见微知著，你真的搞懂Google的Wide&Deep模型了吗？

原创 王喆的机器学习笔记 王喆的机器学习笔记 5月26日

见微知著，你真的搞懂Google的Wide&Deep模型了吗？

这里是**「王喆的机器学习笔记」**的第三十二篇文章。今天的文章内容来源于一次跟网友的讨论，同行网友的问题是这样的：

为什么在Google的Wide&Deep模型中，要使用带L1正则化项的FTRL作为wide部分的优化方法，而使用AdaGrad作为deep部分的优化方法？

论文原文的描述是这样的：

> In the experiments, we used Follow- the-regularized-leader (FTRL) algorithm with L1 regularization as the optimizer for the wide part of the model, and AdaGrad for the deep part.

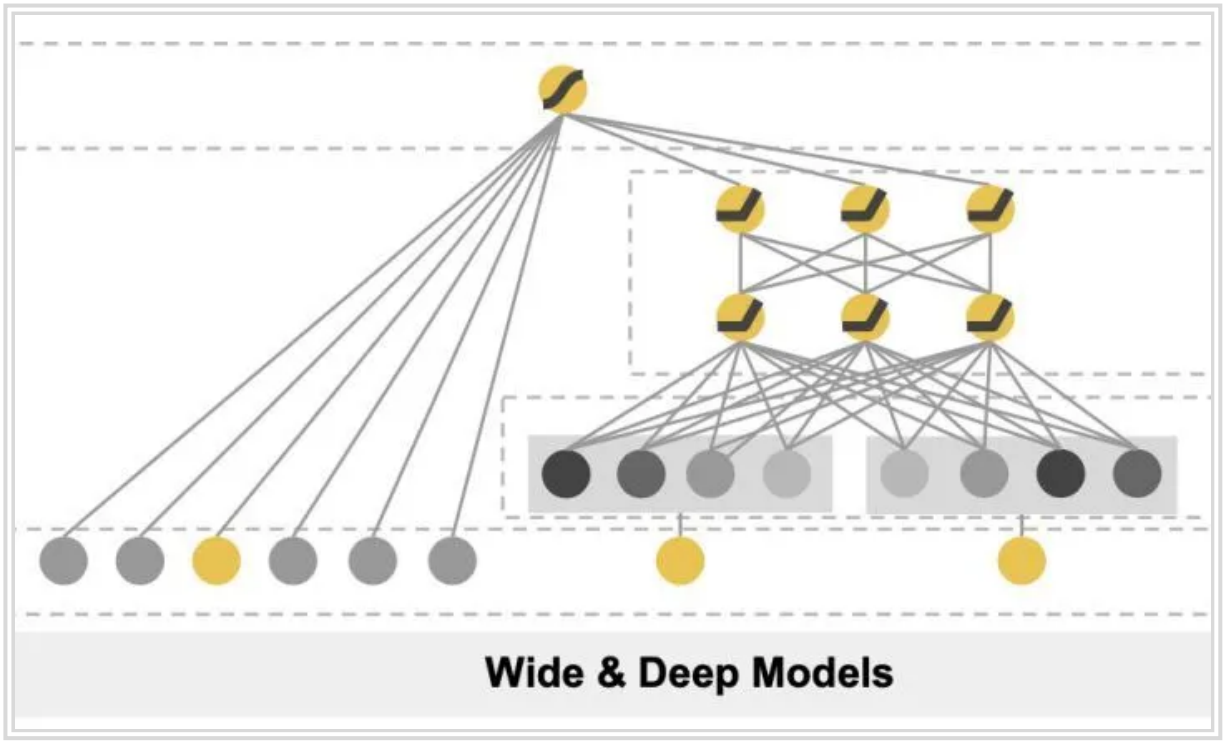
这个问题是一个很有意思的问题，因为原文中一带而过，所以很多同学也没有注意到这一点。但深究起来，这又是一个关键的问题，它涉及到不同训练方法的区别联系，涉及到模型的稀疏性，甚至涉及到特征选择和业务理解。

我们这篇文章就深入到Wide&Deep模型中去，从FTRL和AdaGrad出发，再剖析一次Wide&Deep模型（简称W&D）。

一句话概括W&D

由于W&D被剖析过太多次，也被应用过太多次，所以原理上这里不再赘述，一句话概括：

W&D由浅层（或单层）的Wide部分神经网络和深层的Deep部分多层神经网络组成，输出层采用softmax或logistics regression综合Wide和Deep部分的输出。



Wide&Deep模型示意图

一句话概括此结构的优点：

Wide部分有利于增强模型的“记忆能力”，Deep部分有利于增强模型的“泛化能力”。

相信大家对这些知识点都已经驾轻就熟，那就直接进入这篇文章的主要切入点，为什么Wide部分要用FTRL训练？

为什么Wide部分要用L1 FTRL训练？

这个问题是一个很有意思的问题，可能近几年毕业的同学都不大清楚FTRL是什么了。四五年前FTRL曾风靡全部互联网头部公司，成为线性模型在线训练的主要方法。

彻底解释清楚FTRL并不是一件容易的事情，可能要花上10-20页左右的篇幅，感兴趣的同学可以参考冯扬当时的著名文章“在线最优化求解”。

这里简要介绍一下，你可以把FTRL当作一个稀疏性很好、精度又不错的随机梯度下降方法

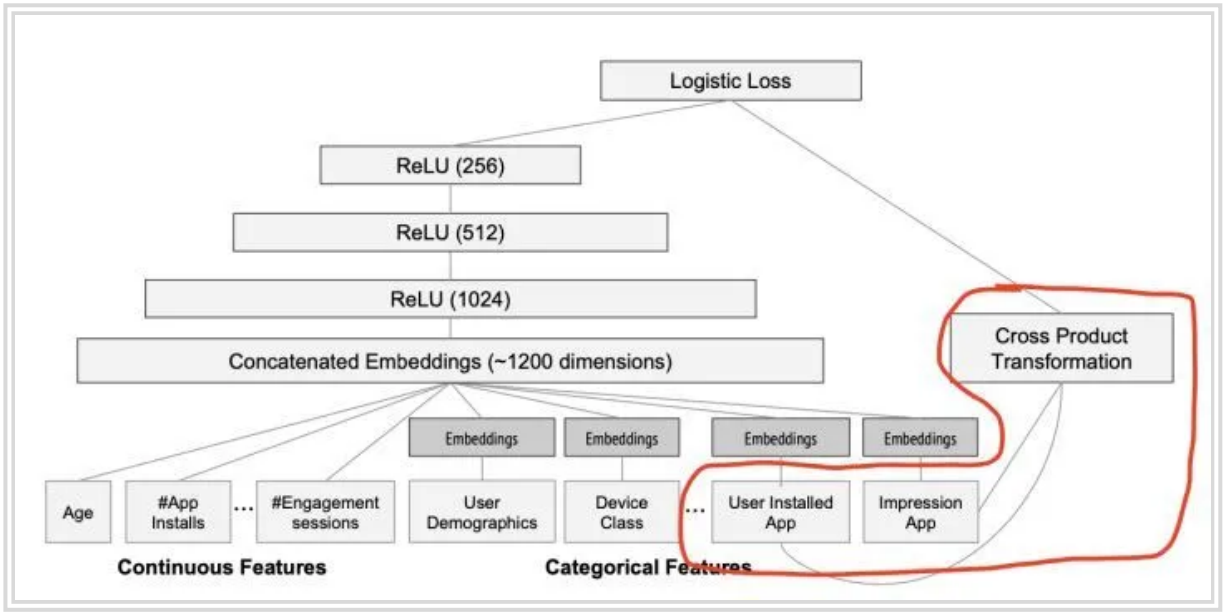
这里简要介绍一下，你可以把FTRL当作一个**确定性很好，精度又不错的随机梯度下降方法。**由于是随机梯度下降，当然可以做到来一个样本就训练一次，进而实现模型的在线更新。所以在四五年前，大部分公司还是线性模型为主的时代，FTRL凭借非常好的在线学习能力成为主流。

说完了FTRL，再说L1正则化，参加过算法岗面试的同学可能都碰到过那个经典面试题“为什么L1正则化比L2正则化更容易产生稀疏解？”。问题的答案现在当然已经是显学了，但这里“**稀疏**”这个性质又冒出来了。**也就是说FTRL with L1非常注重模型的稀疏性。这也就是问题的答案，W&D采用L1 FTRL是想让Wide部分变得更加稀疏。**

再白话一点就是，L1 FTRL会让Wide部分的大部分权重都为0，我们准备特征的时候就不用准备那么多0权重的特征了，这大大压缩了模型权重，也压缩了特征向量的维度。

Wide部分的稀疏性为什么这么关键?

稀疏性不见得一直是一个好东西，它不管怎样都会让模型的精度有一定的损伤。肯定是特征向量维度过高导致“稀疏性”成为了关键的考量。这就涉及到Google Wide部分的特征选取了，到底Google选了什么特征需要这么注重稀疏性。我们回到他的业务场景中来。



Wide部分

大家可以看到红圈内的Wide部分采用了什么特征，它居然采用了两个id类特征的乘积，这两个id类特征是：

User Installed App 和 Impression App

这篇文章是Google的应用商店团队Google Play发表的，我们不难猜测Google的工程师使用这个组合特征的意图，他们是想发现当前曝光app和用户安装app的关联关系，以此来直接影响最终的得分。

但是两个id类特征向量进行组合，在维度爆炸的同时，会让原本已经非常稀疏的multihot特征向量，变得更加稀疏。正因如此，wide部分的权重数量其实是海量的。为了不把数量如此之巨的权重都搬到线上进行model serving，采用FTRL过滤掉哪些稀疏特征无疑是非常好的工程经验。

为什么Deep部分不特别考虑稀疏性的问题?

大家注意观察可以发现Deep部分的输入，要么是Age，#App Installs这些数值类特征，要么是已经降维并稠密化的Embedding向量，工程师们不会也不敢把过度稀疏的特征向量直接输入到Deep网络中。所以Deep部分不存在严重的特征稀疏问题，自然可以使用精度更好，更适用于深度学习训练的AdaGrad去训练。

再说回模型的泛化能力和记忆能力

我想到这应该把文首的问题回答清楚了。最后我想再说回所谓wide部分的“记忆能力”。其实大家可以看到，所谓的“记忆能力”，可以简单理解为发现“直接的”、“暴力的”、“显然的”关联规则的能力。比如该问题中，Google W&D期望在wide部分发现这样的规则：

用户安装了应用A，此时曝光应用B，用户安装的B概率大。

而Deep部分就更黑盒一些，它把能想到的所有特征扔进这个黑盒去做函数的拟合，显然这样的过程会“模糊”一些直接的因果关系，泛化成一些间接的，可能的相关性。

从这个角度来说，所谓“泛化能力”和“记忆能力”就更容易被直观的理解了。

最后，感谢当初网友的提问，**注重细节，见微知著我想永远是一个算法工程师可贵的能力。**

按惯例的讨论问题（欢迎点击阅读原文，跳转到知乎原文参与讨论）：

> **在模型结构日渐复杂的今天，你认为wide部分存在的意义还大吗？在你的模型中还保留这wide部分这种简单暴力的结构，还是已经用更复杂的结构来代替？**
>
> **能把Wide部分和Deep部分分开训练吗？能让FTRL在线学习，而深度部分batch训练吗？**

这里是**「王喆的机器学习笔记」**的第三十二篇文章。

关于W&D的细节讨论亦收录在我的新书「**深度学习推荐系统**」中，这本书系统性地整理、介绍了专栏中所有的重点内容，如果您曾在**「王喆的机器学习笔记」**中受益，欢迎购买。

—END—

扫码关注我们

认为文章有价值的同学，欢迎关注「王喆的机器学习笔记」（wangzhenotes），跟踪计算广告、推荐系统、个性化搜索等机器学习领域前沿。

阅读原文