

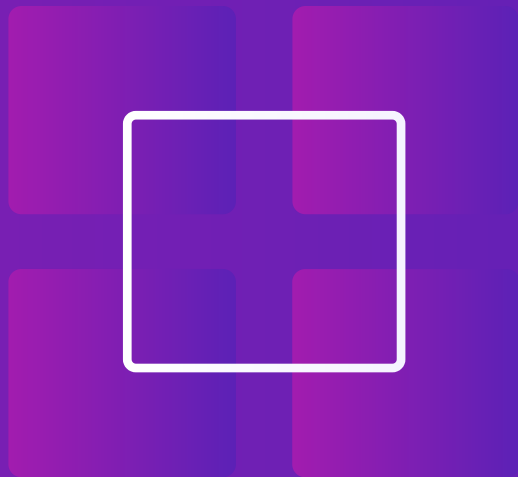


Fi-GNN: Modeling Feature Interactions via Graph Neural Networks for CTR Prediction

CIKM 19 中国科学院大学
乔梁 2022.10.28



- 一、摘要
- 二、引言
- 三、相关工作
- 四、Fi-GNN
- 五、结论
- 六、学习收获





一、摘要



CTR

背景：点击率 (CTR) 预测是web应用程序 (例如在线广告和推荐系统) 中的一项重要任务，其特征通常是多字段形式的。此任务的关键是对不同特征字段之间的特征交互进行建模。

之前的方法：基于深度学习的模型遵循一个通用范例: 原始的稀疏输入多域特征首先被映射到稠密嵌入向量中，然后简单地串联在一起以馈送到 DNN 或其他专门设计的网络中以学习高阶特征交互。但是，特征字段的简单非结构化组合将不可避免地限制以足够灵活和明确的方式对不同字段之间的复杂交互进行建模的能力。

本文工作：在这项工作中，我们建议直观地表示图结构中的多字段特征，其中每个节点对应于一个特征字段，并且不同的字段可以通过边进行交互。因此，可以将对特征交互进行建模的任务转换为对相应图形上的节点交互进行建模。为此，我们设计了一种新颖的模型特征交互图神经网络 (Fi-GNN)。利用图的强大代表能力，我们提出的模型不仅可以灵活、显式地对复杂的特征交互进行建模，而且可以为 CTR 预测提供良好的模型解释。在两个真实世界数据集上的实验结果表明，它优于最新技术。



二、引言



特征交互方法

点击率预测的目标是预测用户点击广告或项目的概率，这对于许多web应用程序 (例如在线广告和推荐系统) 至关重要。对复杂的特征交互进行建模在CTR预测的成功中起着核心作用。

二阶特征交互

- 因式分解机 (FM) 是一个众所周知的模型，旨在从向量内积中学习二阶特征相互作用。
- 场感知因式分解机 (FFM) 进一步考虑了场信息，并引入了场感知嵌入。遗憾的是，这些基于FM的模型只能对二阶交互进行建模，而线性建模限制了其代表性。

多阶特征交互

- 因式分解机支持的神经网络 (FNN)
- 神经因式分解机 (NFM)
- Wide & Deep
- DeepFM

缺点：这些基于DNN的模型以一种隐式的方式学习高阶特征交互，这缺乏良好的模型解释。



特征交互方法

引入专门设计的网络

一些模型试图通过引入专门设计的网络来显式学习高阶交互。

- Deep&Cross 引入了交叉网络 (CrossNet)
- xDeepFM 引入了压缩交互网络 (CIN)

缺点：尽管如此，我们认为它们仍然不够有效和明确，因为它们仍然遵循将特征字段组合在一起以对其交互进行建模的一般范式。简单的非结构化组合将不可避免地限制以灵活和明确的方式对不同特征字段之间的复杂交互进行建模的能力。



本文工作

我们考虑了多字段特征的结构。具体来说，我们在名为特征图的图结构中表示多字段特征。

图中的每个节点对应一个特征字段，不同的字段可以通过边进行交互。因此，可以将对特征字段之间的复杂交互进行建模的任务转换为对特征图上的节点交互进行建模。

为此，我们设计了一种基于图神经网络 (GNN) 的新型模型`特征交互图神经网络 (Fi-GNN)`，该模型能够以灵活，显式的方式对复杂的节点 (特征) 交互进行建模。在Fi-GNN中，节点将通过与邻居通信节点状态进行交互，并以循环的方式更新自身。在每个时间步长，模型与邻居的交互更深。因此，交互步骤的数量等于特征交互的顺序。此外，Fi-GNN可以学习反映不同特征交互的重要性的边缘权重和反映每个特征字段对最终CTR预测的重要性的节点权重，这可以提供很好的解释。总体而言，我们提出的模型可以以`显式，灵活的方式`对复杂的特征交互进行建模，并提供`良好的模型解释`。



本文贡献

我们的贡献可以归纳为三个方面：

1. 我们指出了现有方法的局限性，这些方法将多字段特征视为特征字段的非结构化组合。为此，我们首次提出在图结构中表示多字段特征。
2. 我们设计了一种新颖的模型特征交互图神经网络 (Fi-GNN)，以更加灵活和明确的方式对图结构特征上的特征域之间的复杂交互进行建模。
3. 在两个方面进行了广泛的实验真实世界的数据集表明，我们提出的方法不仅可以超越现有方法，而且可以提供良好的模型解释。



三、相关工作



特征交互

一阶特征交互

LR是一种线性方法，只能对单个特征的线性组合进行一阶交互建模。

二阶特征交互

FM FFM AFM

高阶

- 因式分解-机器支持的神经网络 (FNNs) 在应用DNN之前使用预先训练的 FM 进行场嵌入。
- PNN 通过在场嵌入层和 DNN 层之间引入 Product 层来对二阶和高阶相互作用进行建模。
- NFM 在嵌入层和 DNN 层之间有一个双向交互池化层，用于对二阶交互进行建模，但是随后的操作是求和而不是PNN中的级联。
- 混合体系结构共同建模二阶和高阶交互。Wide & Deep 和DeepFM 包含用于建模低阶交互的 Wide 部分，以及用于建模高阶交互的 Deep 部分。但是，所有这些利用DNN的方法都以隐式的，按位方式学习高阶特征交互，因此缺乏良好的模型解释能力。



特征交互

专门设计的网络

- Deep & Cross 引入了一个CrossNet，该CrossNet在位级别上具有特征的外部乘积。
- xDeepFM 引入了CIN在矢量水平上获取外积。

尽管如此，它们仍然不能解决最根本的问题，那就是将域嵌入向量串联在一起。特征字段的简单非结构化组合将不可避免地限制以灵活和显式方式对不同字段之间的复杂交互进行建模的能力。为此，我们提出在图结构中表示多字段特征，其中每个节点表示一个字段，并且不同的特征字段可以通过边进行交互。因此，我们可以对图形上不同特征域之间的灵活交互进行建模。



图神经网络GNN

图是一种对一组对象 (节点) 及其关系 (边) 进行建模的数据结构。近年来, 用机器学习分析图形的研究受到了越来越多的关注。因为图形的巨大代表性。早期的工作通常会将图结构的数据转换成序列结构的数据来处理。

图神经网络 GNN 中的节点通过聚合来自邻域的信息并更新其隐藏状态来与邻居进行交互。如今, GNN 的许多变体都提出了各种聚合器和更新程序。这里我们只介绍一些有代表性的经典方法。

- 门控图神经网络 (GGNN) 使用GRU 作为更新器。
- 图卷积网络 (GCN) 考虑了图的频谱结构, 并利用了卷积聚合器。
- GraphSAGE 考虑了空间信息。介绍了三种聚合器: 平均聚合器、LSTM聚合器和池化聚合器。
- 图注意网络 (GAT) 将注意机制纳入传播步骤。



图神经网络GNN

由于具有令人信服的性能和高度的可解释性，GNN已成为一种广泛应用的图分析方法。

GNN 的应用：

- 神经机器翻译
- 语义分割
- 图像分类
- 情况识别
- 推荐

GNN适用于本质上对图结构特征上的节点交互建模。在这项工作中，我们提出了一种基于GGNN的模型Fi-GNN，用于在图结构特征上对特征交互进行建模，以进行CTR预测。



四、Fi-GNN



图1: 本文提出的方法框架

首先将输入的原始多域特征向量通过嵌入层转换为域嵌入向量, 并表示为特征图, 然后将其馈送到Fi - GNN中, 以对特征交互进行建模。在Fi-GNN的输出上应用注意力层, 以预测点击率 y 。

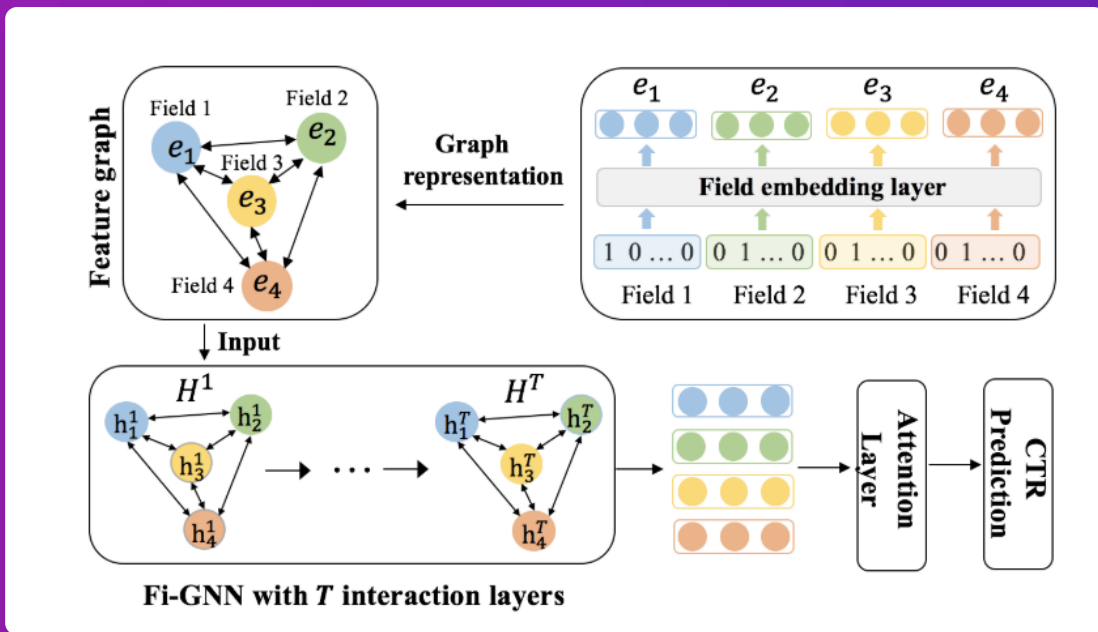
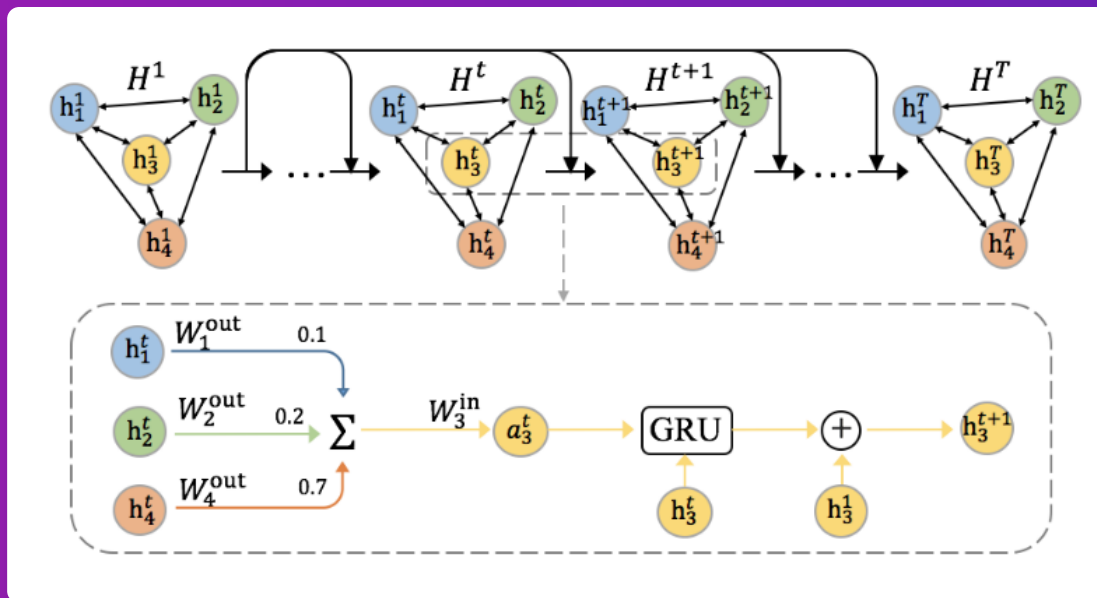




图2: Fi-GNN框架

节点与邻居交互并以循环方式更新其状态。在每个交互步骤中，每个节点将首先聚合来自邻居的转换状态信息，然后通过GRU和残差连接根据聚合的信息和历史更新其状态。





问题定义

假设训练数据集由 m 个字段分类特征 (m 是数字特征字段) 和指示用户单击行为的关联标签 $y \in \{0, 1\}$ 组成。CTR预测的任务是预测输入 m 个域特征的 y , 该特征估计用户点击的概率。该任务的关键是对不同特征字段之间的复杂交互进行建模。



嵌入层

多字段分类特征 x 通常是稀疏的，并且维度很大。我们将每个字段表示为独热编码向量，然后将其嵌入到一个稠密向量中，称为字段嵌入向量。如，电影 {语言: 英语，类型: 小说，导演: 克里斯托弗·诺兰，主演: 莱昂纳多·迪卡普里奥} 首先通过one-hot编码转换为高维稀疏特征：

$$\underbrace{[1, 0, \dots, 0]}_{\text{Language}}, \underbrace{[0, 1, \dots, 0]}_{\text{Genre}}, \underbrace{[0, 1, \dots, 0]}_{\text{Director}}, \underbrace{[0, 1, \dots, 0]}_{\text{Starring}}$$

然后将域感知嵌入层应用于独热向量，以将其嵌入到低维，稠密的实值域嵌入向量中，如图1所示。同样，可以获得 m 个域特征的域嵌入向量：

$$\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \dots, \mathbf{e}_m]$$

其中 $\mathbf{e}_i \in \mathbb{R}^D$ 表示域 i 的嵌入向量， D 表示域嵌入向量的维数。



特征图

与以前简单地将域嵌入向量连接在一起并将其输入到设计模型中以学习特征交互的工作不同，我们以图形结构表示它们。

特别地，我们将每个输入的多字段特征表示为特征图 $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ ，其中每个节点 $n_i \in \mathcal{N}$ 对应于特征字段 i ，并且不同的字段可以通过边相互作用，因此 $|\mathcal{N}| = m$ 。由于每两个字段应该相互作用，这是一个加权的全连通图，而边权重反映了不同特征交互的重要性。因此，可以将对特征交互进行建模的任务转换为对特征图上的节点交互进行建模。



特征交互图神经网络

Fi-GNN旨在对特征图上的节点交互进行建模，该特征图上基于GGNN。它能够以灵活和明确的方式对特征交互进行建模。在Fi-GNN中，每个节点都与隐藏状态向量 \mathbf{h}_i^t 相关联，并且图的状态由这些节点状态组成

$$\mathbf{H}^t = [\mathbf{h}_1^t, \mathbf{h}_2^t, \mathbf{h}_3^t, \dots, \mathbf{h}_m^t]$$

其中 t 表示交互步骤。如图2所示，节点以循环方式交互和更新其状态。在每个交互步骤中，节点将转换后的状态信息与邻居聚合，然后通过GRU和残差连接根据聚合后的信息和历史更新其节点状态。接下来，将详细介绍Fi-GNN的细节。



特征交互图神经网络

初始节点状态。初始节点状态是输入特征图的状态，即字段嵌入向量。特别地，域 i 的嵌入向量 \mathbf{e}_i 被用作相应节点 n_i 的初始状态向量：

$$\begin{aligned}\mathbf{H}^1 &= \mathbf{E}, \\ \mathbf{h}_i^1 &= \mathbf{e}_i, i \in [1, 2, \dots, m].\end{aligned}$$

因此, $\mathbf{h} \in \mathbb{R}^D$, $\mathbf{H} \in \mathbb{R}^{m \times D}$ 。



特征交互图神经网络

状态聚合。在交互步骤 t ，每个节点将聚合来自邻居的状态信息。形式上，节点 n_i 的聚合信息是其邻居变换状态信息的总和。

$$\mathbf{a}_i^t = \sum_{n_j \rightarrow n_i \in \mathcal{E}} \mathbf{A}[n_j, n_i] \mathbf{W}_p \mathbf{h}_j^{t-1} \quad (3)$$

其中 \mathbf{W}_p 是变换函数。 $\mathbf{A} \in \mathbb{R}^{m \times m}$ 是包含边权的邻接矩阵。例如， $\mathbf{A}[n_j, n_i]$ 是从节点 n_j 到 n_i 的边的权重，可以反映出它们相互作用的重要性。显然，变换函数和邻接矩阵决定了节点的相互作用。由于每个边上的相互作用应该不同，因此我们的目标是实现边方向的相互作用，这需要每个边的唯一权重和转换函数。



特征交互图神经网络

(1) 注意力边权重。传统GNN模型中的邻接矩阵通常是二进制形式，即只包含0和1。它只能反映节点的连接关系，而不能反映节点关系的重要性。为了推断不同节点之间交互的重要性，通过注意力机制来学习边权重。特别地，从节点 n_i 到节点 n_j 的边的权重是用它们的初始节点状态，即相应的域嵌入向量来计算的。

$$w(n_i, n_j) = \frac{\exp(\text{LeakyRelu}(\mathbf{W}_w [\mathbf{e}_i \parallel \mathbf{e}_j]))}{\sum_k \exp(\text{LeakyRelu}(\mathbf{W}_w [\mathbf{e}_i \parallel \mathbf{e}_k]))} \quad (4)$$

其中 $\mathbf{W}_w \in \mathbb{R}^{2D}$ 是权重矩阵， \parallel 是级联运算。softmax函数用于使权重在不同节点之间易于比较。因此，邻接矩阵是，

$$A[n_i, n_j] = \begin{cases} w(n_i, n_j), & \text{if } i \neq j, \\ 0, & \text{else.} \end{cases}$$

由于边权重反映了不同交互的重要性，Fi-GNN 可以很好地解释输入实例的不同特征字段之间的关系。



特征交互图神经网络

(2) 边变换。如前所述，所有边上的固定变换函数无法对灵活的相互作用进行建模，并且每个边的唯一变换至关重要。尽管如此，我们的图是具有大量边的完整图。简单地为每个边分配唯一的转换权重将消耗太多的参数空间和运行时间。为了降低时间和空间复杂度并实现边缘变换，我们将输出矩阵 $\mathbf{W}_{\text{out}}^i$ 和输入矩阵 \mathbf{W}_{in}^i 分配给与 (4) 相似的每个节点 n_i 。如图2所示，当节点 n_i 将其状态信息发送给节点 n_j 时，状态信息将首先由其输出矩阵 $\mathbf{W}_{\text{out}}^i$ 进行变换，然后在 n_j 接收到它之前由节点 n_j 的输入矩阵 \mathbf{W}_{in}^j 进行变换。因此，从节点 n_i 到节点 n_j 的边 $n_i \rightarrow n_j$ 的变换函数可以写为：

$$\mathbf{W}_p^{n_i \rightarrow n_j} = \mathbf{W}_{\text{out}}^i \mathbf{W}_{\text{in}}^j$$

同样，边 $n_j \rightarrow n_i$ 从节点 n_j 到节点 n_i 的变换函数为：

$$\mathbf{W}_p^{n_j \rightarrow n_i} = \mathbf{W}_{\text{out}}^j \mathbf{W}_{\text{in}}^i.$$

相应地，等式3可以重写为，

$$\mathbf{a}_i^t = \sum_{n_j \rightarrow n_i \in \mathcal{E}} \mathbf{A}[n_j, n_i] \mathbf{W}_{\text{out}}^j \mathbf{W}_{\text{in}}^i \mathbf{h}_j^{t-1} + \mathbf{b}_p$$

这样，参数的数量与节点的数量成正比，而不是多条边，这大大降低了空间和时间复杂度，同时实现了边交互。



特征交互图神经网络

状态更新。聚合状态信息后，节点将通过GRU和残差连接更新状态向量。

(1) 通过GRU进行状态更新。在传统的GGNN中，基于 i 处的聚合状态信息及其最后一步的状态，通过GRU更新节点 n_i 的状态向量。形式上，

$$\mathbf{h}_i^t = GRU(\mathbf{h}_i^{t-1}, \mathbf{a}_i^t) \quad (9)$$

$$\mathbf{z}_i^t = \sigma(\mathbf{W}_z \mathbf{a}_i^t + \mathbf{U}_z \mathbf{h}_i^{t-1} + \mathbf{b}_z)$$

$$\mathbf{r}_i^t = \sigma(\mathbf{W}_r \mathbf{a}_i^t + \mathbf{U}_r \mathbf{h}_i^{t-1} + \mathbf{b}_r)$$

$$\tilde{\mathbf{h}}_i^t = \tanh(\mathbf{W}_h \mathbf{a}_i^t + \mathbf{U}_h (\mathbf{r}_i^t \odot \mathbf{h}_i^{t-1}) + \mathbf{b}_h)$$

$$\mathbf{h}_i^t = \tilde{\mathbf{h}}_i^t \odot \mathbf{z}_i^t + \mathbf{h}_i^{t-1} \odot (1 - \mathbf{z}_i^t)$$

其中， \mathbf{W}_z , \mathbf{W}_r , \mathbf{W}_h , \mathbf{b}_z , \mathbf{b}_r , \mathbf{b}_h 是更新函数门控递归单元 (GRU) 的权重和偏差。 \mathbf{z}_i^t 和 \mathbf{r}_i^t 分别为更新 i 门向量和重置门向量。



特征交互图神经网络

(2) 通过残差连接更新状态。已经证明，将低阶和高阶相互作用结合在一起是有效的。因此，我们引入了额外的残差连接来更新节点状态以及GRU，这可以促进低阶特征重用和梯度反向传播。因此，方程(9) 可以改写为

$$\mathbf{h}_i^t = GRU(\mathbf{h}_i^{t-1}, \mathbf{a}_i^t) + \mathbf{h}_i^1$$



注意力层

经过 T 个传播步骤，我们可以得到节点状态

$$\mathbf{H}^T = [\mathbf{h}_1^T, \mathbf{h}_2^T, \dots, \mathbf{h}_m^T]$$

由于节点已与其 T 阶邻居进行了交互，因此对 T 阶特征交互进行了建模。我们需要一个图级的输出来预测CTR。

注意力节点权重：每个字段节点的最终状态已捕获全局信息。换句话说，这些归档节点是邻域感知的。在这里，我们分别预测每个字段的最终状态的分数，并通过注意力机制对它们进行总结，以衡量它们对整体预测的影响。形式上，可以通过两个多层感知机分别估计每个节点 n_i 的预测得分及其注意力节点权重，

$$\begin{aligned}\hat{y}_i &= MLP_1(\mathbf{h}_i^p) \\ a_i &= MLP_2(\mathbf{h}_i^p)\end{aligned}$$



注意力层

总体预测是所有节点的总和:

$$\hat{y} = \sum_{i=1}^m a_i \hat{y}_i$$

直观地, MLP_1 用于建模每个领域的预测得分, 知道全局信息和 MLP_2 用于对每个字段的权重 (即字段对整体预测的影响的重要性) 进行建模。



训练

我们的损失函数是对数损失，其定义如下：

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

其中N是训练样本的总数，i为训练样本索引。使用RMSProp [28] 通过最小化对数损失来更新参数。大多数CTR数据集的正负样本比例不均衡，这会误导预测结果。为了平衡比例，在训练过程中，我们在每批中随机选择相等数量的正负样本。



模型分析

与以前的CTR模型的比较

- 以前的基于深度学习的CTR模型以一般范式对高阶交互进行建模: 原始稀疏输入多场特征首先被映射到稠密域嵌入向量中, 然后简单地串联在一起并馈入深度神经网络 (DNN) 或其他专门设计的网络来学习高阶特征交互。
- 特征字段的简单非结构化组合不可避免地限制了以足够灵活和明确的方式对不同字段之间的复杂交互进行建模的能力。这样, 无论使用的网络多么复杂, 不同领域之间的交互都以固定的方式进行。
- 此外, 他们缺乏良好的模型解释。



模型分析

与以前的CTR模型的比较`

- 由于我们在图结构中表示多字段特征，因此我们提出的模型Fi-GNN能够以节点交互的形式对不同字段之间的交互进行建模。
- 与以前的CTR模型相比，Fi-GNN可以通过灵活的边交互功能对复杂的特征交互进行建模，这更加有效和明确。此外，可以在Fi-GNN中学习反映不同交互作用重要性的边缘权重，这为CTR预测提供了良好的模型解释。
- 实际上，如果边权均为1，并且每个边上的变换矩阵相同，我们的模型Fi-GNN 退化为 FM 。利用GNN的强大功能，我们可以在不同的特征领域应用灵活的交互。



模型分析

与以前的GNNModels的比较

我们提出的模型Fi-GNN是基于GGNN设计的，我们主要做了两个改进：

1. 我们通过注意力边权重和边变换实现边交互；
2. 我们引入了额外的残差连接和GRU来更新状态，这可以帮助恢复低阶信息。如前所述，GNN中每个边缘上的节点交互取决于边权重和边上的变换函数。传统的GGNN使用无法反映关系重要性的二进制边权重，并在所有边上使用固定的变换函数。相反，我们提出的Fi-GNN可以通过注意力边权重和边变换函数对边交互进行建模。当交互顺序高时，节点状态趋于平滑，即所有节点的状态趋于相似。残差连接可以通过添加初始节点状态来帮助标识节点。





结论

- 在本文中，我们指出了先前的CTR模型的局限性，这些模型将多字段特征视为非结构化的组合特征字段。
- 为了克服这些限制，建议首次在图结构中表示多字段特征，其中每个节点对应于一个特征字段，并且不同的字段可以通过边进行交互。因此，建模特征交互可以转换为在图上建模节点交互。
- 为此，我们设计了一种新颖的模型Fi-GNN，该模型能够以灵活，显式的方式对特征字段之间的复杂交互进行建模。
- 总体而言，我们提出了一种新的CTR预测范式：在图结构中表示多字段特征，并将建模特征交互的任务转换为在图上建模节点交互，这可能会激励这一行的未来工作。



六、学习收获



学习收获

GNN 在特征交互中的使用。

Thank you