTR模型当之无愧的主流。下一篇文章，我们将继续探讨深度学习CTR模型，从模型演化的角度揭开所有主流深度学习CTR模型之间的关系和每个模型的特点，期待继续与你一同学习讨论。



最后再次强调一下模型的演化图，希望大家能从更高的角度审视整个CTR模型的发展过程。

这里是**「王喆的机器学习笔记」**，因为文章的内容较多，水平有限，希望大家能够踊跃指出文章中的错误，先多谢大家！也欢迎大家关注我的**微信公众号 王喆的机器学习笔记**（**wangzhenotes**），跟踪计算广告、推荐系统等机器学习领域前沿，谢谢！

参考资料：

- [LR] Predicting Clicks - Estimating the Click-Through Rate for New Ads (Microsoft 2007)
- [FFM] Field-aware Factorization Machines for CTR Prediction (Criteo 2016)
- [GBDT+LR] Practical Lessons from Predicting Clicks on Ads at Facebook (Facebook 2014)
- [PS-PLM] Learning Piece-wise Linear Models from Large Scale Data for Ad Click Prediction (Alibaba 2017)
- [FTRL] Ad Click Prediction a View from the Trenches (Google 2013)
- [FM] Fast Context-aware Recommendations with Factorization Machines (UKON 2011)

发布于 2019-04-02