

DeepFM算法详解-推荐算法中的皇冠

原创 傲海 凡人机器学习 2019-08-15

今天来讲下DeepFM哈，网上有很多人都解析过，所以重复的观点我就不多说了，重点讲些自己的理解

01 概述

DeepFM是一种推荐领域炙手可热的算法，在非常多大家熟识的互联网公司都有落地场景。对于DeepFM最佳的阐述一定是来自于它的原始的论文。

地址：<https://arxiv.org/pdf/1703.04247.pdf>

DeepFM: A Factorization-Machine based Neural Network for CTR Prediction

Huifeng Guo¹, Ruiming Tang², Yunming Ye¹, Zhenguo Li², Xiuqiang He²

¹Shenzhen Graduate School, Harbin Institute of Technology, China

²Noah's Ark Research Lab, Huawei, China

¹huifengguo@yeah.net, yeyunming@hit.edu.cn

²{tangruiming, li.zhenguo, hexiuqiang}@huawei.com

Abstract

Learning sophisticated feature interactions behind user behaviors is critical in maximizing CTR for recommender systems. Despite great progress, existing methods seem to have a strong bias towards low- or high-order interactions, or require expertise feature engineering. In this paper, we show that it is possible to derive an end-to-end learning model that emphasizes both low- and high-order feature interactions. The proposed model, DeepFM, combines the power of factorization machines for recommendation and deep learning for feature learning in a new neural network architecture. Compared to the latest Wide & Deep model from Google, DeepFM has a shared input to its "wide" and "deep" parts, with no need of feature engineering besides raw features. Comprehensive experiments are conducted to demonstrate the effectiveness and efficiency of DeepFM over the existing models for CTR prediction, on both benchmark data and commercial data.

1 Introduction

The prediction of click-through rate (CTR) is critical in recommender system, where the task is to estimate the probability a user will click on a recommended item. In many recommender systems the goal is to maximize the number of clicks, and so the items returned to a user can be ranked by estimated CTR; while in other application scenarios such as online advertising it is also important to improve revenue, and so the ranking strategy can be adjusted as $CTR \times bid$ across all candidates, where "bid" is the benefit the system receives if the item is clicked by a user. In either case, it is clear that the key is in estimating CTR correctly.

It is important for CTR prediction to learn implicit feature interactions behind user click behaviors. By our study in a mainstream apps market, we found that people often download apps for food delivery at meal-time, suggesting that the (order-2) interaction between app category and time-stamp

^{*}This work is done when Huifeng Guo worked as intern at Noah's Ark Research Lab, Huawei.

[†]Corresponding Author.

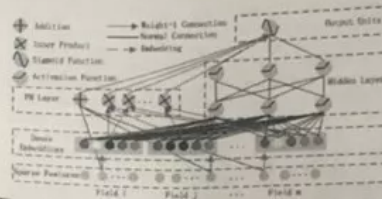


Figure 1: Wide & deep architecture of DeepFM. The wide and deep component share the same input raw feature vector, which enables DeepFM to learn low- and high-order feature interactions simultaneously from the input raw features.

can be used as a signal for CTR. As a second observation, male teenagers like shooting games and RPG games, which means that the (order-3) interaction of app category, user gender and age is another signal for CTR. In general, such interactions of features behind user click behaviors can be highly sophisticated, where both low- and high-order feature interactions should play important roles. According to the insights of the Wide & Deep model [Cheng et al., 2016] from Google, considering low- and high-order feature interactions simultaneously brings additional improvement over the cases of considering either alone.

The key challenge is in effectively modeling feature interactions. Some feature interactions can be easily understood, thus can be designed by experts (like the instances above). However, most other feature interactions are hidden in data and difficult to identify *a priori* (for instance, the classic association rule "diaper and beer" is mined from data, instead of discovering by experts), which can only be captured automatically by machine learning. Even for easy-to-understand interactions, it seems unlikely for experts to model them exhaustively, especially when the number of features is large.

Despite their simplicity, generalized linear models, such as FTRL [McMahan et al., 2013], have shown decent performance in practice. However, a linear model lacks the ability to learn feature interactions, and a common practice is to manually include pairwise feature interactions in its feature vector. Such a method is hard to generalize to model high-order feature interactions or those never or rarely appear in the training data [Rendle, 2010]. Factorization Machines

然后再介绍下推荐算法的本源，推荐其实是一个典型的二分类场景。在推荐算法中需要做的事情无非是把一个商品或者广告推送给一个人，然后预测出来这个人喜欢或者不喜欢。这里最难的不是如何实现分类，而是特征的提取。

而特征的提取又分为低阶特征和高阶特征。举个简单的例子什么叫低阶特征，比如一个人 n 天买了 m 个馒头，我们可以提取一个特质表示这个人每天的馒头消耗量 m/n ，这个特征只考虑了2个原始特征的交叉，所以是一个低阶特征。那高阶特征顾名思义，可以通过一个特征表示非常多原始特征的交叉，这也是深度学习所擅长的。

论文中指出，无论是低阶特征还是高阶特征，都有它自己的价值，都对推荐准确性有着很大的影响。而如何利用好低阶和高阶的特征，就是DeepFM最大的特点。

02 DeepFM与其它算法的比较

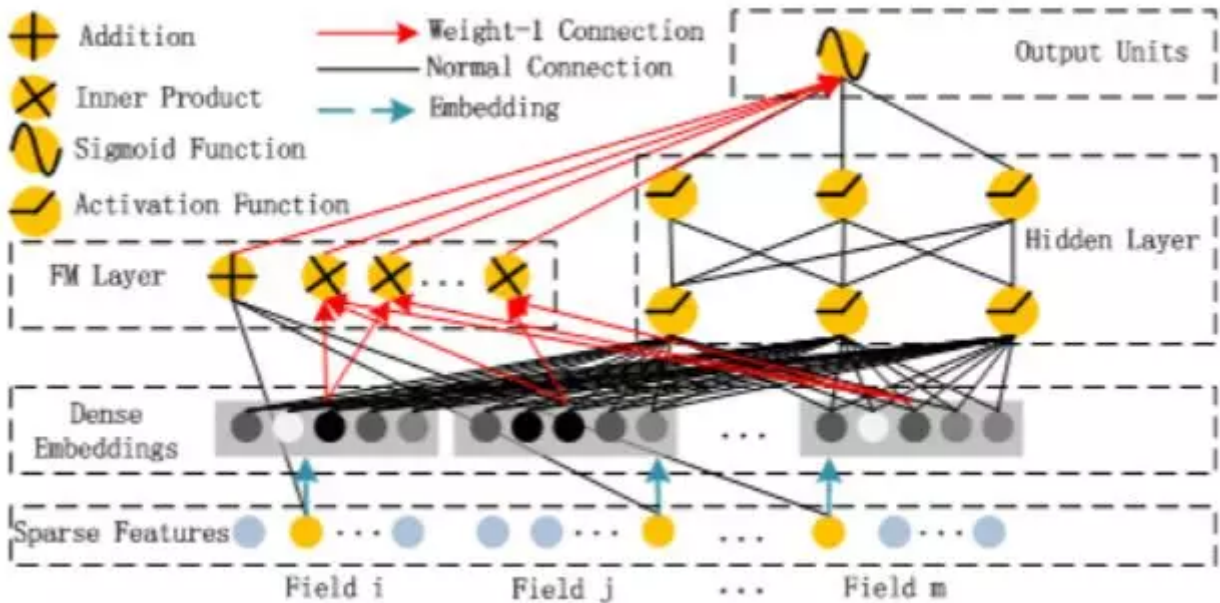
通过第一节，大家了解到推荐其实是特征利用的一种博弈。现在就通过特征利用程度介绍下DeepFM与其它算法的区别：

- * 逻辑回归(LR)：更多考虑线性特征，缺少特征交叉性和高阶特征
 - * DNN：考虑了高阶特征，缺少了对于低阶特征的考虑
 - * CNN：考虑近邻特征的关系。较单一，适合图片分类
 - * RNN：考虑更多的是数据时序性，较单一
 - * FM：考虑更多低阶特征，缺少高阶特征
 - * Wide&Deep：同时考虑了低阶特征和高阶特征，但是低阶特征需要手动交叉生成，对用户不友好
 - * DeepFM：兼顾了低阶和高阶特征，且计算过程中不需要用户干预
- 文中一份基于开源数据的测试结果也很有说服力：

	Company*		Criteo	
	AUC	LogLoss	AUC	LogLoss
LR	0.8640	0.02648	0.7686	0.47762
FM	0.8678	0.02633	0.7892	0.46077
FNN	0.8683	0.02629	0.7963	0.45738
IPNN	0.8664	0.02637	0.7972	0.45323
OPNN	0.8658	0.02641	0.7982	0.45256
PNN*	0.8672	0.02636	0.7987	0.45214
LR & DNN	0.8673	0.02634	0.7981	0.46772
FM & DNN	0.8661	0.02640	0.7850	0.45382
DeepFM	0.8715	0.02618	0.8007	0.45083

03 DeepFM架构介绍

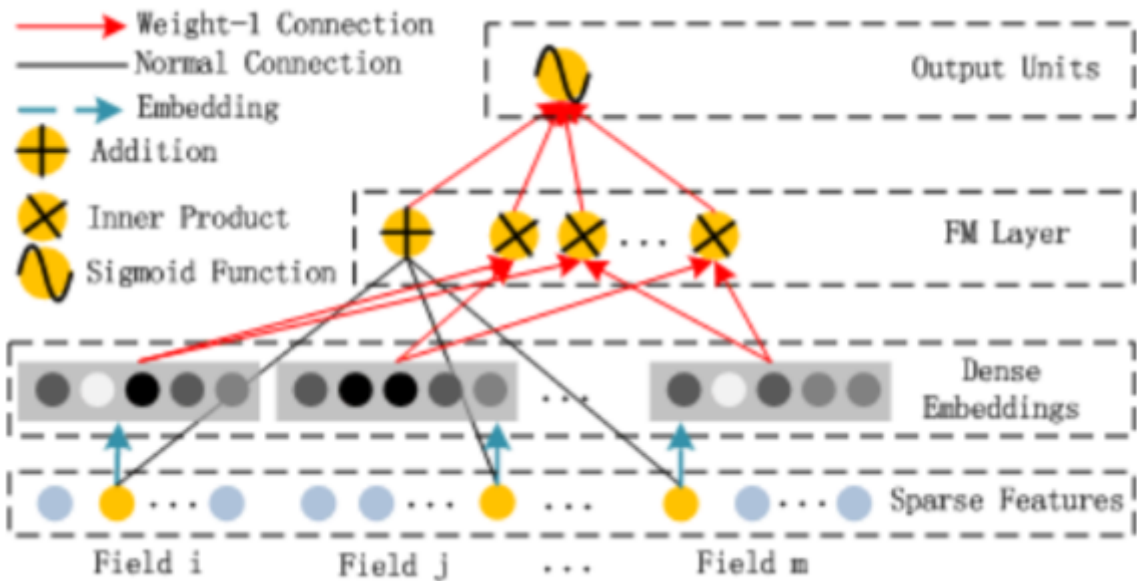
DeepFM这么牛，究竟是如何实现的呢？从架构上可以拆解成FM部分和DNN部分。



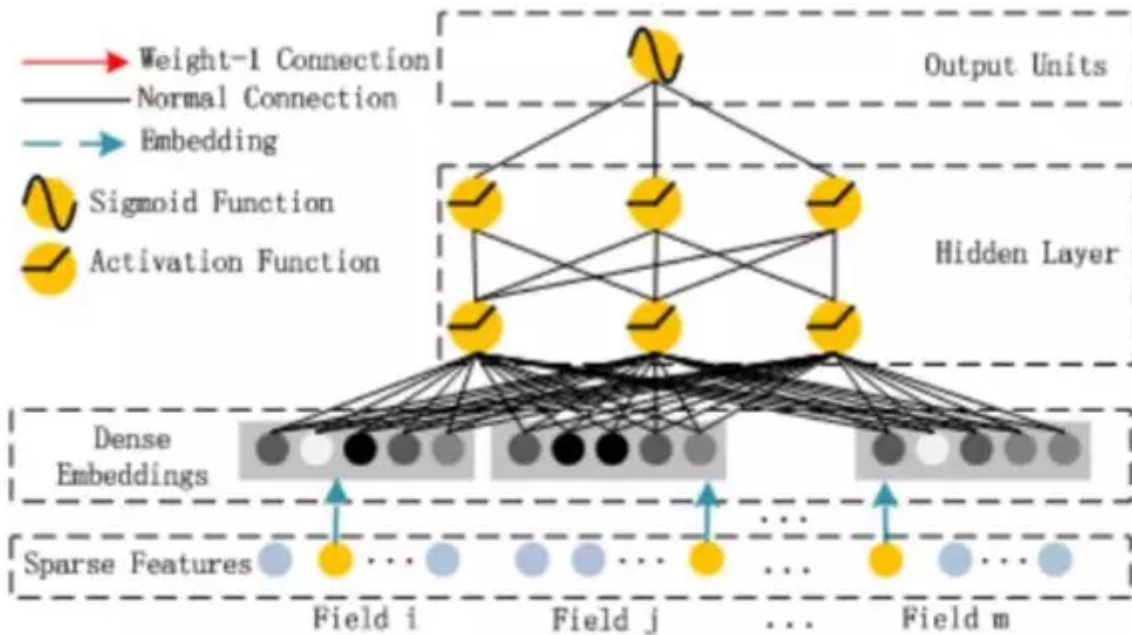
最底层是输入的稀疏原始特征(One-hot结果)，将特征按照Field进行区分，不理解Field含义的同学请学习下FFM。接着进入了Embedding层，将特征限定到有限的向量空间。Embedding层启到一个压缩的作用，不然推荐场景数据是非常稀疏的，比如一个用户A到电商平台B买东西，A可能只购买了一个商品，但是电商平台B有1000个商品，那么购买产品这个特征通过one-hot衍生就是一个1000维度向量，而且在1000纬度里只有1维度有值。Embedding层会缓解参数爆炸。

接着进入FM层和DNN层，两者共用底层的embedding数据，所以效率很高，整个数据embedding只需要计算一次。

在FM层，主要的工作是通过特征间交叉得到低阶特征，以二阶特征为主。这里面指的注意的是，FM算法的特征交叉不是直接拿原始特征相互交叉计算，而是交叉特征因子分解后的结果，这样更能挖掘出特征深层的信息。



而在DNN层就没有特别多要说的点，就是利用深度学习数据的在网络间的传播性得到高阶特征。



那最终二分类结果怎么判定也是比较粗暴，利用一个sigmoid求和函数的方式，分别参考FM和DNN的结果做出判断：

$$\hat{y} = \text{sigmoid}(y_{FM} + y_{DNN}),$$

04 总结

总体来看，DeepFM会在接下来引领推荐场景一段时间，毕竟这种既考虑低阶又考虑高阶特征的方式具有好的覆盖性，不过整体架构设计上我觉得没有特别多新颖的地方，个人还是很期待在算法领域有本质性突破的技术产生。

目前基于Tensorflow的版本已经比较成熟而且可以达到不错的效果。有兴趣的同学可以看看这个代码实现：

<https://github.com/ChenglongChen/tensorflow-DeepFM>

里面有数据可以下载，跑起来很简单~