xDeepInt：一种用于建模向量级和按位特征交互的混合架构

Yachen Yan
yachen.yan@creditkarma.com
Credit Karma 旧金山，加利福尼亚州

Liubo Li
liubo.li@creditkarma.com
Credit Karma 旧金山，加利福尼亚州

arXiv:2301.01089v1 [cs.LG] 3 Jan 2023

摘要

学习特征交互是大规模点击率预测和推荐取得成功的关键。在实践中，手工特征工程通常需要穷举搜索。为了降低特征工程所需的大量人工成本，研究人员提出了多种基于深度神经网络（DNN）的方法，以端到端的方式学习特征交互。然而，现有方法要么无法同时学习向量级交互和位级交互，要么无法以可控的方式将二者结合起来。本文提出了一种新模型 xDeepInt，它基于一种名为多项式交互网络（PIN）的新型网络架构，可递归地学习高阶向量级交互。通过集成子空间交叉机制，我们使 xDeepInt 能够在有界阶数下平衡向量级和按位特征交互的组合。基于该网络架构，我们定制了一种组合优化策略，用于执行特征选择和交互选择。我们实现了所提出的模型，并在三个真实世界数据集上评估了模型性能。实验结果证明，xDeepInt 的效果和性能均优于最先进的模型。我们已开源 xDeepInt 的 TensorFlow 实现：https://github.com/yanyachen/xDeepInt。

CCS 概念

• 计算方法；**• 机器学习**；**• 机器学习方法**；**• 神经网络**；**关键词**

点击率预测、推荐系统、显式特征交互、深度神经网络

ACM 参考格式：
Yachen Yan 和 Liubo Li。2020。xDeepInt：一种用于建模向量级和按位特征交互的混合架构。载于 圣迭戈 *2020*：*ACM SIGKDD*知识发现与数据挖掘大会，*2020*年*8*月*22*日至*27*日，美国加利福尼亚州圣迭戈。ACM，纽约，美国，9页。https://doi.org/10. 1145/nnnnnnn.nnnnnnn

允许免费制作本作品全部或部分内容的数字或纸质副本，用于个人或课堂用途，前提是副本不得用于盈利或商业利益，也不得分发，并且副本首页须附上本声明和完整引文。对于本作品中由 ACM 以外的其他方拥有版权的部分，必须尊重其版权。允许在注明出处的情况下摘录。若要以其他方式复制、转载、发布到服务器或分发至列表，则须事先获得明确许可和/或支付费用。请通过 permissions@acm.org 申请许可。*DLP-KDD 2020*，*2020*年*8*月*24*日，美国加利福尼亚州圣迭戈© 2020 计算机协会。ACM ISBN 978-1-4503-XXXX-X/18/06...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 引言

点击率（CTR）预测模型 [23] 是大规模推荐系统、在线广告和搜索排序 [7, 12, 19, 30]的重要组成部分。在在线市场场景中，准确估算点击率能够使推荐系统向用户展示他们更愿意浏览和探索的商品，这会直接影响短期收入和长期用户体验。

点击率预测模型的输入特征（例如用户 ID、物品 ID、物品类别、站点域名）通常采用多字段分类格式 [31]，并通过字段感知独热编码和多热编码 [34] 进行转换。每个字段的表示都是一个稀疏二进制向量。每个字段对应的基数决定了稀疏向量的维度。这些稀疏向量的拼接自然会生成高维且稀疏的特征表示。

在点击率预测模型中，探索有用的特征交互对于提升模型性能起着至关重要的作用[7, 8, 16, 22, 28]。传统上，数据科学家会基于领域知识搜索并构建人工设计的特征交互，以增强模型性能。在实践中，高质量特征交互的构建需要耗费大量时间和人力 [12]。此外，鉴于特征数量庞大且基数较高，手动提取所有可能的特征交互是不可行的 [7]。因此，在高维且稀疏的特征空间中，自动且高效地学习低阶和高阶特征交互，成为学术界和工业界提升点击率预测模型性能的关键问题。

深度学习模型凭借强大的特征学习能力，在推荐系统中取得了巨大成功。学术界和工业界提出了多种深度学习架构（例如，[7, 8, 13, 16, 20, 21, 24, 25, 28]）。然而，现有模型都将深度神经网络作为构建模块，用于学习阶数不受限的高阶隐式按位特征交互。在对显式特征交互进行建模时，现有方法只能高效地捕捉低阶显式交互。学习高阶交互通常需要更高的计算成本。

本文提出了一种高效的基于神经网络的模型 xDeepInt，能够显式学习向量级和位级乘法特征交互的组合。受多项式回归的启发，我们设计了新颖的多项式交互网络层，以显式捕获有界次数的向量级交互。为了以可控的方式同时学习位级和向量级交互，我们将 PIN 与子空间交叉机制相结合，从而显著提升了模型性能，并带来更大的灵活性。位级交互的次数

会随着子空间数量的增加而增长。总之，本文的贡献如下：

- 我们设计了一种名为 xDeepInt 的新型神经网络架构，可同时对向量级交互和按位特征交互进行显式建模，无需联合训练深度神经网络，也无需使用非线性激活函数。该模型轻量，却比许多结构更复杂的现有模型性能更优。

- 受高阶多项式逻辑回归的启发，我们设计了多项式交互网络（PIN）层，以递归方式学习高阶显式特征交互。通过调整 PIN 层的数量，可以控制交互的阶数。我们还进行了分析，以展示 PIN 的多项式逼近特性。

- 我们引入了一种子空间交叉机制，用于在 PIN 层内部对不同字段之间的按位交互进行建模。PIN层与子空间交叉机制相结合，使我们能够控制按位交互的阶数。随着子空间数量的增加，我们的模型能够动态学习更细粒度的按位特征交互。

- 我们设计了一种与所提出模型架构相协调的优化策略。我们对嵌入表应用组套索FTRL，将整行权重压缩为零，从而实现特征选择。对于PIN层中的权重，我们直接应用FTRL进行优化。权重的稀疏性能够筛选出特征交互。

- 我们在三个真实世界数据集上进行了全面实验。结果表明，在极高维

nal

且稀疏的设置下，xDeepInt优于现有最先进模型。我们还对xDeepInt的超参数设置进行了敏感性分析，并针对DNN的集成进行了消融研究。

2 相关工作

自深度神经网络在学习混合类型输入数据的有效特征表示以及以端到端方式拟合模型方面占据主导地位以来，基于深度学习的模型便已应用于工业界的点击率预测问题[30]。这一优势能够减少人工特征设计的工作，并自动学习特征交互。

2.1 隐式交互建模

大多数基于 DNN 的方法在初始步骤中，将高维稀疏类别特征和连续特征映射到低维潜在空间。基于 DNN 的方法无需设计特定的模型架构，只需将堆叠后的嵌入特征向量输入深度前馈神经网络，即可学习高阶隐式特征交互。

Deep Crossing 网络 [24] 在前馈结构中采用残差层，以更稳定地学习高阶交互。一些混合网络架构包括 Wide & Deep 网络（WDL）[7],、基于乘积的神经网络（PNN）[20,21],、Deep & CrossNetwork（DCN）[28],、深度因子分解机（DeepFM）[8] 和极深因子分解机（xDeepFM）[16]

将前馈神经网络用作深度组件，以学习高阶隐式交互。对隐式高阶交互的补充，提升了仅对显式交互建模的网络的性能 [2]。

然而，这类方法会在位级上隐式检测所有特征交互，效率不高。此外，交互的阶数也不受限制。 [16]

2.2 显式交互建模

Deep & Cross 网络（DCN） [28]以显式方式探索位级特征交互。具体而言，DCN 的每个交叉层都会构造所有交叉项，以实现位级交互。递归交叉层的数量决定了位级特征交互的阶数。

一些近期模型通过特定形式的向量乘积显式学习向量级特征交互。深度因子分解机（DeepFM） [8] 通过联合学习特征嵌入，将因子分解机层与前馈神经网络结合起来。因子分解机层通过特征 i 与特征 j 的内积 $x_i, x_j = \sum_{t=1}^{k} x_{it} x_{jt}$ 对成对向量级交互进行建模。随后，将向量级交互与前馈神经网络的输出单元拼接。产品神经网络（PNN） [20, 21]引入内积层和外积层，分别用于学习显式向量级交互和位级交互。xDeepFM [16] 使用具有类 RNN 架构的压缩交互网络（CIN）学习显式向量级交互，该网络通过哈达玛积学习所有可能的向量级交互。卷积滤波器和池化机制用于提取信息。FiBiNET [13] 利用压缩与激励网络动态学习特征的重要性，并通过双线性函数对特征交互进行建模。

在近期的序列模型研究中，Transformer [26] 架构已被广泛用于理解相关特征之间的关联。借助多层多头自注意力神经网络，AutoInt [25] 能够学习输入特征的不同阶特征组合。残差连接 [9, 27] 则用于传递不同程度的特征交互。

上述方法通过外积、核积或多头自注意力来学习显式特征交互，但计算成本高昂。

3 模型

本节概述 xDeepInt 的架构。首先，我们介绍输入与嵌入层，该层将连续特征和高维类别特征映射到一个稠密向量。其次，我们介绍多项式交互网络（PIN），它利用带有残差连接的迭代交互层，显式地学习向量级交互 [9, 27] 。第三，我们采用子空间交叉机制来建模位级交互。子空间的数量控制着位级交互与向量级交互的混合程度。

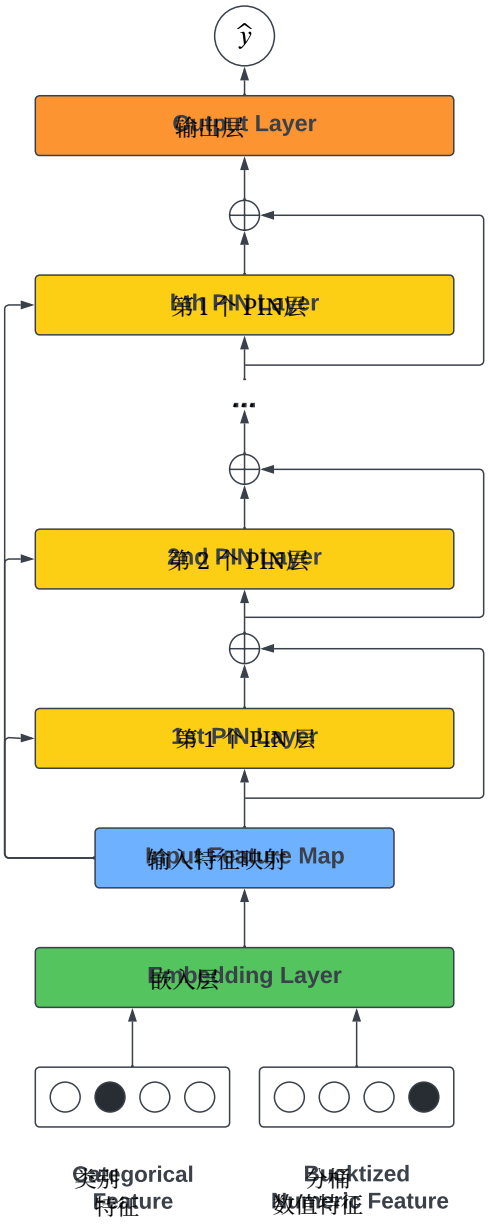


图 1：带残差连接的展开式多项式交互网络架构

3.1 嵌入层

在大规模推荐系统中，输入既包括连续特征，也包括类别特征。类别特征通常直接采用独热编码，这会导致特征空间维度过高且过于稀疏。假设我们有 F 个字段。在特征预处理步骤中，我们将所有连续特征分箱为频率相等的区间，然后将分箱后的连续特征和类别特征嵌入到相同的潜在空间 R^K，

$$\mathbf{x}_f^e = \mathbf{x}_f^o \mathbf{V}_f,$$

其中 $\mathbf{x}_f^e = [x_{f,1}, x_{f,2}, \cdots x_{f,K}]$，$x_{f,k}$是嵌入特征映射中第 f 个字段的第 k 位，$\mathbf{V}f$ 是字段 f 的嵌入矩阵，而 $\mathbf{x}_f^o$是独热向量。最后，我们将 F 个嵌入向量堆叠起来，得到一个 $F \times K$ 的输入特征映射 X_0:

$$X_0 = \begin{bmatrix} \mathbf{x}_1^e \\ \mathbf{x}_2^e \\ \vdots \\ \mathbf{x}_F^e \end{bmatrix}$$

3.2 多项式交互网络

考虑一个具有 f 个变量的 l 阶多项式，其形式如下:

$$\Pi_{j=1}^{l}(\sum_{i=1}^{f} a_{ij}x_i + b_j). \tag{1}$$

该多项式包含 x_i 的所有阶数小于或等于 l 的可能乘法组合，其迭代形式为:

$$F^{(l-1)}(x_1, \cdots, x_f)(\sum_{i=1}^{f} a_{il}x_i + b_l) \tag{2}$$

其中 $F(l-1)(x_1, \cdots x_f) = \Pi_{j=1}^{l-1}(\sum_{i=1}^{f} a_{ij}x_i + b_j)$。受其迭代形式启发，我们提出了多项式交互网络，其定义如下:

$$\begin{aligned} X_l &= f(W_{l-1}, X_{l-1}, X_0) + X_{l-1} \\ &= X_{l-1} \circ (W_{l-1}X_0) + X_{l-1} \\ &= X_{l-1} \circ [W_{l-1}X_0 + \mathbf{1}] \end{aligned} \tag{3}$$

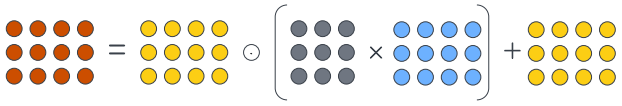
其中 $\circ$ 表示哈达玛积。例如，$[a_{i,j}]_{m\times n} \circ [b_{i,j}]_{m\times n} = [a_{i,j}b_{i,j}]_{m\times n}$。 $W_{l-1} \in R^{F\times F}$ 和 $\mathbf{1} \in R^{F\times K}$ 的所有元素均为 1。 X_{l-1}， $X_l \in R^{F\times K}$ 分别是第 (l-1) 个和第 l 个交互层的输出矩阵。与式（1）类似，第 l 个 PIN 层的输出是所有阶数小于或等于 l 的向量级特征交互的加权和。

多项式交互网络的架构受以下几个方面的启发。

首先，多项式交互网络具有递归结构。当前层的输出基于前一层的输出和一阶特征映射构建，从而确保高阶特征交互建立在前几层的低阶特征交互之上。

其次，我们使用哈达玛积对显式的向量级交互进行建模，这使我们能够更灵活地交叉共享潜在空间中各维度的比特，并保留不同阶特征交互的更多信息。

第三，我们构建一个域聚合层 $Agg(l)(X) = W_lX$，它通过线性变换 W_l 在向量级别上组合特征映射。域聚合特征图中的每个向量都可以视为一个组合特征向量，由输入特征映射的加权和构成。接着，我们对上一层的输出与本层的域聚合特征图进行哈达玛积运算。此操作使我们能够



$$X_l = X_{l-1} \circ (W_{l-1} \cdot X_0) + X_{l-1}$$

图 2：PIN层的细节

聚合层定义为 $Agg^{(l-1)}(X_0) = W_{l-1} \cdot X_0$。PIN将聚合后的1阶特征映射与前一PIN层的输出进行哈达玛积，以生成高阶向量级交互。

基于已有的（l-1）阶特征交互，探索所有可能的 l 阶多项式特征交互。

最后，我们在多项式交互网络中采用残差连接 [9, 27]，从而将不同阶数的向量级多项式特征交互结合起来，其中也包括第一层特征映射。由于多项式交互层的输出包含所有阶数的特征交互，跳跃连接使下一多项式交互层能够专注于寻找有用的高阶特征交互，同时补充低阶特征交互。随着层数增加，多项式特征交互的阶数也随之增加。所提出的多项式交互网络的循环架构能够限制多项式特征交互的阶数。

3.3 子空间交叉机制

多项式交互网络（PIN）对向量级交互进行建模。然而，PIN 并未在共享潜在嵌入空间中学习位级交互。为了交叉不同嵌入维度的比特，我们提出了子空间交叉机制，使 xDeepInt 能够学习位级交互。假设我们将嵌入空间划分为 h 个子空间，则输入特征映射 X_0 可表示为 h 个子矩阵，如下所示：

$$X_0 = [X_{0,1}, X_{0,2}, \cdots, X_{0,h}] \tag{4}$$

其中 $X_{0,i} \in R^{F \times K}/h$ 和 $i = 1, \ 2 \cdots \ h$。接下来，我们沿域维度堆叠所有子矩阵，构造堆叠输入特征映射 $X_0^{'} \in R(F*h)\times(K/h)$。

$$X_0^{'} = \begin{bmatrix} X_{0,1} \\ X_{0,2} \\ \vdots \\ X_{0,h} \end{bmatrix}, \tag{5}$$

其中，$X_{0,j} \in R^{F\times}(K/h)$ 和 h 表示子空间数量。将每个特征域的嵌入向量拆分为 h 个子向量并堆叠在一起，我们便能对齐不同嵌入维度的比特，并在堆叠后的子嵌入上构建向量级交互。因此，我们将 $X_0^{'}$ 输入多项式交互网络（PIN）：

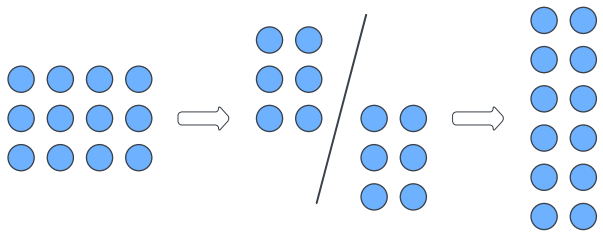


图 3：子空间交叉机制

交互网络（PIN）：

$$\begin{aligned} X_1^{'} &= X_0^{'} \circ [W_0^{'} X_0^{'} + 1] \\ &\vdots \\ X_l^{'} &= X_{l-1}^{'} \circ [W_{l-1}^{'} X_0^{'} + 1] \end{aligned} \tag{6}$$

其中$W' l \in R_{(F*h)\times(F*h)}$， $1 \in R(F*h)\times(K/h)$ 和 $X_l^{'} \in R(F*h)\times(K/h)$。

在普通 PIN 层中，特征映射的字段聚合以及通过哈达玛积构建的乘性交互都属于向量级别。经过子空间交叉机制增强的 PIN 以 h 对齐子空间作为输入，从而鼓励 PIN 通过交叉不同子空间中的特征来捕获显式的位级交互。子空间的数量 h 决定了位级交互的复杂度。更大的 h 有助于模型学习更复杂的特征交互。

3.4 输出层

多项式交互网络的输出是一个特征映射，其中包含不同阶数的特征交互，包括由残差连接保留的原始输入特征映射，以及由 PIN 学习到的高阶特征交互。对于最终预测，我们仅使用如下公式：

$$\hat{y} = \sigma\big((W_{out} X_l + b\mathbf{1}^T)\mathbf{1}\big) \tag{7}$$

其中，σ 是 Sigmoid函数，$W_{out} \in R^{1\times F}$ 是一个特征映射聚合向量，用于对特征映射中的所有特征进行线性组合；$\mathbf{1} \in R^K$ 和 $b \in R$ 是偏置。

3.5 优化与正则化

在优化方面，我们采用组套索正则化跟随正则化领导者（G-FTRL）[?] 作为嵌入层的优化器，以进行特征选择；采用跟随正则化领导者（FTRL）[19] 作为 PIN 层的优化器，以进行交互选择。

组套索 FTRL 会将每个字段中不重要特征的整个嵌入向量正则化为精确的零值，这实质上实现了特征选择，并提高了工业场景下的训练效率。组套索正则化在子空间划分机制之前应用，从而确保各子空间之间的特征选择保持一致。

FTRL 将 PIN 层中权重核的单个元素正则化为精确的零，从而排除不显著的特征交互，并对模型复杂度进行正则化。

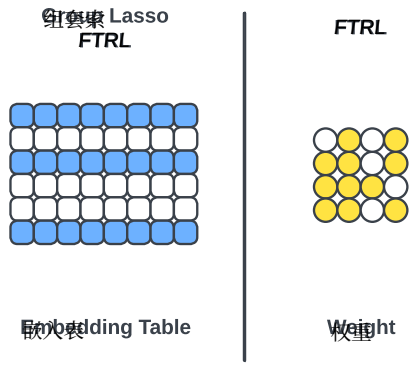


图 4：组套索FTRL 与 FTRL

组套索FTRL通过组级稀疏性对嵌入表进行正则化。FTRL通过元素级稀疏性对PIN层的权重核进行正则化。

这种优化策略利用了不同优化器的特性，分别在嵌入表和权重核上实现行级稀疏性和元素级稀疏性。因此，它提升了训练和在线服务的泛化能力与效率，在模型压缩中也发挥着重要作用。

3.6 训练

我们使用的损失函数是对数损失，

$$\text{LogLoss} = -\frac{1}{N}\sum_{i=1}^{N}(y_i log(\hat{y}_i) + (1-y_i) log(1-\hat{y}_i)) \quad (8)$$

其中 y_i 是真实标签，$\hat{y}_i$ 是估计的点击率。N 是训练样本的总数。

3.7 PIN与DCN的区别

PIN和DCN都具有迭代形式。然而，这两种网络架构在提取特征交互方面存在很大差异。

$$\begin{aligned} x_l &= f_{DCN}(x_{l-1}, w_{l-1}, b_{l-1}) \\ &= x_0 x_{l-1}^T w_{l-1} + b_{l-1} + x_{l-1} \end{aligned} \quad (9)$$

对于 DCN，特征映射会被展平并拼接成一个向量。所有高阶位级交互首先由项 $x_0 x_{l-1}^T$ 构建，然后通过线性回归聚合，以供下一层使用。这种结构导致输出具有一种特殊形式。正如 [16]，中所讨论的，DCN层的输出是 x_0 的标量倍数。[16] 也指出了其缺点：1）DCN的输出形式特殊，每个隐藏层都是 x_0 的标量倍数，因此表达能力有限；2）交互仅以位级形式进行。

PIN 使用哈达玛积构建向量级别的特征交互，从而保留向量级别的信息。为了让不同字段在向量级别进行交叉，PIN 首先通过线性变换聚合输入特征映射 $W_{l-1}X_0$，并在每个迭代 PIN 层中构建项 $X_{l-1} \circ W_{l-1}X_0$ 形式的交互。因此，PIN 保留了特征交互的向量级别结构，不会将输出限制为 X_0 的标量倍数。此外，每个 PIN 层都直接连接到输入特征映射 X_0，从而提高了模型可训练性。我们还将在后文证明 PIN 的多项式逼近性质。

3.8 xDeepInt 分析

本节将分析所提出的 xDeepInt 模型的多项式逼近性质。我们考虑一个具有l 个 PIN层、采用包含 h 个子空间的子空间交叉机制，且具有 F 个输入特征的 xDeepInt 模型，其中嵌入维度相同，均为 K。

3.8.1 多项式逼近。为了理解 PIN 如何利用向量级交互，我们考察 PIN 的多项式逼近性质。设$\mathbf{X}_{(0)} \in R^{F\times K}$ 为嵌入特征向量，其中$\mathbf{x}_i = [x_{i1}, \cdots x_{iK}]$ 是第 i行。$\mathbf{x}_{(l)i}$ $\exists x_{(l)i1}, \cdots x_{(l)iK}]$ 是第 l层输出的第 i行。那么，$x_{(l)ik}$具有以下显式形式：

$$x_{ik}^{(l)} = x_{ik}^{(0)} \prod_{r=0}^{l-1}(\sum_{i=1}^{F} w_{ij}^{(r)} x_{jk}^{(0)} + 1), \text{for } k = 1, \cdots, K \quad (10)$$

其中 $W_{(r)}$= $[w_{(r)ij}]_{1\le i,\ j\le F}$是第 r 个 PIN 层的权重矩阵。乘积 $\prod_{r=0}^{l-1}(\sum_{i=1}^{F} w_{(r)ij}\ x_{(0)jk} + 1)$ 是第 k 个比特位的嵌入输入中所有阶数小于或等于 $l-1$ 的可能交叉项的加权和。因此，$x_{(l)ik}$ 是所有包含 $x_{(0)ik}$ 且阶数小于或等于 l 的交叉项的加权和。

对于通过子空间交叉机制建模的位级交互，我们考虑子空间数量等于嵌入维度 K 的情况。在这一极端情况下，$W_{,0}X_{,0}$的每一行都是所有字段中所有比特的加权和。这种设计使得不同比特上的嵌入特征能够组合起来。更明确地说，我们考虑如下堆叠输入特征映射 $h = K$

$$X_0' = \begin{bmatrix} X_{0,1} \\ X_{0,2} \\ \vdots \\ X_{0,K} \end{bmatrix}, \quad (11)$$

where $X_{0,i} = \begin{bmatrix} x_{i,1}^{(0)} \\ x_{i,2}^{(0)} \\ \vdots \\ x_{i,F}^{(0)} \end{bmatrix} \in R^F$ .The weight matrix W_0' is given by

$$W_0' = \begin{bmatrix} w_{1,1}^{(0)} & w_{1,2}^{(0)} & \cdots & w_{1,K}^{(0)} & w_{1,K+1}^{(0)} & \cdots & w_{1,FK}^{(0)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ w_{FK,1}^{(0)} & w_{FK,2}^{(0)} & \cdots & w_{FK,K}^{(0)} & w_{FK,K+1}^{(0)} & \cdots & w_{FK,FK}^{(0)} \end{bmatrix} \in R^{FK\times FK}. \quad (12)$$

因此，$W_{\prime 0}{}^{X}{}_{\prime_0} + 1$ 的第 k 行具有如下形式：

$$\sum_{i=1}^{F}\sum_{j=1}^{K} w_{k,(i-1)K+j}^{(0)} x_{i,j} + 1. \tag{13}$$

这是所有字段中比特的线性组合，使 PIN 能够在位级利用特征映射的所有交叉。例如，$[(i-1)K+j]$-th 行 $X\prime$ 定义为

$$x_{i,j}^{(l)} = x_{i,j}^{(0)} \prod_{r=0}^{l-1}\sum_{i=1}^{F}\sum_{j=1}^{K} w_{k,(i-1)K+j}^{(r)} x_{i,j} + 1 \tag{14}$$

$x(l)\,i,j$是包含 $x(0)\,i,j$且阶数小于或等于 l 的所有交叉项的加权和。

3.8.2 时间复杂度。 在第 l 个 PIN 层计算特征映射 $W_{l-1}X$ 的成本为 O(hF^2K)。对于一个包含 L 层的 xDeepInt 模型，特征映射的总成本为 O(LhF^2K)。额外成本来自哈达玛积和残差连接，为 O(LFK)。实际上，h 并不大。因此，总时间复杂度主要取决于字段数 F 和嵌入维度 K。对于一个包含 L 层、每层有 D_k 个隐藏节点的深度神经网络，时间复杂度为 O($FK\times D_1\times D_2+\sum_{k=2}^{L-1} D_{k-1}D_kD_{k+1}$)。xDeepInt 的时间复杂度取决于子空间数量。因此，在建模更高阶的位级交互时，xDeepInt 的时间复杂度高于深度神经网络。

3 8 3 $\sum_{f=1}^{F} K\times..$

C_f C_f f

空间复杂度。 嵌入层包含参数，其中是第个字段的基数。输出层汇总最后一个 PIN 层的特征映射。因此，输出层需要 $F+1$ 个参数。子空间交叉机制在每个 PIN 层需要 $h^2\times F^2$ 个参数，这正是权重矩阵 $W_{\prime r}$ 的大小，其中 $0 \le r \le l-1$。l 个 PIN 层中共有 $(K/h)\ \times k\prime + h^2\times F^2\times l$ 个参数。通常，我们会选择较小的 h 来控制模型复杂度，而 $k\prime$ 与 K 相当。因此，xDeepInt 的整体复杂度约为 $O(h^2\times F^2\times l)$，且很大程度上受嵌入维度 K 的影响。一个普通的 L 层深度神经网络，每层有 D_k 个隐藏节点，需要 $FK\times D_1+\sum_{k=2}^{L} D_{k-1}D_k$ 个参数。其复杂度主要取决于嵌入维度以及每层的隐藏节点数。为降低 xDeepInt 的空间复杂度，我们可以采用 [16] 中介绍的方法。通过利用 L 阶分解，并将权重矩阵 $W_{\prime r}$ 替换为两个低秩矩阵，还可以进一步降低模型的空间复杂度。

4 实验

本节重点评估我们提出的模型的有效性，并回答以下问题：

- **Q1**：我们提出的 xDeepInt 在点击率预测问题中的表现如何？在极高维、稀疏的数据环境下，它是否有效且高效?
- **Q2**：不同的超参数设置会如何影响 xDeepInt 的性能?
- **Q3**：对隐式高阶特征交互进行建模，能否进一步提升 xDeepInt 的性能?

4.1 实验设置

4.1.1 数据集。 我们在三个广泛用于研究的公开真实世界数据集上评估了所提出的模型。

1. Avazu。[1] Avazu 数据集来自 2015 年的 Kaggle 竞赛。Avazu 提供了 10 天的点击数据。我们使用了 21 个特征进行建模。该数据集中的所有特征均为类别特征。

2. Criteo。[2] Criteo数据集来自 2014 年的 Kaggle 竞赛。此后，Criteo AI Lab 正式发布了该数据集，供学术研究使用。该数据集包含 13 个数值特征和 26 个类别特征。我们通过转换函数 $\lfloor Log\left(V^2\right)\rfloor$ 将所有数值特征离散化为整数，并将其视为类别特征；这一处理方式由 Criteo 竞赛的获胜团队采用。

3. 品友。[3] 品友数据集来自 2013 年的品友全球实时竞价算法竞赛。我们遵循 [32] 的数据处理步骤，并考虑全部 16 个类别特征。

对于所有数据集，我们将样本随机划分为三部分：70% 用于训练，10% 用于验证，20% 用于测试。我们还移除了出现次数少于 20 次的类别特征取值，以降低稀疏性问题。请注意，我们希望比较自动学习高阶特征交互的有效性和效率，因此不进行任何特征工程，只进行特征转换，例如数值特征分桶和类别特征频次阈值处理。

4.1.2 评估指标。我们使用曲线下面积和对数损失来评估模型的性能。

对数损失对数损失既是我们的损失函数，也是评估指标。它衡量所有样本的预测概率与真实标签之间的平均距离。

曲线下面积ROC曲线下面积（曲线下面积）衡量随机选取的正样本被模型排在随机选取的负样本之前的概率。曲线下面积只考虑正样本与负样本之间的相对顺序。曲线下面积越高，排序性能越好。

4.1.3 对比模型。我们将 xDeepInt 与以下模型进行比较：逻辑回归（逻辑回归） [18, 19], 因子分解机（因子分解机） [22],深度神经网络（普通多层感知机）、Wide & Deep [7],DeepCrossing [24], DCN（深度交叉网络） [28], PNN（同时采用内积层和外积层） [20, 21], DeepFM [8], xDeepFM [16],AutoInt [25]和 FiBiNET [13]。其中一些模型是点击率预测领域的先进模型，并在工业界得到广泛应用。

4.1.4 可复现性。 我们使用 Tensor- flow [1]实现所有模型。小批量大小为 4096，所有特征的嵌入维度均为 16。优化方面，对于所有神经网络模型，我们采用 Adam [15]，学习率设为 0.001；对于 LR 和 FM，我们采用 FTRL [18, 19]，学习率设为 0.01。正则化方面，我们对全连接层采用 L2正则化，系数为 $\lambda = 0.0001$。我们在验证数据集上对每个竞品模型的超参数进行网格搜索。

[1]https://www.kaggle.com/c/avazu-ctr-prediction [2] https://www.kaggle.com/c/criteo-display-ad-challenge [3] http://contest.ipinyou.com/

DNN、交叉网络、压缩交互网络和交互层的数量为 1 至 4。神经元数量范围为 128 至 1024。所有模型均采用早停进行训练，并且每隔 2000 个训练步数进行一次评估。

对于 xDeepInt 的超参数搜索，递归特征交互层的数量范围为 1 到 4。对于子空间数量 h，搜索值为 1、2、4、8 和 16。由于我们的嵌入维度为 16，这一范围涵盖了从完全向量级交互到完全位级交互的情况。我们使用 G-FTRL 优化器处理嵌入表，并使用 FTRL 处理 PIN 层，学习率设为 0.01。

4.2 模型性能比较（Q1）

表1：不同算法在 Criteo、Avazu 和 iPinYou 数据集上的性能比较。

	Criteo		Avazu		iPinYou	
模型	AUC	对数损失	AUC	对数损失	AUC	对数损失
LR	0.7924	0.4577	0.7533	0.3952	0.7692	0.005605
FM	0.8030	0.4487	0.7652	0.3889	0.7737	0.005576
DNN	0.8051	0.4461	0.7627	0.3895	0.7732	0.005749
Wide&Deep	0.8062	0.4451	0.7637	0.3889	0.7763	0.005589
DeepFM	0.8069	0.4445	0.7665	0.3879	0.7749	0.005609
DeepCrossing	0.8068	0.4456	0.7628	0.3891	0.7706	0.005657
DCN	0.8056	0.4457	0.7661	0.3880	0.7758	0.005682
PNN	0.8083	0.4433	0.7663	0.3882	0.7783	0.005584
xDeepFM	0.8077	0.4439	0.7668	0.3878	0.7772	0.005664
AutoInt	0.8053	0.4462	0.7650	0.3883	0.7732	0.005758
FiBiNET	0.8082	0.4439	0.7652	0.3886	0.7756	0.005679
xDeepInt	**0.8111**	**0.4408**	**0.7675**	**0.3872**	**0.7791**	**0.005565**

不同模型的整体表现列于表1。关于模型的有效性，我们有以下观察：

- 逻辑回归通常不如其他算法，这表明学习高阶特征交互对于点击率模型的性能至关重要。
- 随着模型复杂度的提高，因子分解机带来的性能提升最为显著。这体现了学习显式向量级特征交互的重要性。
- 同时结合向量级交互和位级交互的模型始终优于其他模型。这一现象表明，两类特征交互对于预测性能都至关重要，且彼此互补。
- 在所有模型中，xDeepInt 的预测性能最佳。不过，不同数据集对不同阶数的特征交互以及不同复杂度的按位特征交互各有偏好。我们模型的优异性能可能归功于 xDeepInt 能够通过调整 PIN 的深度来建模有界阶数的多项式特征交互，并通过改变子空间数量实现不同复杂度的按位特征交互。

4.3 超参数研究（Q2）

为了更深入地了解所提出的模型，我们在三个数据集上进行了实验，并比较了不同超参数设置下 xDeepInt 的若干变体。

4.3.1 网络深度。PIN 的深度决定了特征交互的阶数。表2展示了 PIN 层数变化所带来的性能变化。当层数设为 0 时，我们的模型等同于逻辑回归，且不会学习任何交互。xDeepInt 在层数约为 3 或 4 时性能最佳。在本实验中，我们将子空间数量设为 1，以禁用按位特征交互。

表2：超参数的影响：层数

	#层数	0	1	2	3	4	5
Criteo	AUC	0.7921	0.8038	0.8050	0.8057	**0.8063**	0.8061
	对数损失	0.4580	0.4477	0.4466	0.4461	**0.4452**	0.4454
Avazu	AUC	0.7536	0.7654	0.7664	**0.7675**	0.7670	0.7662
	对数损失	0.3951	0.3888	0.3879	**0.3872**	0.3875	0.3883
iPinYou	AUC	0.7690	0.7740	0.7775	**0.7791**	0.7783	0.7772
	对数损失	0.005604	0.005576	0.005569	**0.005565**	0.005580	0.005571

4.3.2 子空间数量。子空间交叉机制使所提出的模型能够控制按位交互的复杂度。表3表明，子空间交叉机制能够提升性能。在本实验中，我们将 PIN 层的数量设为 3；这通常是一个不错的选择，但并非每个数据集上的最佳设置。

表3：超参数的影响：子空间数量

	#子空间数量	1	2	4	8	16
Criteo	AUC	0.8072	0.8081	0.8089	0.8096	**0.8101**
	对数损失	0.4445	0.4435	0.4425	0.4421	**0.4418**
Avazu	AUC	0.7660	0.7668	**0.7674**	0.7672	0.7668
	对数损失	0.3880	0.3877	**0.3875**	0.3878	0.3879
iPinYou	AUC	0.7772	0.7783	**0.7788**	0.7787	0.7784
	对数损失	0.005590	0.005583	**0.005568**	0.005572	0.005580

4.3.3 激活函数。默认情况下，我们在 PIN 层的神经元上使用线性激活函数。我们还希望探究 PIN 的不同激活函数对性能的影响。表4显示，线性激活函数在 PIN 中的表现最佳。我们研究了激活函数对 Criteo数据集的影响。

表4：超参数的影响：激活函数

	AUC	对数损失
线性	**0.8111**	**0.4408**
tanh	0.8100	0.4418
Sigmoid函数	0.8082	0.4434
Softplus函数	0.8080	0.4436
Swish函数	0.8100	0.4418
relu	0.8098	0.4419
Leaky ReLU函数	0.8102	0.4415
elu	0.8099	0.4418
selu	0.8100	0.4418

4.3.4 优化器。与所有对比模型一样，我们也使用 Adam 优化器训练模型，以便与我们的 G-FTRL 和 FTRL 组合优化策略进行比较。表5显示，我们的 G-FTRL 和 FTRL 组合优化策略取得了更好的

性能。表6显示，我们的优化策略实现了更高的特征稀疏率（嵌入表中所有零嵌入向量所占的比例）和稀疏率（PIN层中零权重所占的比例），从而得到轻量级模型。值得注意的是，即使使用 Adam 优化器，xDeepInt 在所有模型中仍取得最佳预测性能，这证明了 xDeepInt 架构的有效性。

表 5：超参数的影响：优化器

数据集	模型	对数损失	AUC
Criteo	G-FTRL/FTRL	**0.4408**	**0.8111**
	Adam	0.4415	0.8105
Avazu	G-FTRL/FTRL	**0.3872**	**0.7675**
	Adam	0.3873	0.7674
iPinYou	G-FTRL/FTRL	**0.005565**	**0.7791**
	Adam	0.005583	0.7784

表6：模型稀疏性分析

数据集	特征稀疏率	权重稀疏率
Criteo	0.6506	0.1030
Avazu	0.2193	0.0448
iPinYou	0.8274	0.0627

4.4 消融研究：整合隐式交互（Q3）

在本节中，我们开展消融研究，对比所提出模型在整合和未整合隐式特征交互时的性能。

前馈神经网络被各种模型架构广泛用于学习隐式特征交互。在本实验中，我们将 xDeepInt 与三层前馈神经网络联合训练，并将组合模型命名为 xDeepInt+，以便与原始 xDeepInt 进行比较。

表 7 对比了原始 xDeepInt 和 xDeepInt+。我们发现，联合训练的前馈神经网络并未提升原始 xDeepInt 的性能。这是因为原始 xDeepInt 模型已通过子空间交叉机制学习了位级交互。因此，前馈神经网络并未带来额外的预测能力。

表 7：对比整合和未整合深度神经网络时 xDeepInt 的性能的消融研究

数据集	模型	对数损失	AUC
Criteo	xDeepInt	**0.4408**	**0.8111**
	xDeepInt+	0.4412	0.8107
Avazu	xDeepInt	**0.3872**	**0.7675**
	xDeepInt+	0.3874	0.7673
iPinYou	xDeepInt	**0.005565**	**0.7791**
	xDeepInt+	0.005581	0.7787

5 结论

本文设计了一种名为多项式交互网络（PIN）的新型网络层，用于在嵌入空间中学习高阶向量级特征交互。通过将 PIN 与子空间交叉机制相结合，我们提出的模型 xDeepInt 能够以可控的方式同时学习有界阶数的位级和向量级特征交互。我们在 PIN 层中加入残差连接，使每一层的输出都融合了低阶和高阶交互。交互阶数由 PIN 层的数量控制，而位级交互的复杂度则由子空间数量控制。此外，我们还提出了一种基于网络结构进行特征选择和交互选择的优化方法。实验结果表明，所提出的 xDeepInt 在真实世界数据集上优于现有最先进的方法。据我们所知，xDeepInt 是首个无需整合采用非线性激活函数的前馈神经网络，便能达到最先进性能的神经网络架构。

未来工作有多个方向。首先，所提出的模型仅侧重于对定长特征向量进行建模。为了对推荐系统中的历史行为和序列行为进行建模[33, 34], 我们希望让模型架构能够适用于变长特征向量。其次，我们希望将多项式交互层扩展到更多建模场景，并挖掘 PIN 在其他问题上的潜力。第三，禁用子空间交叉机制时，模型完全可解释。模型的可解释性是未来工作的另一个方向。

参考文献

[1] Martín Abadi、Paul Barham、Jianmin Chen、Zhifeng Chen、Andy Davis、Jeffrey Dean、Matthieu Devin、Sanjay Ghemawat、Geoffrey Irving、Michael Isard 等。2016。TensorFlow：面向大规模机器学习的系统。载于第 12 届 {USENIX} 操作系统设计与实现研讨会（{OSDI} 16）。265–283。[2] Alex Beutel、Paul Covington、Sagar Jain、Can Xu、Jia Li、Vince Gatto 和 Ed H Chi。2018。潜在交叉：在循环推荐系统中利用上下文。载于第十一届 ACM网络搜索与数据挖掘国际会议论文集。ACM，46–54。[3] Mathieu Blondel、Akinori Fujino、Naonori Ueda 和 Masakazu Ishihata。2016。高阶分解机。载于神经信息处理系统进展。3351–3359。[4] Mathieu Blondel、Masakazu Ishihata、Akinori Fujino 和 Naonori Ueda。2016。多项式网络与分解机：新见解与高效训练算法。arXiv 预印本 arXiv:1607.08810（2016）。[5] Patrick PK Chan、Xian Hu、Lili Zhao、Daniel S Yeung、Dapeng Liu 和 Lei Xiao。2018。基于卷积神经网络的多特征序列点击率预测。载于国际人工智能联合会议。2007–2013。[6] Chen Cheng、Fen Xia、Tong Zhang、Irwin King 和 Michael R Lyu。2014。梯度提升分解机。载于第八届 ACM推荐系统会议论文集。ACM，265–272。[7] Heng-Tze Cheng、Levent Koc、Jeremiah Harmsen、Tal Shaked、Tushar Chandra、Hrishi Aradhye、Glen Anderson、Greg Corrado、Wei Chai、Mustafa Ispir 等。2016。面向推荐系统的宽深学习。载于首届推荐系统深度学习研讨会论文集。ACM，7–10。[8] Huifeng Guo、Ruiming Tang、Yunming Ye、Zhenguo Li 和 Xiuqiang He。2017。DeepFM：一种基于因子分解机的点击率预测神经网络。arXiv 预印本 arXiv:1703.04247（2017）。[9] Kaiming He、Xiangyu Zhang、Shaoqing Ren 和 Jian Sun。2016。用于图像识别的深度残差学习。载于 IEEE计算机视觉与模式识别会议论文集。770–778。[10] Xiangnan He 和 Tat-Seng Chua。2017。面向稀疏预测分析的神经分解机。（2017）。（2017）。[11] Xiangnan He、Lizi Liao、Hanwang Zhang、Liqiang Nie、Xia Hu 和 Tat-Seng Chua。2017。神经协同过滤。载于第 26 届国际万维网会议论文集。国际万维网会议指导委员会，173–182。[12] Xinran He、Junfeng Pan、Ou Jin、Tianbing Xu、Bo Liu、Tao Xu、Yanxin Shi、Antoine Atallah、Ralf Herbrich、Stuart Bowers 等。2014。Facebook 广告点击预测的实践经验。载于第八届在线广告数据挖掘国际研讨会论文集。ACM，1–9。[13] Tongwen Huang、Zhiqi Zhang 和 Junlin Zhang。2019。FiBiNET：结合特征重要性与双线性特征交互进行点击率预测。arXiv 预印本 arXiv:1905.09433（2019）。[14] Yuchin J uan、Yong Zhuang、Wei-Sheng Chin 和 Chih-Jen Lin。2016。用于点击率预测的域感知分解机。载于第十届 ACM推荐系统会议论文集。ACM，43–50。[15] Diederik P Kingma 和 Jimmy Ba。2014。Adam：一种随机优化方法。arXiv 预印本 arXiv:1412.6980（2014）。[16] Jianxun Lian、Xiaohuan Zhou、Fuzheng Zhang、Zhongxia Chen、Xing Xie 和 Guangzhong Sun。2018。xDeepFM：结合显式与隐式特征交互的推荐系统模型。载于第 24 届 ACM SIGKDD知识发现与数据挖掘国际会议论文集。ACM，1754–1763。[17] Xiaoliang Ling、Weiwei Deng、Chen Gu、Hucheng Zhou、Cui Li 和 Feng Sun。2017。必应搜索广告点击预测的模型集成。载于第 26 届国际万维网会议伴随会议论文集。国际万维网会议指导委员会，689–698。[18] H Brendan McMahan。2011。正则化领导者跟随算法与镜像下降：等价性定理和 L1 正则化。（2011）。[19] H Brendan McMahan、Gary Holt、David Sculley、Michael Young、Dietmar Ebner、Julian Grady、Lan Nie、Todd Phillips、Eugene Davydov、Daniel Golovin 等。2013。广告点击预测：来自一线的观察。载于第 19 届 ACM SIGKDD知识发现与数据挖掘国际会议论文集。ACM，1222–1230。[20] Yanru Qu、Han Cai、Kan Ren、Weinan Zhang、Yong Yu、Ying Wen 和 Jun Wang。2016。用于用户响应预测的基于乘积的神经网络。载于 2016 年 I EEE 第 16 届国际数据挖掘会议（ICDM）。IEEE，1149–1154。[21] Yanru Qu、Bohui Fang、Weinan Zhang、Ruiming Tang、Minzhe Niu、Huifeng Guo、Yong Yu 和 Xiuqiang He。2018。基于乘积的神经网络：面向多域类别数据的用户响应预测。ACM信息系统汇刊（TOIS）37，1（2018），5。[22] Steffen Rendle。2010。分解机。载于 2010 年 IEEE 国际数据挖掘会议。IEEE，995–1000。[23] Matthew Richardson、Ewa Dominowska 和 Robert Ragno。2007。点击预测：估计新广告的点击率。载于第 16 届 国际万维网会议. ACM，521–530. [24] Ying Shan、T Ryan Hoens、Jian Ji ao、Haijing Wang、Dong Yu 和 JC Mao。2016。Deep crossing：无需手工构造组合特征的 Web规模建模。载于第*22*届 *ACM SIGKDD*知识发现与数据挖掘国际会议论文集。ACM，255–262。[25] Weiping Song、Chence Shi、Zhiping Xiao、Zhijian Duan、Yewen Xu、Ming Zhang 和 Jian Tang。2018。AutoInt：通过自注意力神经网络实现自动特征交互学习。*arXiv* 预印本 *arXiv:1810.11921* (2018)。[26] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。2017。Attention is all you need。载于神经信息处理系统进展。5998–6008。[27] Andreas Veit、Michael J Wilber 和 Serge Belongie。2016。残差网络的表现类似于相对浅层网络的集成。载于神经信息处理系统进展。550–558。[28] Ruoxi Wang、Bin Fu、Gang Fu 和 Mingliang Wang。2017。用于广告点击预测的深度交叉网络。载于*ADKDD'17*论文集。ACM，12。[29] Jun Xiao、Hao Ye、Xiangnan He、Hanwang Zhang、Fei Wu 和 Tat-Seng Chua。2017。注意力因子分解机：通过注意力网络学习特征交互的权重。*arXiv* 预印本 *arXiv:1708.04617* (2017)。[30] Shuai Zhang、Lina Yao、Aixin Sun 和 Yi Tay。2019。基于深度学习的推荐系统：综述与新视角。*ACM*计算调查 *(CSUR)* 52, 1 (2019), 5。[31] Weinan Zhang、Tianming Du 和 Jun Wang。2016。基于多域类别数据的深度学习。载于欧洲信息检索会议。Springer，45–57。[32] Weinan Zhang、Shuai Yuan、Jun Wang 和 Xuehua Shen。2014。基于品友数据集的实时竞价基准测试。*arXiv* 预印本 *arXiv:1407.7073* (2014)。[33] Guorui Zhou、Na Mou、Ying Fan、Qi Pi、Weijie Bian、Chang Zhou、Xiaoqiang Zhu 和 Kun Gai。2018。用于点击率预测的深度兴趣演化网络。*arXiv* 预印本 *arXiv:1809.03672* (2018)。[34] Guorui Zhou、Xiaoqiang Zhu、Chenru Song、Ying Fan、Han Zhu、Xiao Ma、Yanghui Yan、Junqi Jin、Han Li 和 Kun Gai。2018。用于点击率预测的深度兴趣网络。载于第*24*届 *ACM SIGKDD*知识发现与数据挖掘国际会议论文集。ACM，1059–1068。[35] Jie Zhu、Ying Shan、JC Mao、Dong Yu、Holakou Rahmanian 和 Yi Zhang。2017。深度嵌入森林：基于深度嵌入特征的森林服务。载于第*23*届 *ACM SIGKDD*知识发现与数据挖掘国际会议论文集。ACM，1703–1711。