深入理解推荐系统：特征交叉组合模型演化简史

原创 Coggle Coggle数据科学 9月30日

收录于话题
#深入理解推荐系统

7个

写在前面

【推荐系统】专栏历史文章：

深入理解YouTube推荐系统算法

深入理解推荐系统：召回

深入理解推荐系统：排序

深入理解推荐系统：Fairness、Bias和Debias

深入理解推荐系统：推荐系统中的attention机制

作为【推荐系统】系列文章的第六篇，将以推荐系统中的“特征的自动化组合”作为今天的主角，主要介绍能够自动学习特征组合的模型（DNN、FM、FNN、PNN、DeepFM、DCN和xDeepFM等），并从其发展和演变角度进行介绍。

背景

我们知道mlp的网络结构天然的就能学习到高阶的特征组合能力，但它是以一种隐式的方式建模特征之间的交互关系，并且我们无法确定它到底学习到了多少阶的交叉关系。对于高维稀疏的原始特征，在输入dnn之前，一般会经过embedding处理，每个类别的原始特征都会被映射到一个低位稠密的实数向量，称为embedding向量。FM模型中的隐向量也可以理解为embedding向量。embedding中的元素用我们称为bit，因此可以得知，DNN中的高阶特征交互是bit-wise级的（同一个类别中的对应的emb向量中的元素也会相互影响）。但FM是显示构建的特征交叉，是以向量级（vector-wise）的方式构构建的

- DNN—隐式—bit-wise级的特征交叉
- FM—显示—vector-wise级的特征交叉

FM

论文： Factorization Machines

核心思想

FM是一种通用的预测方法，在即使数据非常稀疏的情况下，依然能估计出可靠的参数进行预测。与传统的简单线性模型不同的是，因子分解机考虑了特征间的交叉，对所有嵌套变量交互进行建模（类似于SVM中的核函数）。此外，FM的模型还具有可以用线性时间来计算，以及能够与许多先进的协同过滤方法（如Bias MF、svd++等）相融合等优点。

FM通过特征对之间的隐变量内积来提取特征组合，其函数形式如下：

$$y = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle v_i, v_j \rangle x_i x_j$$

对于每个原始特征，FM都会学习一个隐向量。模型通过穷举所有的特征对，并进行逐一检测特征对的权值来自动识别出有效的特征组合。特征对的权值则是通过该特征对涉及的两个原始特征的隐向量的内积来计算。

FM模型总结

FM的优势就在于对特征组合和维度保证两方面的处理。首先是**特征组合**，通过对两两特征组合，引入交叉项特征，提高模型得分；其次是应对**维度爆炸**，通过引入隐向量（且对参数矩阵进行矩阵分解），完成对特征的参数估计。

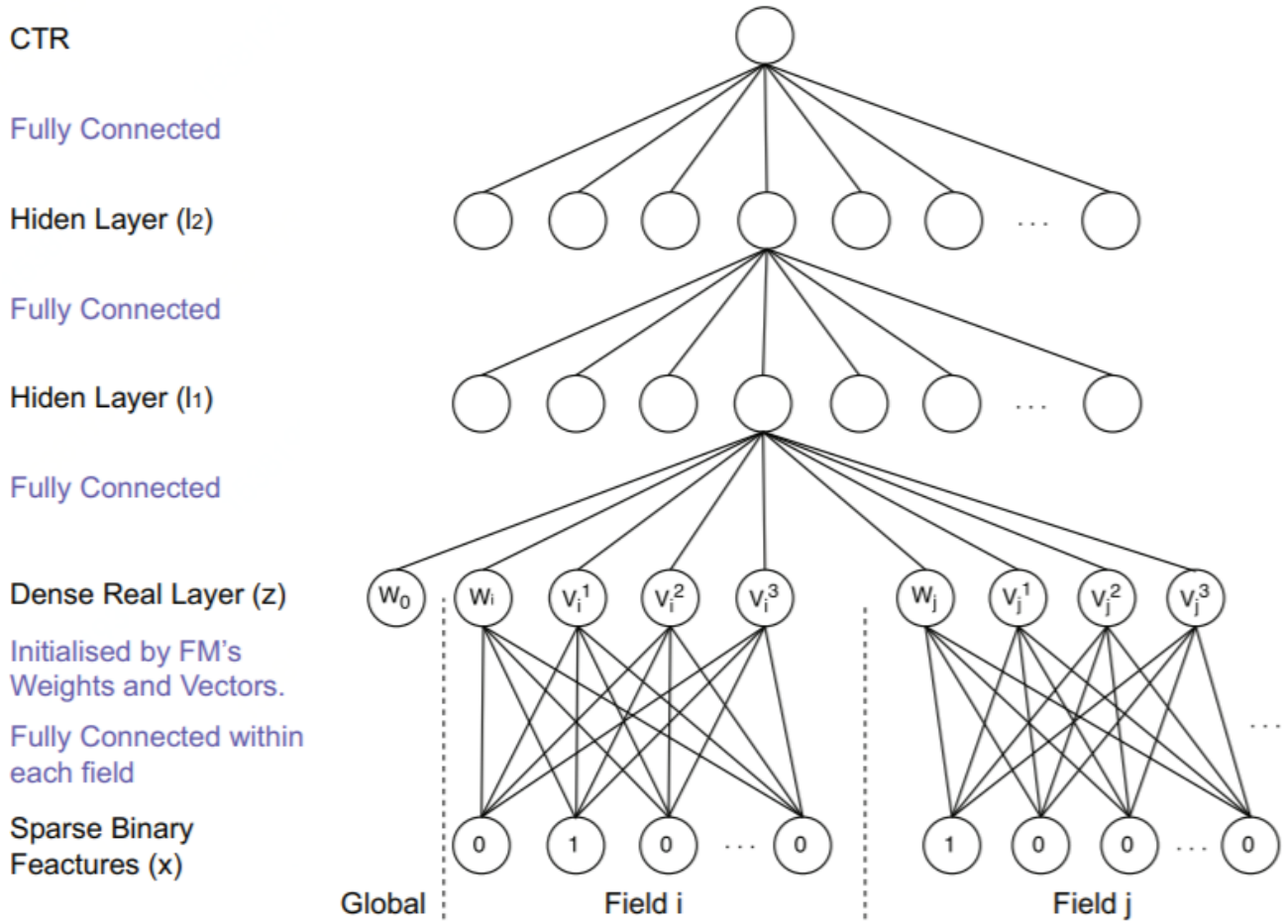
FNN

论文：Deep Learning over Multi-field Categorical Data – A Case Study on User Response Prediction

核心思想

FNN模型最先提出了一种增强FM模型的思路，就是用FM模型学习到的隐向量初始化深度神经网络模型（MLP），再由MLP完成最终学习

模型结构图



对图中的一些变量进行一下解释：x是输入的特征，它是大规模离散稀疏的。它可以分成N个Field，每一个Field中，只有一个值为1，其余都为0（即one-hot）。

$$\boldsymbol{z}_i = \boldsymbol{W}_0^i \cdot \boldsymbol{x}[\text{start}_i : \text{end}_i] = (w_i, v_i^1, v_i^2, \ldots, v_i^K),$$

Field i的则可以表示成 x[start_i : end_i]，w^i_0为Field i的embedding矩阵。z_i为embedding后的向量。它由一次项 w_i 二次项 v_i=(v^1_i,v^2_i,...,v^k_i) 组成，其中K是FM中二次项的向量的维度。而后面的 l_1，l_2 则为神经网络的全连接层的表示。详细解释一下基于FM的预训练：嵌入后的向量 Z_i=(w_i,v^1_i,v^2_i,...,v^k_i) ,其中 w_i 就是FM里面的一次性系数，而 v^k_i 就是二次项的系数，可以详细对照看FM的公式：

$$y_{\text{FM}}(\boldsymbol{x}) := \text{sigmoid}\Big(w_0 + \sum_{i=1}^{N} w_i x_i + \sum_{i=1}^{N} \sum_{j=i+1} \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle x_i x_j\Big),$$

模型有着十分显著的特点

1. 采用FM预训练得到的隐含层及其权重作为神经网络的第一层的初始值，之后再不断堆叠全连接层，最终输出预测的点击率.

2. 可以将FNN理解成一种特殊的embedding+MLP，其要求第一层嵌入后的各领域特征维度一致，并且嵌入权重的初始化是FM预训练好的

3. 这不是一个端到端的训练过程，有贪心训练的思路。而且如果不考虑预训练过程，模型网络结构也没有考虑低阶特征组合

FNN模型总结

FNN为高阶bit-wise级的特征交叉，其优点是每个特征的嵌入向量是预先采用FM模型训练的，因此在学习DNN模型时，训练开销降低，模型能够更快达到收敛。不过也存在一些缺点，具体Embedding 的参数受 FM 的影响，不一定准确；预训练阶段增加了计算复杂度，训练效率低；FNN 只能学习到高阶的组合特征，模型中没有对低阶特征建模。

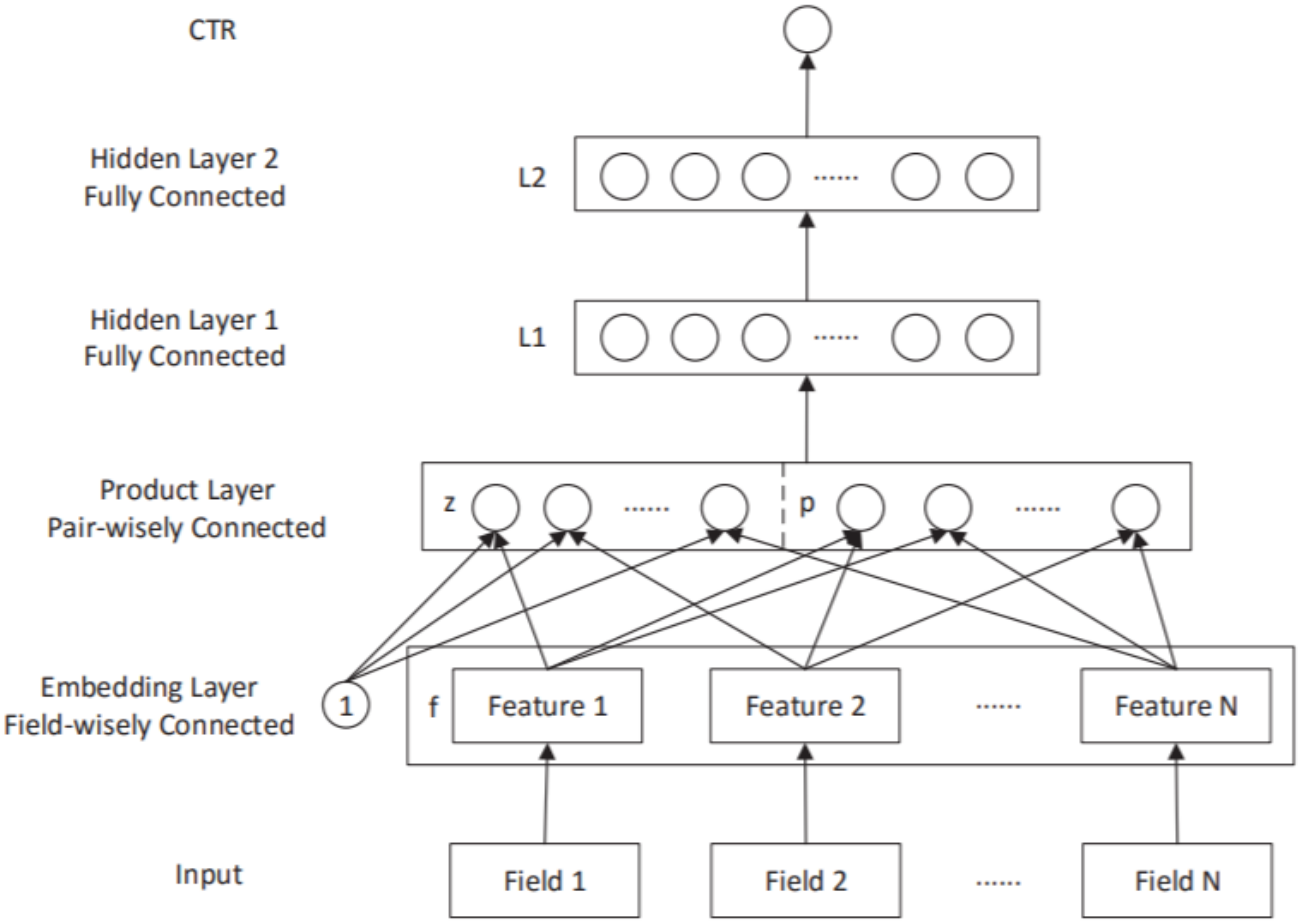
PNN

论文：Product-based Neural Networks for User Response Prediction

核心思想

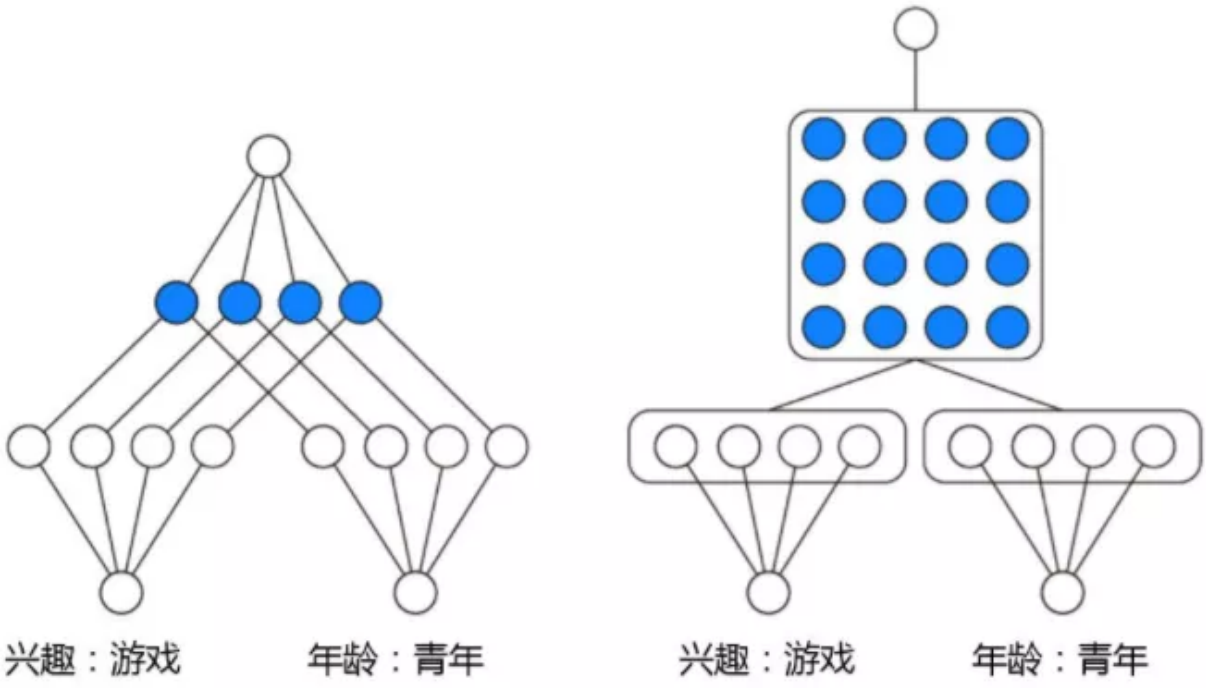
普通的Embedding+MLP模型，NN层之间都使用“add operation”，通过激活函数来引入非性。作者认为，单纯的“add”也许不足以捕获不同的Filed特征间的相关性。而且一些相关研究表明“product”相比“add”能更好得捕捉特征间的关系，因此作者希望在NN中显示地引入“product”操作，从而更好地学习不同Field特征间的相关性，于是有了在DNN结构中引入product layer的尝试

模型结构图



PNN与标准的「Embedding+MLP」差异仅在于引入了Product Layer。上图中Product Layer左边Z部分其实就是将Embedding层学到的嵌入直接原封不动地搬来，右边P部分才是我们讨论的重点。

注意，product layer 中每个节点是两两Field的embedding对应的“product”结果，而非所有Field的。根据 product 函数的不同选择，PNN也有不同实现，这里的想象空间就很多了。文中尝试了相对常见的向量内积（inner product）和外积（outer product），对应 IPNN 和OPNN。



- **IPNN**

IPNN的交叉项使用了内积g(fi, fj) = <fi, fj>。f个filed，两两求内积共计交叉项p部分的参数共f*(f-1)/2（f为特征的field个数，原始论文里用的N）个，线性部分z部分参数共f*k个。需要学习的参数为：

（1）FM部分：1+ n + n*k

（2）product部分：（f*k + f*(f-1)/2）*H1

（3）MLP部分：H1*H2+H2*1

- **OPNN**

OPNN用矩阵乘法来表示特征的交叉，g(fi, fj)=fifit。f个field两两求矩阵乘法，交叉项p共f*(f-1)/2*k*k个参数。线性部分z部分参数共f*k个。需要学习的参数为：

（1）FM部分：1+ n + n*k

（2）product部分：（f*k + f*(f-1)/2*k*k）*H1

（3）MLP部分：H1*H2+H2*1

PNN模型总结

pnn在特征交叉组合方面是vector-wise低价交叉+bit-wise高阶交叉，但是没有线性部分。PNN与FM相比，舍弃了低阶特征，也就是线性的部分，这在一定程度上使得模型不太容易记住一些数据中的规律。

另外注意，PNN的product layer并不是一次简单的内积或者外积操作。而是先使用内积或者外积对特征进行交叉（field之间），并对交叉后接全连接层的操作进行了优化（IPNN）。此外由于一般情况下N>M，所以IPNN特征交叉后保留的信息更多（N*N）,OPNN保留的信息更少（M*M）。同时IPNN的复杂度更高。

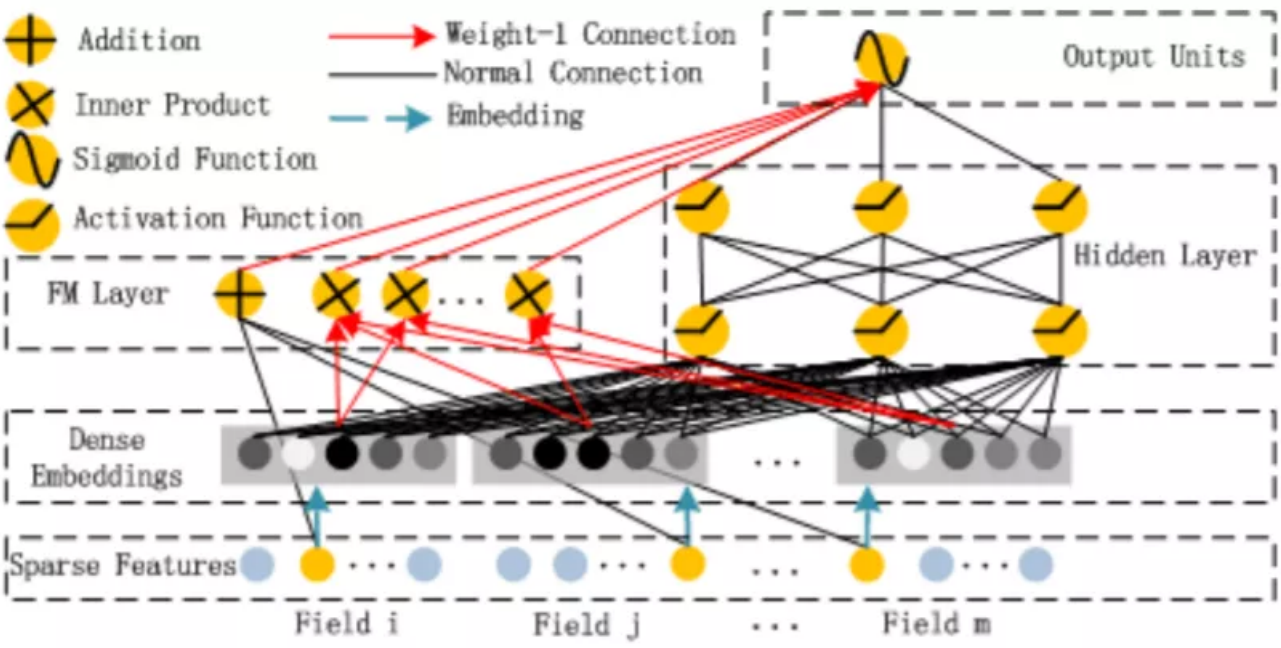
DeepFM

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

核心思想

DeepFM模型是一种可以从原始特征中抽取到各种复杂度特征的端到端模型，没有人工特征工程的困扰，DeepFM模型包含FM和DNN两部分，FM模型可以抽取low-order特征，DNN可以抽取high-order特征。无需类似Wide&Deep模型人工特征工程。

模型结构图



从模型结构图可以看出，FM 和 DNN共用embedding层的结果，然后FM部分负责低阶特征组合（二阶），而DNN负责高阶特征组合，然后将低阶和高阶部分合在一起训练。

FM Layer

不单可以建模1阶特征，还可以通过隐向量点积的方法高效的获得2阶特征表示，即使交叉特征在数据集中非常稀疏甚至是从来没出现过。这也是FM的优势所在。

$$y_{FM} = \langle w, x \rangle + \sum_{j_1=1}^{d} \sum_{j_2=j_1+1}^{d} \langle V_i, V_j \rangle x_{j_1} \cdot x_{j_2},$$

DNN部分

$$y_{DNN} = \sigma(W^{|H|+1} \cdot a^H + b^{|H|+1}),$$

输出部分

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

为什么需要同时使用高阶和低阶的结果？在wide&deep的论文中指出是generalization 和memorization，低阶部分（memorization）主要是从历史数据中，发现item和特征之间的直接相关性，体现准确性；而高阶部分（generalization）主要是从历史数据中发现之前很少出现的或者没有出现的特征组合，体现出新颖性。通俗地说，把准确性高的特征用低阶部分简单地表达出来，而对于目前还不清楚是对结果有什么影响的特征在利用隐藏层做高阶特征组合，然后把低阶和高阶结果合并。

为什么FM和DEEP部分要共用embedding层？因为这样能使得DeepFM从原始的数据输入中学习到低维和高维的特征交叉。

DeepFM模型总结

模型是FM(是显示一阶+二阶向量级交叉)与DNN（隐式高阶bit级交叉）两部分的组合。DeepFM最大的突破点是，将低阶特征和高阶特征组合起来，由浅层模型和深层模型联合训练得到，加和起来进行ctr的预测，其中FM layer可以抽取低纬特征，对稀疏输入非常友好，而深度神经网络可以获取高维特征，具有很好的对于隐含特征关系的挖掘能力，使模型具有很好的泛化性能。FM layer和DeepFM的是共享embedding向量的，而embedding也是基于FM机制的，FM得到的隐向量作为权重得到新的固定长度的稠密特征。

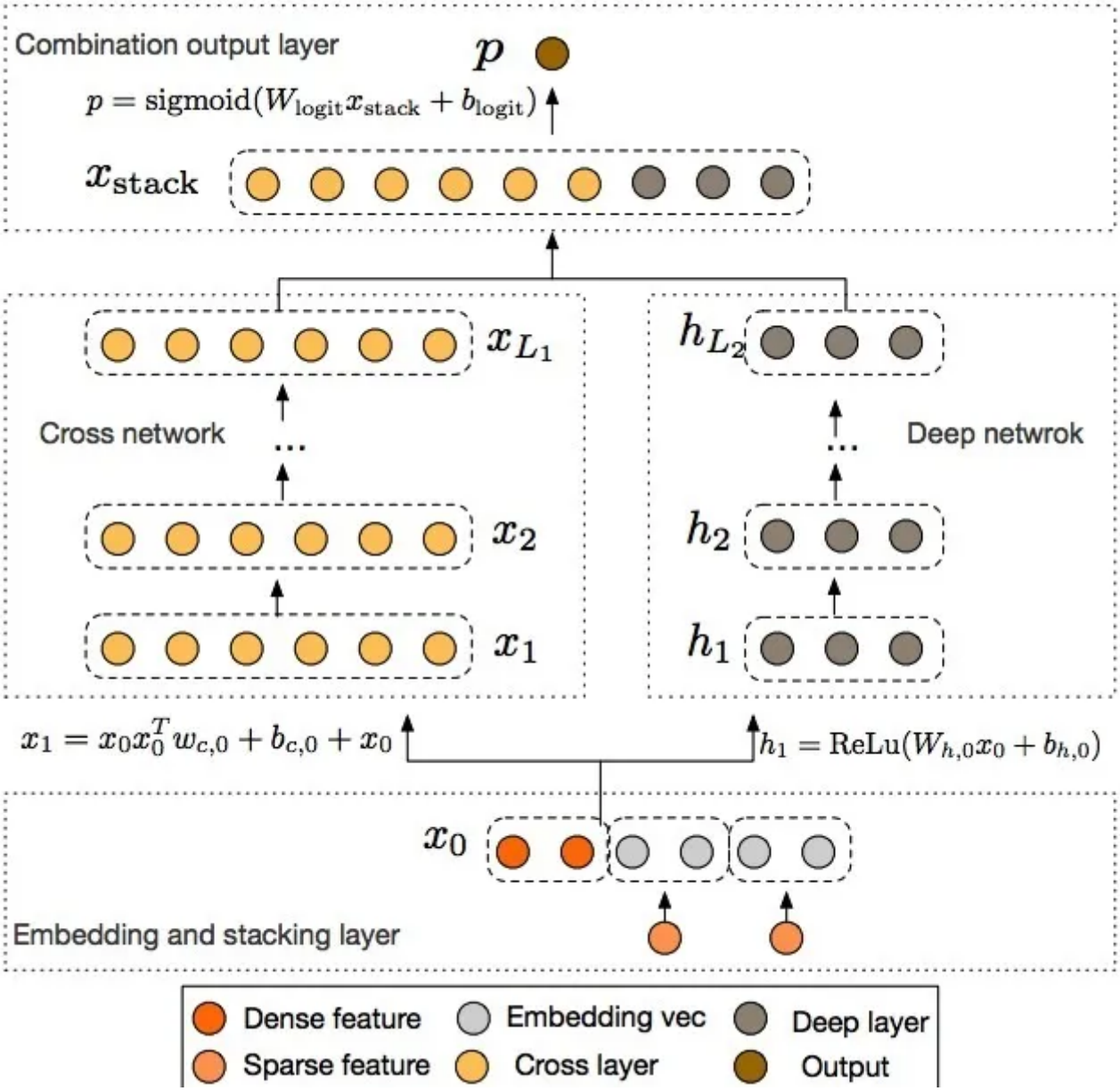
DCN

论文：Deep & Cross Network for Ad Click Predictions

核心思想

在ctr预估中，特征交叉是很重要的一步，但目前的网络结构，最多都只学到二级交叉。LR模型采用原始人工交叉特征，FM自动学习xi和xj的二阶交叉特征，而PNN用product方式做二阶交叉对于更高阶的特征交叉，只有让deep去学习了。为解决这个问题，google在2017年提出了Deep&Cross Network，简称DCN的模型，可以任意组合特征，而且不增加网络参数.Cross的目的是以一种显示、可控且高效的方式，自动构造有限高阶交叉特征

模型结构图



DCN模型结构

模型结构共分为4个部分，分别为 Embedding and Stacking Layer（特征预处理输入）、Cross network（自动化特征显式交叉）、Deep network（特征隐式交叉）和Combination output Layer（输出）。

- **Embedding and Stacking Layer**

此部分主要是对原始特征进行预先处理操作，其中，对sparse特征进行embedding，对于multi-hot的sparse特征，embedding之后再做一个简单的average pooling；对dense特征归一化，然后和embedding特征拼接，作为随后Cross层与Deep层的共同输入，即：

$$\mathbf{x}_0 = \left[\mathbf{x}_{\text{embed},1}^T, \ldots, \mathbf{x}_{\text{embed},k}^T, \mathbf{x}_{\text{dense}}^T\right],$$

之所以不把embedding和stacking分开来看，是因为很多时候，embedding和stacking过程是分不开的。前面讲到的各种 XX-based FM 网络结构，利用FM学到的v向量可以很好的作为embedding。而在很多实际的业务结构，可能已经有了提取到的embedding特征信息，例如图像的特征embedding，

text的特征embedding，item的embedding等，还有其他连续值信息，例如年龄，收入水平等。这些embedding向量stack在一起后，一起作为后续网络结构的输入。当然，这部分也可以用前面讲到的FM来做embedding。为了和原始论文保持一致，这里我们假设X0向量维度为d（上文的网络结构中为k），这一层的做法就是简答的把各种embedding向量concat起来。

- **Deep network**

在embedding and stacking layer之后，网络分成了两路，一路是传统的DNN结构。表示如下

$$\mathbf{h}_{l+1} = f(W_l \mathbf{h}_l + \mathbf{b}_l),$$

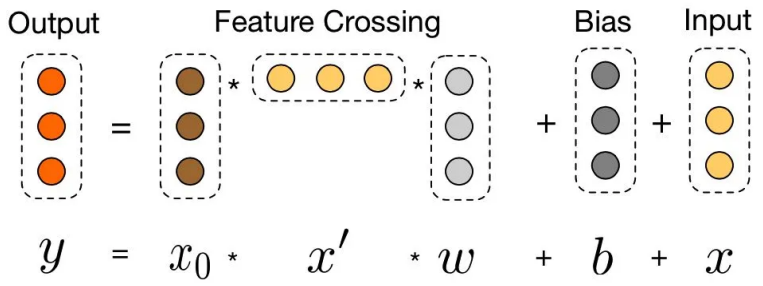
为简化理解，假设每一层网络的参数有m个，一共有Ld层，输入层由于和上一层连接，有d*m个参数（d为x0向量维度），后续的Ld-1层，每层需要m*(m+1)个参数，所以一共需要学习的参数有d*m+m*(m+1)*(Ld-1)。

- **Cross network**

Embedding and stacking layer输入后的另一路就是DCN的重点工作了。假设网络有L1层，每一层和前一层的关系可以用如下关系表示

$$\mathbf{x}_{l+1} = \mathbf{x}_0 \mathbf{x}_l^T \mathbf{w}_l + \mathbf{b}_l + \mathbf{x}_l = f(\mathbf{x}_l, \mathbf{w}_l, \mathbf{b}_l) + \mathbf{x}_l,$$

可以看到f是待拟合的函数，xl即为上一层的网络输入。需要学习的参数为wl和bl，因为xl维度为d， 当前层网络输入xl+1也为d维，待学习的参数wl和bl也都是d维度向量。因此，每一层都有2*d的参数（w和b）需要学习，经过Lc层的cross layer network后，在该layer最后一层Lc层的输出为Lc2的d维向量。网络结构如下



- **Combination output Layer**

经过cross network的输出XL1(d维）和deep network之后的向量输入（m维）直接做concat，变为一个d+m的向量，最后套一个LR模型，需要学习参数为1+d+m。

DCN模型总结

DCN为bit-wise高阶交叉，DCN引入的cross network理论上可以表达任意高阶组合，同时每一层保留低阶组合，参数的向量化也控制了模型的复杂度。

1) 有限高阶：叉乘阶数由网络深度决定，深度Lc对应最高Lc+1阶的叉乘。

2) 自动叉乘：Cross输出包含了原始特征从一阶（即本身）到 Lc+1阶的所有叉乘组合，而模型参数量仅仅随输入维度成线性增长：

$$2 * d * L_c$$

3) 参数共享：不同叉乘项对应的权重不同，但并非每个叉乘组合对应独立的权重（指数数量级）， 通过参数共享，Cross有效降低了参数量。此外，参数共享还使得模型有更强的泛化性和鲁棒性。例如，如果独立训练权重，当训练集中

$$x_i \neq 0 \wedge x_j \neq 0$$

这个叉乘特征没有出现 ，对应权重肯定是零，而参数共享则不会，类似地，数据集中的一些噪声可以由大部分正常样本来纠正权重参数的学习。

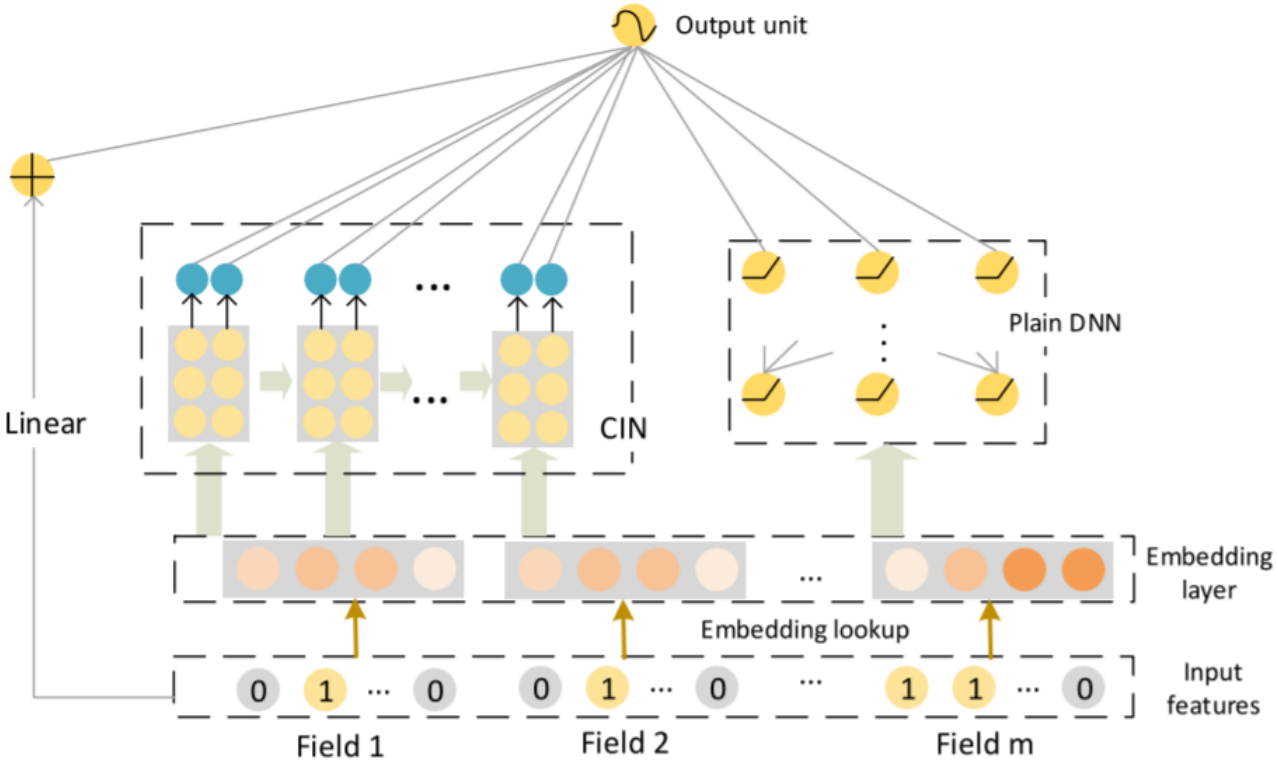
xDeepFM

论 文 ： xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems

核心思想

为了实现自动学习显式的高阶特征交互，同时使得交互发生在向量级上，xDeepFM首先提出了一种新的名为压缩交互网络（Compressed Interaction Network，简称CIN）的模型

模型结构图

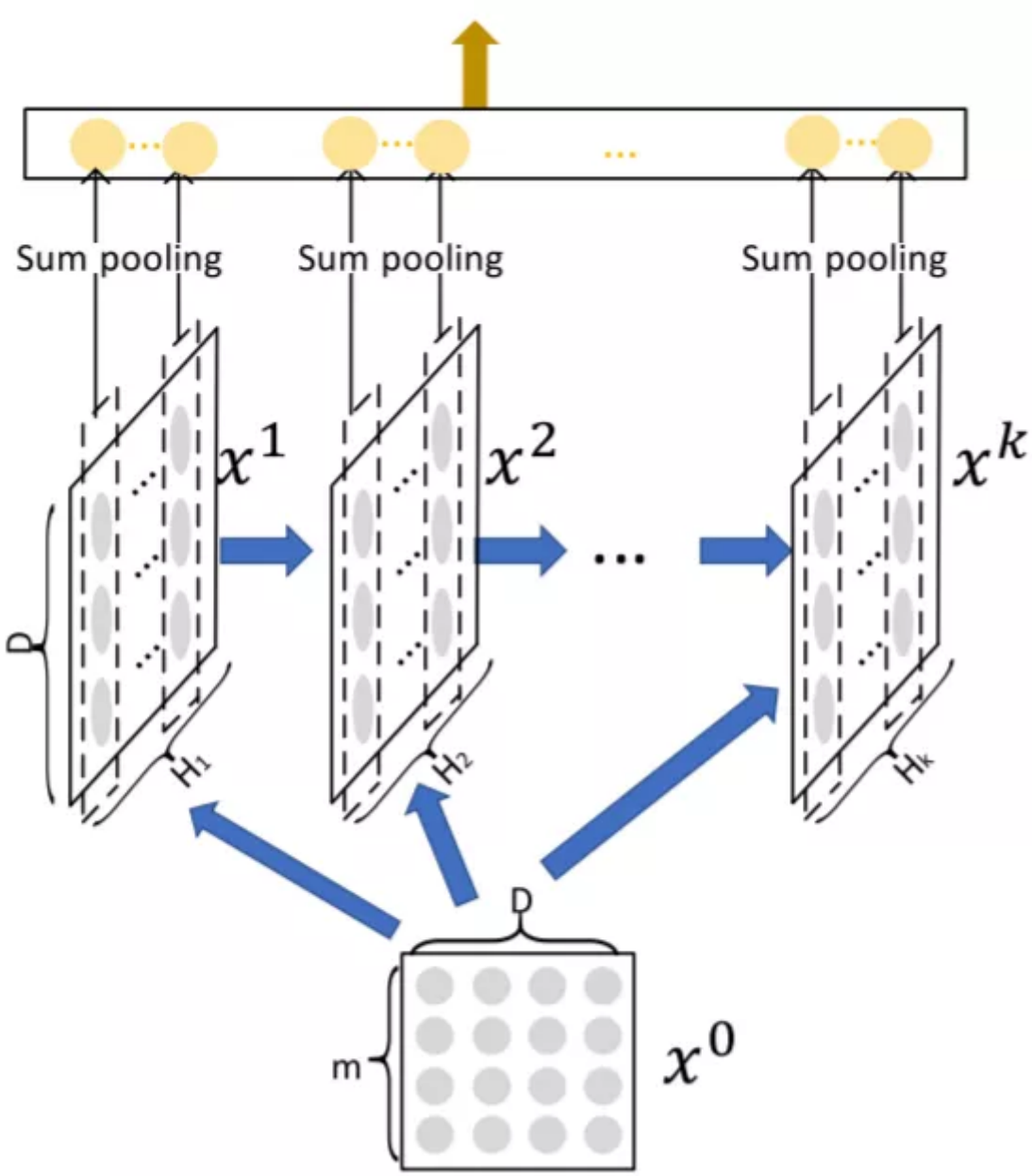


如图所示，CIN层的输入来自Embedding层，假设有 m 个field，每个field的embedding vector维度为 D，则输入可表示为矩阵 $\boldsymbol{X}^0 \in \mathbb{R}^{m*D}$

$$\hat{y} = \sigma(\mathbf{w}_{linear}^T \mathbf{a} + \mathbf{w}_{dnn}^T \mathbf{x}_{dnn}^k + \mathbf{w}_{cin}^T \mathbf{p}^+ + b)$$

可以看到整体的模型分为三个子模型。这里只有CIN是论文的创新部分，是显式特征组合。加上Linear是为了引入人工设计的特征，加上DNN是为了引入隐式特征组合。所以只要理解了CIN的结构就能理解这篇论文。

压缩交互网络CIN结构



首先看一下CIN的整体架构，可以看到类似于之前的Deep&Cross，同样是用 X^0 层跟第k个中间层做交叉，生成k+1个中间层。最终把所有的中间过程以sum pooling的方式结合在一起，输出结果。

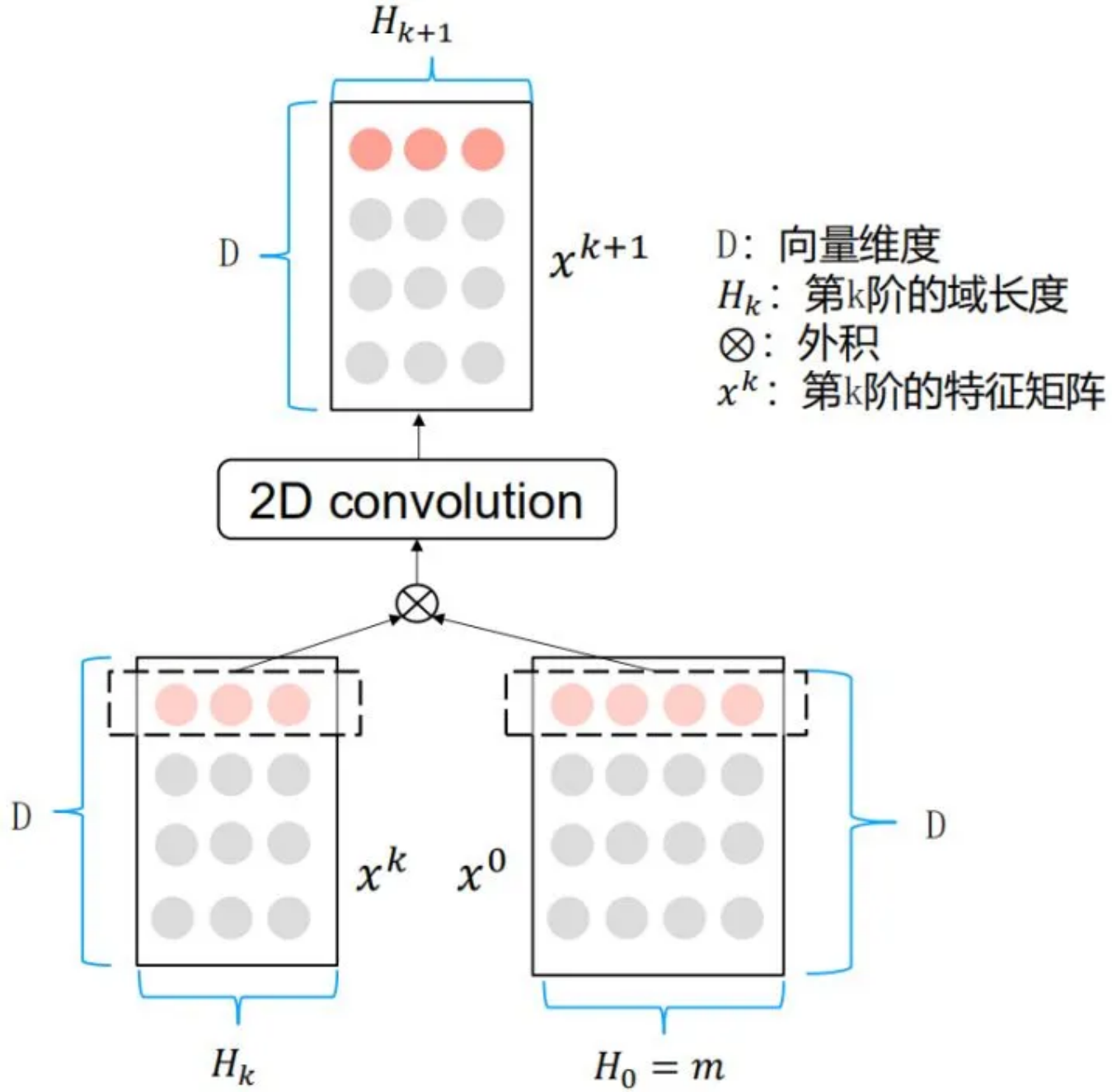
具体的特征组合公式如下：

$$\mathbf{X}_{h,*}^{k} = \sum_{i=1}^{H_{k-1}} \sum_{j=1}^{m} \mathbf{W}_{ij}^{k,h} (\mathbf{X}_{i,*}^{k-1} \circ \mathbf{X}_{j,*}^{0})$$

（1）交互是在向量层次上应用的，而不是在位层次上；

（2）高阶特征交互是明确测量的；

（3）网络的复杂度不会随着相互作用的程度。



每个维度上的外积用于特征交互。张量 x^{k+1} 是进一步学习的中间结果。

xDeepFM模型总结

xDeepFM为vector-wise的高阶交叉（CIN）+bit-wise的高阶交叉+低阶特征三部分的组合。xDeepFM的核心在于CIN网络，提出了vector级别的交叉特征，不同filed的embedding vector进行交叉；在利用Hadamard product做完交叉特征之后，为了不使维数扩大，使用了sum pooling得到一个和embedding长度一样的向量（假设为d），然后做H个这样的操作，就形成了一个H*d的新向量， 此时H与对应的原始的filed数量m可以不同，有点类似卷积中channel的感觉，最后在每一个交叉layer都会将d维向量sum pooling送到最后和deep model部分拼接后运算。