xDeepFM:CTR预估之特征交叉的艺术

原创 二品炼丹师一元 炼丹笔记 1周前

收录于话题
#搜索推荐前沿算法

20个

xDeepFM:Combining Explicit and Implicit Feature Interactions for Recommender Systems(KDD18)

开学

xDeepFM是19年之前所有竞赛中排名非常靠前的一种方案，而xDeepFM最出名的在于它的特征交叉学习部分,也就是CIN层，可谓是一种艺术般的交叉。其也在海量的数据竞赛中展现了不俗的成绩。下面我们来看看这个模型究竟做了啥？为什么做CTR预估不得不读呢？

模型解析

xDeepFM的网络框架如下图所示：

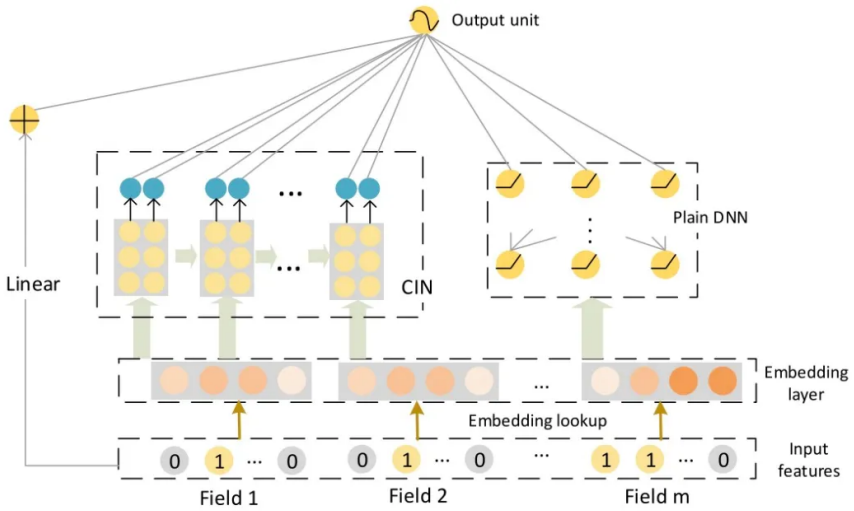


Figure 5: The architecture of xDeepFM.

从上面的图上我们可以发现xDeepFM可以细分为三大块,一个基于底层特征的线性部分，一个基于特征Embedding的DNN部分(implicit feature interactions)以及CIN部分(Explicit feature interactions)。我们按照该图看看模型每一步都在做什么，尤其是CIN层做了哪些操作，为什么能在诸多数据竞赛中拿到相较于DeepFM等模型那么大的优势。

Embedding Layer

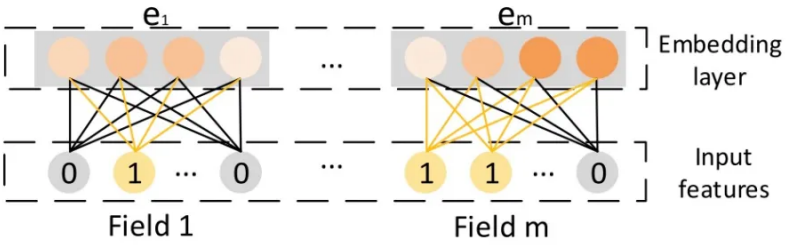
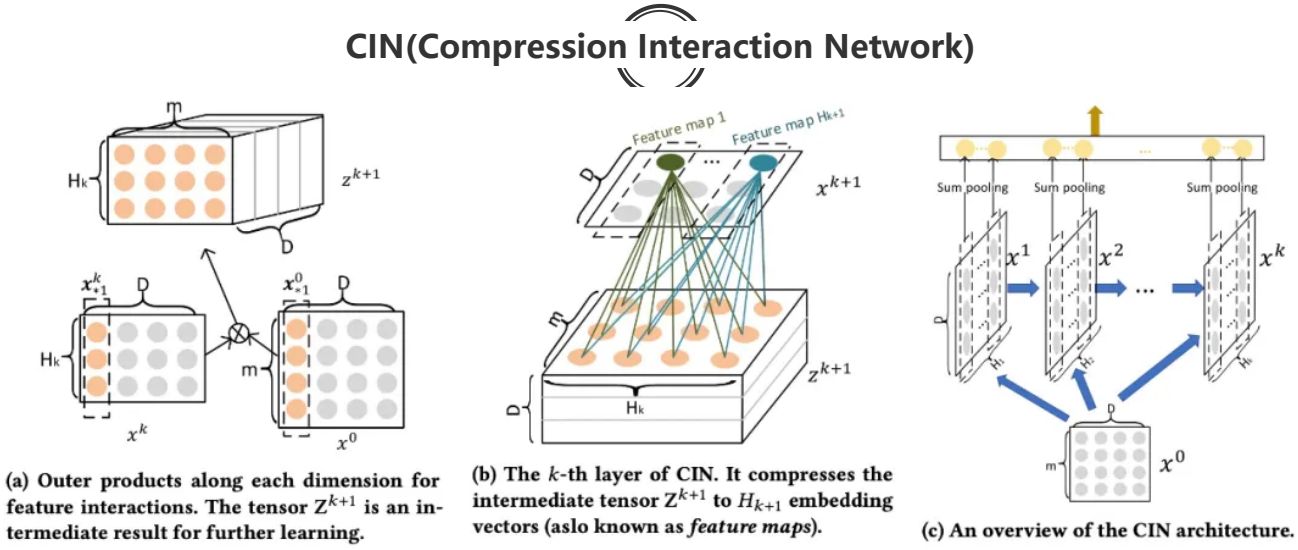


Figure 1: The field embedding layer. The dimension of embedding in this example is 4.

此处embedding做的事情就是将传统的单个特征映射到一个D维度的dense特征上,假设我们有m个field,最终我们便得到:

$$e = [e_1, e_2, \ldots, e_m] \in R^{m*D}$$



(a) Outer products along each dimension for feature interactions. The tensor Z^{k+1} is an intermediate result for further learning.

(b) The k-th layer of CIN. It compresses the intermediate tensor Z^{k+1} to H_{k+1} embedding vectors (aslo known as *feature maps*).

(c) An overview of the CIN architecture.

Figure 4: Components and architecture of the Compressed Interaction Network (CIN).

xDeepFM在CIN层实现了特征的**显示交叉**，究竟是如何做到的呢？就是M层 = M-1层 + 0层的思路，即第M阶的交叉特征是由M-1层的特征和第0层(原始特征)交叉得到的。具体地，我们假设第0层的原始特征为$X_0 \in R^{m*D}$, $X^0_{i,*} = e_i$,同时我们假设第k层的特征为$X^k \in R^{H_k*D}$,其中H_k为第k层的网络的特征向量个数,所以$H_0 = m$,那么我们要想得到第k层的特征,就可以通过下面的式子进行计算:

- $X^k_{h,*} = \sum_{i=1}^{H_{k-1}} \sum_{i=1}^{H_0=m} W^{k,h}_{i,j} \left(X^k_{i,} \right.$

其中$1 \le h \le H_k$, $W^{k,h} \in R^{H_{k-1} *m}$为第$h$层特征向量的参数矩阵。也就是说我们第$k$层的第$h$个特征向量是由第$k-1$层的每一个特征向量与第0层的每一个特征向量进行Hadamard乘积然后乘上一个系数矩阵最后全部相加得到的。所以说特征交叉是显示的。

为了能显示利用到每一层的交叉特征,我们最后需要将每一层的交叉特征输出,但是如果我们直接全部输出的话,可能会带来一个较大的问题,就是特征太多了，后面再接入全连接层的话会占据更多的内存和计算资源。所以我们使用sum pooling。这样第k层第i个向量的输出为:

- $p^k_i = \sum_{j=1}^{D} X^k_{i,j}$,

那么第k层的输出即为:

- $p^k = \left[p_1^k, p_2^k, \ldots, p_{H_k}^k\right]$

最终CIN的输出为:

- $p^+ = \left[p^1, p^2, \ldots, p^T\right] \in R^{\sum_i^T H_i}$

1. CIN与RNN的关系

CIN中下一层的输出都依赖于**上一层的输入**以及额外的输入，和RNN是非常相似的。

2. CIN与CNN的关系

我们发现第k层的每一个新的向量都是由第$k-1$层的所有向量以及第0层的所有向量分别进行element-wise相乘, 然后形成"图像"$H_{k-1} * H_0 * D$,我们再使用filter - $W^{k,h}$与其进行操作得到下一层的新向量，最终我们将$H_{k-1} * H_0$压缩为了H_k个向量,这也是compressed名字的由来。

模型输出

$$y_{xDeepFM} = \sigma(w_{linear}^T a + w_{dnn}^T x_{dnn}^k +$$

其中σ为sigmoid函数, p^+为CIN层的输出, a是原始特征。

CIN复杂度分析

空间复杂度

- **CIN层的空间复杂度**：第k层的参数W^k为$H_k * H_{k-1} * m$, 最后一层的输出有$\sum_{k=1}^T H_k$个参数,所以CIN一共有的参数个数为:$\sum_{k=1}^T H_k * (H_{k-1} * m)$

- **PlainDNN层的空间复杂度**:

因为平时我们的m和H_k通常不会非常大，所以我们$W^{k,h}$一般是可以接受的。此外我们还可以用矩阵分解的方式来降低空间复杂度。

时间复杂度

- CIN层的计算时间复杂度：计算Z^{k+1}的时间复杂度为$O(mHD)$,因为我们有H个特征map,所以计算T层的CIN时间复杂度为$O(mH^2DT)$

- PlainDNN的时间计算复杂度：$O(mDH + H^2T)$

所以xDeepFM的核心问题在于时间复杂度上面。

实验

实验部分主要回答下面几个问题：

1. 是否CIN真的做到了高阶的交叉？
2. 是否有必要将Explicit和Implicit的网络结合？
3. 网络的设置对于模型最终的影响是什么样的？

是否CIN真的做到了高阶的交叉？

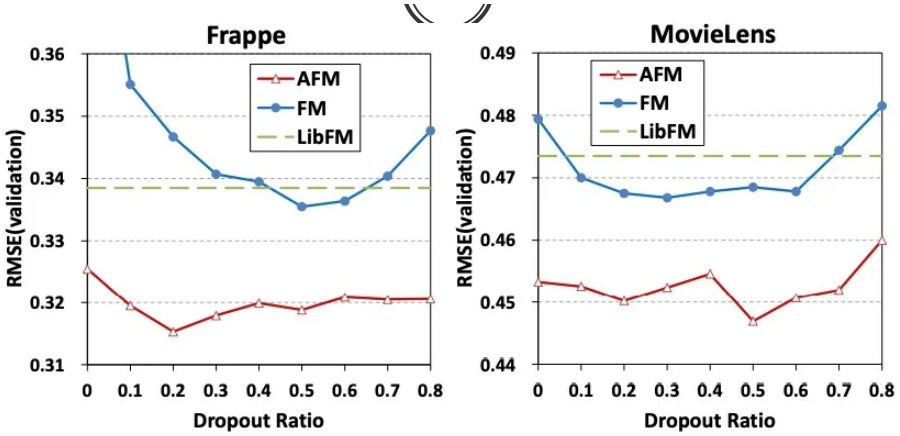


Figure 2: Validation error of AFM and FM *w.r.t.* different dropout ratios on the pair-wise interaction layer

- 单独的CIN在所有的数据集上都取得了最好的效果，所以CIN模块是非常有必要的。

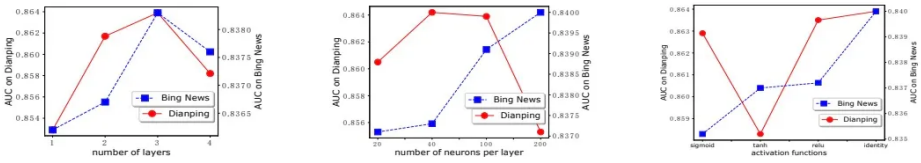
是否有必要将Explicit和Implicit的网络结合？

Table 3: Overall performance of different models on Criteo, Dianping and Bing News datasets. The column *Depth* presents the best setting for network depth with a format of (cross layers, DNN layers).

	Criteo			Dianping			Bing News		
Model name	AUC	Logloss	Depth	AUC	Logloss	Depth	AUC	Logloss	Depth
LR	0.7577	0.4854	-,-	0.8018	0.3608	-,-	0.7988	0.2950	-,-
FM	0.7900	0.4592	-,-	0.8165	0.3558	-,-	0.8223	0.2779	-,-
DNN	0.7993	0.4491	-,2	0.8318	0.3382	-,3	0.8366	0.2730	-,2
DCN	0.8026	0.4467	2,2	0.8391	0.3379	4,3	0.8379	0.2677	2,2
Wide&Deep	0.8000	0.4490	-,3	0.8361	0.3364	-,2	0.8377	0.2668	-,2
PNN	0.8038	0.4927	-,2	0.8445	0.3424	-,3	0.8321	0.2775	-,3
DeepFM	0.8025	0.4468	-,2	0.8481	0.3333	-,2	0.8376	0.2671	-,3
xDeepFM	**0.8052**	**0.4418**	3,2	**0.8639**	**0.3156**	3,3	**0.8400**	**0.2649**	3,2

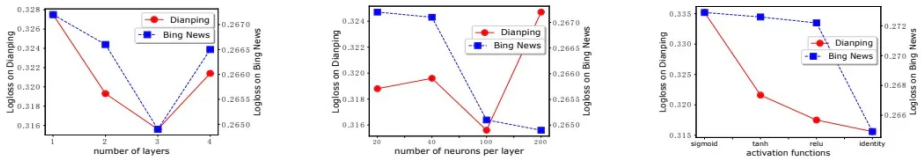
- 从实验结果上看，我们发现将Explicit和Implicit的网络结合能带来非常大的提升；xDeepFM相较于DNN有很大的提升。

网络的设置对于模型最终的影响是什么样的？



(a) Number of layers. (b) Number of neurons per layer. (c) Activation functions

Figure 6: Impact of network hyper-parameters on AUC performance.



(a) Number of layers. (b) Number of neurons per layer. (c) Activation functions

Figure 7: Impact of network hyper-parameters on Logloss performance.

- 增加网络层的深度可以提升效果, 把网络层数设置为3在数据集上的效果是最好的;
- 增加CIN中feature maps的个数早期可以提升效果的，太大可能会带来过拟合(例如Dianping数据集,100就可以了,200的时候效果会下降);
- 激活函数使用identity效果是最好的;

小结

本文提出了xDeepFM算法，CIN模块可以显示的控制特征交叉的阶数(通过vector-wise的形式)在大量数据集上的结果也显示了xDeepFM的卓越效果。

参考文献

1. xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems:https://arxiv.org/pdf/1803.05170.pdf
2. https://github.com/Leavingseason/xDeepFM/blob/master/exdeepfm/src/exDeepFM.py

我是二品炼丹师一元，目前跟着大哥们学习CTR炼丹已经快四个月了,有兴趣的欢迎关注我们的公众号,周周有彩蛋,月月有惊喜。