

[深度模型] xDeepFM: DeepFM和Deep Cross Network大升级

原创 xxxhuang 推荐算法学习笔记 6月19日

一. 概述

我们知道特征组合对推荐系统非常重要。人工进行特征组合非常耗时耗力，并且要对业务要有较深的理解。基于FM的模型，则可以帮助我们自动学习特征的交叉和组合。同时结合DNN的模型，如Wide& Deep、DCN和DeepFM，可以同时高效学习到低维特征组合和高维特征组合。

在xDeepFM中，paper作者设计了一种新型的网络Compressed Interaction Network(CIN)，可以像DCN那样随着Cross Network层数的递增，特征的交叉维度也会递增。并弥补了DCN存在的一些不足。

二. 需了解的基础知识

- DeepFM: paper名称为《DeepFM: A Factorization-Machine based Neural Network for CTR Prediction》
- DCN: paper名称为《Deep & Cross network for Ad Click Predictions》

如果对DeepFM和DCN没有了解，建议先阅读相关paper或资料。也可以查看笔者曾经介绍DeepFM和DCN的文章。链接如下所示

[深度模型] 推荐算法工程师必学模型DeepFM介绍

xxxhuang, 公众号: 推荐算法学习笔记

[深度模型] 推荐算法工程师必学模型DeepFM介绍

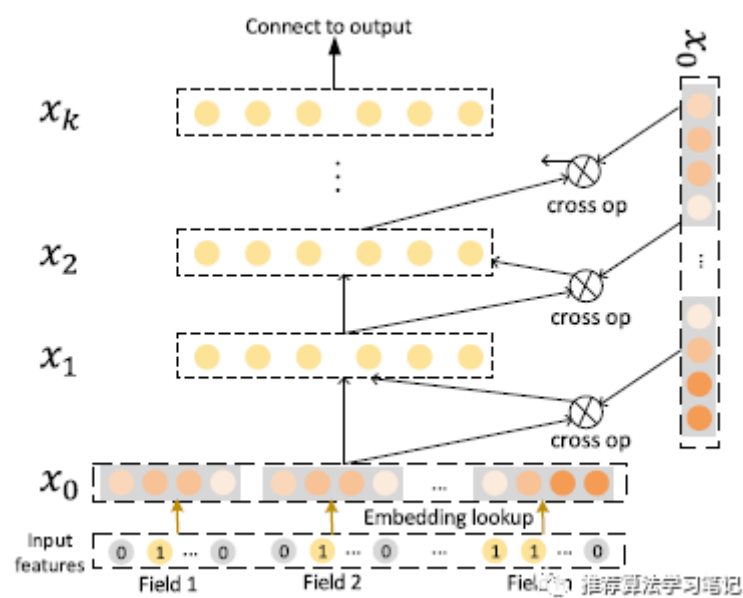
[深度模型] Deep & Cross Network (DCN)

xxxhuang, 公众号: 推荐算法学习笔记

[深度模型] Deep & Cross Network (DCN)

三. DCN的局限性

DCN的CrossNetwork架构如下所示



公式如下所示

$$\mathbf{x}_k = \mathbf{x}_0 \mathbf{x}_{k-1}^T \mathbf{w}_k + \mathbf{b}_k + \mathbf{x}_{k-1}$$

推荐算法学习日记

其中w，b和x分别表示网络的weights，bias和output，k表示第k层。

Cross Network可以显示地学习到高维的特征组合，但它存在一个问题，就是最终的 \mathbf{x}_k 结果是 \mathbf{x}_0 乘以一个标量(scalar)

证明如下所示，截取自paper

PROOF. When $k=1$, according to the associative law and distributive law for matrix multiplication, we have:

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{x}_0(\mathbf{x}_0^T \mathbf{w}_1) + \mathbf{x}_0 \\ &= \mathbf{x}_0(\mathbf{x}_0^T \mathbf{w}_1 + 1) \\ &= \alpha^1 \mathbf{x}_0 \end{aligned} \quad (4)$$

where the scalar $\alpha^1 = \mathbf{x}_0^T \mathbf{w}_1 + 1$ is actually a linear regression of \mathbf{x}_0 . Thus, \mathbf{x}_1 is a scalar multiple of \mathbf{x}_0 . Suppose the scalar multiple statement holds for $k=i$. For $k=i+1$, we have :

$$\begin{aligned} \mathbf{x}_{i+1} &= \mathbf{x}_0 \mathbf{x}_i^T \mathbf{w}_{i+1} + \mathbf{x}_i \\ &= \mathbf{x}_0 ((\alpha^i \mathbf{x}_0)^T \mathbf{w}_{i+1}) + \alpha^i \mathbf{x}_0 \\ &= \alpha^{i+1} \mathbf{x}_0 \end{aligned} \quad (5)$$

where, $\alpha^{i+1} = \alpha^i (\mathbf{x}_0^T \mathbf{w}_{i+1} + 1)$ is a scalar. Thus \mathbf{x}_{i+1} is still a scalar multiple of \mathbf{x}_0 . By induction hypothesis, the output of cross network \mathbf{x}_k is a scalar multiple of \mathbf{x}_0 .

 推荐算法学习日记

虽然 \mathbf{x}_k 结果是 \mathbf{x}_0 乘以一个标量，但是并不意味着 \mathbf{x}_0 和 \mathbf{x}_k 是线性关系的，因为对于不同的 \mathbf{x}_0 ，这个标量是不一样的。

因此，由于这个的局限性，限制了Cross Network的表达。

四. Compressed Interaction Network(CIN)

(1) CIN公式

了解 Cross Network 的局限性后，paper 作者提出了 Compressed Interaction Network(CIN)。

假设我们的输入一个有 m 个field(raw feature)，首先将这 m 个field都转换成都是 D 维的embedding。则可以得到一个 $m \times D$ 维的embedding矩阵，用 \mathbf{X}_0 来表示。

因此对于第 k 层 \mathbf{x}_k ，使用下面的公式进行计算

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

 推荐算法学习日记

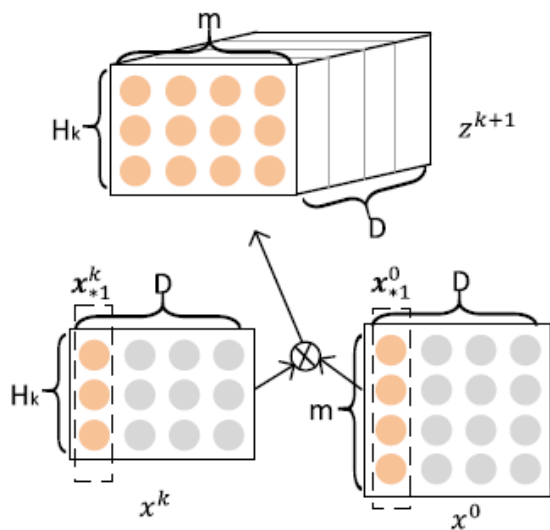
$\mathbf{x}_{k,h,*}$ 表示第 k 层第 h 个field对应的embedding， $\mathbf{w}_{k,h,ij}$ 是一个scalar，空心圆表示Hadamard product，例如 $\langle a_1, a_2, a_3 \rangle \circ \langle b_1, b_2, b_3 \rangle = \langle a_1 b_1, a_2 b_2, a_3 b_3 \rangle$

刚开始看这个公式觉得很复杂，其实它就是将第 $k-1$ 层的embedding和第0层的embedding两两做Hadamard product，然后weight sum后输出一个embedding。

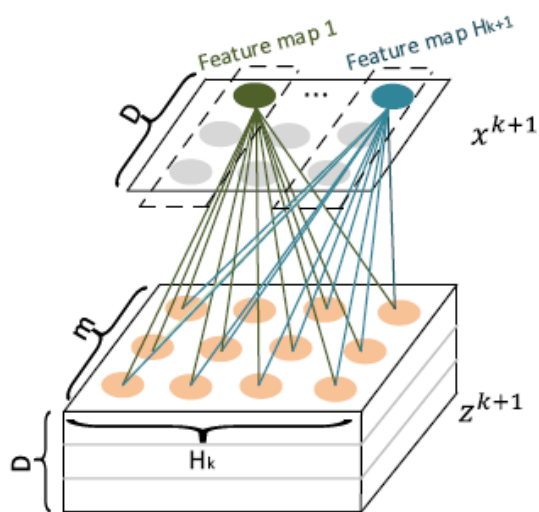
同时，我们可以决定第 k 层有多少个field，使用 H_k 表示field的个数。

(2) CIN和CNN的关系

CIN其实是可以转换成CNN的，如下图所示。具体细节可参见paper。



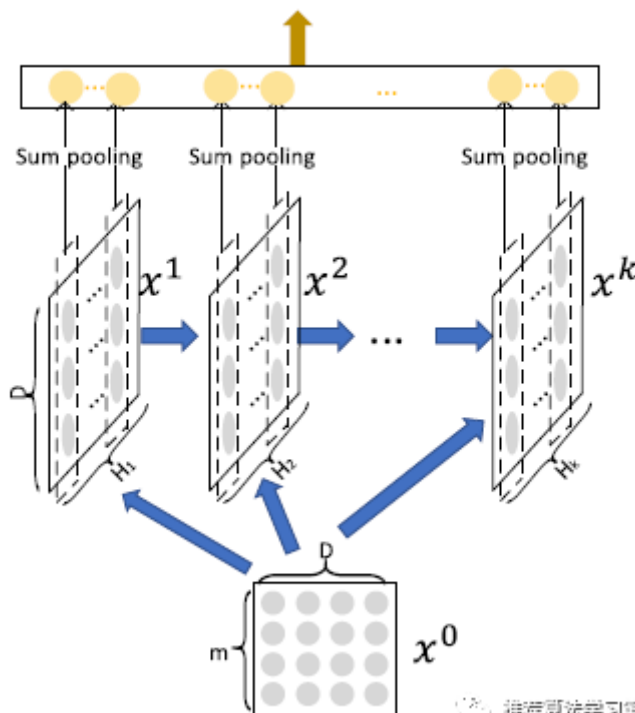
(a) Outer products along each dimension for feature interactions. The tensor Z^{k+1} is an intermediate result for further learning.



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

(3) CIN输出

在得到每一层的 x_k 输出后，我们将每个field（ D 维向量）的元素相加起来，得到一个 H_k 维的向量作为输出。所有层的输出构成一个sum pooling层，如下图所示



最后sum pooling层使用sigmoid函数作为最终输出，公式如下所示

$$y = \frac{1}{1 + \exp(\mathbf{p}^+ \mathbf{w}^o)}$$

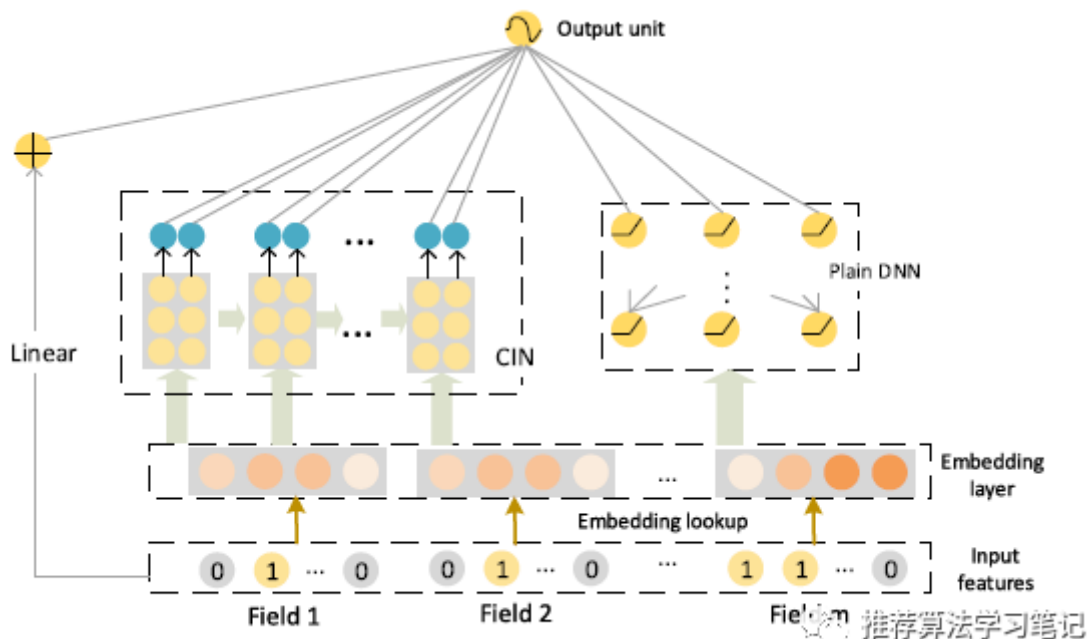
where \mathbf{w}^o are the regression parameters.

推荐算法学习笔记

五. 与DNN结合

最后，我们可以和将CIN和DNN结合起来，构造一个更强大的模型，这便是xDeepFM。

整体架构图如下所示



公式如下所示

$$\hat{y} = \sigma(\mathbf{w}_{linear}^T \mathbf{a} + \mathbf{w}_{dnn}^T \mathbf{x}_{dnn}^k + \mathbf{w}_{cin}^T \mathbf{p}^+ + b) \quad (15)$$

损失函数为

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \quad (16)$$


加上L2正则化

$$\mathcal{J} = \mathcal{L} + \lambda_* \|\Theta\|$$

六. 和DeepFM的关联

根据公式

$$X_{h,*}^k = \sum_{i=1}^{H_{k-1}} \sum_{j=1}^m W_{ij}^{k,h} (X_{i,*}^{k-1} \circ X_{j,*}^0)$$

 推荐算法学习日记

我们将k设为1并且第1层field的个数H1=1，然后sum pooling后，就是DeepFM。可见xDeepFM是DeepFM的泛化版本。

七. 总结

以上便是xDeepFM的全部内容。如果有问题，欢迎随时和我联系