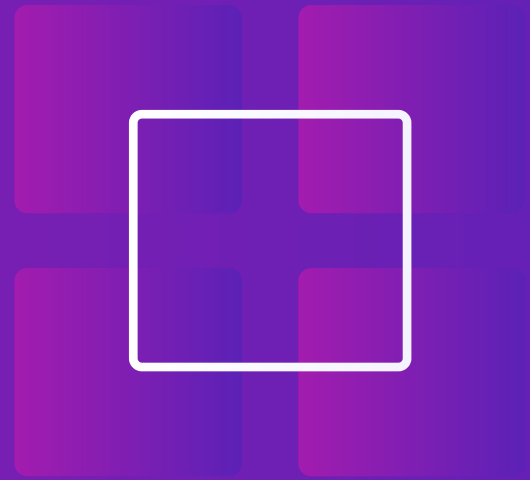
AutoInt: Automatic Feature Interaction Learning via Self-Attentive Neural Networks

乔梁 2022.8.5



一、引言

CTR 问题的挑战

点击率（CTR）预测旨在预测用户点击广告或项目的概率，对许多在线应用程序（如在线广告和推荐系统）至关重要。这个问题非常具有挑战：

1. 输入特征（例如，用户id、用户年龄、项目id、项目类别）通常是稀疏和高维的；
2. 有效的预测依赖于高阶组合特征（也称交叉特征），这些特征对于领域专家手工制作非常耗时，并且不可能被枚举

因此，人们一直在努力寻找稀疏和高维原始特征的低维表示及其有意义的组合。

之前方法的局限性

将多项式回归模型与因子分解技术相结合的因子分解机（FM）被用于建模特征交互，并已被证明对各种任务有效。然而，由于其多项式拟合时间的限制，它只能有效地建模低阶特征交互，而无法捕捉高阶特征交互。最近，人们提出了许多基于深度神经网络的工作来模拟高阶特征交互。具体来说，通常使用多层非线性神经网络来捕捉高阶特征交互。

然而，这种方法有两个局限性。

1. 首先，全连接神经网络在学习乘法特征交互方面效率低下；
2. 其次，由于这些模型以隐式方式学习特征交互，因此它们缺乏关于哪些特征组合有意义的良好解释。

因此，我们正在寻找一种能够显式建模不同阶特征组合的方法，将整个特征表示到低维空间中，同时提供良好的模型解释性。

本文工作

在本文中，我们提出了一种基于多头自注意力机制的方法。我们提出的方法学习稀疏和高维输入特征的有效低维表示，并且适用于分类和/或数字输入特征。具体来说，分类特征和数值特征首先嵌入到低维空间中，这降低了输入特征的维数，同时允许不同类型的特征通过向量算法（例如，求和和内积）相互作用。

然后，我们提出了一种新的交互层来促进不同特征之间的交互。在每个交互层中，每个特征都可以与所有其他特征交互，并且能够自动识别相关特征，通过多头注意机制形成有意义的高阶特征。此外，多头机制将特征投影到多个子空间，因此可以在不同子空间中捕获不同的特征交互。这种交互层模拟了特征之间的一步交互。通过堆叠多个交互层，我们能够对不同顺序的特征交互进行建模。实际上，残差连接被添加到交互层，这允许组合不同顺序的特征组合。我们使用注意力机制来测量特征之间的相关性，这提供了良好的模型解释性。

本文贡献

在本文中，我们做出了以下贡献：

- 我们建议研究显式学习高阶特征交互的问题，同时找到对该问题具有良好解释性的模型。
- 我们提出了一种基于自注意力神经网络的新方法，该方法可以自动学习高阶特征交互，并有效处理大规模高维稀疏数据。
- 我们在几个真实数据集上进行了广泛的实验。在连续时间序列预测任务上的实验结果表明，我们提出的方法不仅优于现有的最先进的预测方法，而且提供了良好的模型解释性。

二、相关工作

本工作涉及的技术

我们的工作涉及三个方面：

1. 推荐系统和在线广告中的点击率预测；
2. 学习特征交互的技术；
3. 深度学习文献中的自注意力机制和残差网络。

CTR

预测点击率对许多互联网公司来说很重要，不同的公司开发了各种系统。例如，谷歌为推荐系统开发了 Wide&Deep 学习系统，该系统结合了线性浅层模型和深度模型的优点。该系统在应用推荐中取得了显著的性能。这个问题在学术界也受到了很多关注。例如，Shan等人[31]提出了一种上下文感知的CTR预测方法，该方法分解了三向<user, ad, context>张量。Oentaryo等人[24]开发了层次重要性感知因子分解机来模拟广告的动态影响。

学习特征交互

学习特征交互是一个基本问题，因此在文献中进行了广泛研究。

1. 因子分解机（FM），它主要用于捕捉一阶和二阶特征交互，并已被证明对推荐系统中的许多任务有效。
2. 因子分解机的不同变体。例如，场感知因子分解机（FFM）模拟了不同场特征之间的细粒度交互。GBFM[7]和AFM[40]考虑了不同二阶特征相互作用的重要性。然而，所有这些方法都侧重于`低阶特征交互`的建模。
3. 最近有一些研究模拟了高阶特征交互。例如，NFM将深度神经网络堆叠在二阶特征交互的输出之上，以建模高阶特征。类似地，PNN、FNN、DeepCrossing、Wide&Deep、DeepFM利用前馈神经网络对高阶特征交互进行建模。然而，所有这些方法都以隐式方式学习高阶特征交互，因此`缺乏良好的模型解释性`。
4. 有三个模型以显式的方式学习特征交互。
    - Deep&Cross和xDeepFM分别在位和矢量级别上取特征的外积。尽管它们执行显式特征交互，但解释哪些组合是有用的并不是小事。
    - 其次，一些基于树的方法结合了基于嵌入的模型和基于树的模型的能力，但必须将训练过程分解为多个阶段。
    - HOFM提出了高阶因子分解机的有效训练算法。然而，HOFM需要太多的参数，只能实际使用其低阶（通常小于5）形式。

与现有工作不同的是，我们以端到端的方式显式地建模具有注意力机制的特征交互，并通过可视化探索学习到的特征组合。

注意力和残差网络

我们提出的模型利用了深度学习文献中的最新技术：注意力和残差网络。

注意力首先是在神经机器翻译的背景下提出的，并已被证明在各种任务中有效，如问答、文本摘要和推荐系统。Vaswani等人进一步提出了多头自注意力，以模拟机器翻译中单词之间的复杂依赖关系。

残差网络在ImageNet竞赛中取得了最先进的性能。由于残差连接可以简单地形式化为$y = F(x) + x$，鼓励通过区间层的梯度流，因此它成为一种用于训练深度神经网络的流行网络结构。

三、问题定义

我们首先将点击率预测问题正式定义如下：

定义1：（点击率预测）令 $\mathbf{x} \in \mathbb{R}^n$ 表示用户 u 的特征和项目 v 的特征的串联，其中分类特征用独热编码表示，n 是串联特征的维数。点击率预测问题旨在根据特征向量 x 预测用户 u 点击项目 v 的概率。

CTR预测的一个简单解决方案是将 x 作为输入特征，并部署现成的分类器，如逻辑回归。然而，由于原始特征向量 x 非常稀疏和高维，模型很容易过拟合。

因此，需要

- 在`低维连续空间中表示原始输入特征`；
- 利用`高阶组合特征`来产生良好的预测性能。

具体来说，我们将高阶组合特征定义如下：

定义2：（p 阶组合特征）给定输入特征向量 $\mathbf{x} \in \mathbb{R}^n$，$p$ 阶组合特征被定义为 $g(x_{i1}, \ldots, x_{ip})$，其中每个特征来自一个不同的字段，$p$ 是涉及的特征字段的数量，$g(\cdot)$ 是一个非加性组合函数，例如乘法和外积。例如，$x_{i1} \times x_{i2}$ 是涉及 x_{i1} 和 x_{i2} 的二阶组合特征。

传统上，有意义的高阶组合特征是由领域专家手工制作的。然而，这非常耗时，很难推广到其他领域。此外，手工制作所有有意义的高阶特征几乎是不可能的。因此，我们的目标是开发一种能够自动发现有意义的高阶组合特征的方法，同时将所有这些特征映射到低维连续空间。

形式上，我们将问题定义如下：

定义3：（问题定义）给定输入特征向量 $\mathbf{x} \in \mathbb{R}^n$，对于点击率预测，我们的目标是学习 $\mathbf{x}$ 的低维表示，它模拟高阶组合特征。

四、AutoInt

方法概述

该方法的目标是将原始稀疏高维特征向量映射到低维空间，同时对高阶特征交互进行建模。

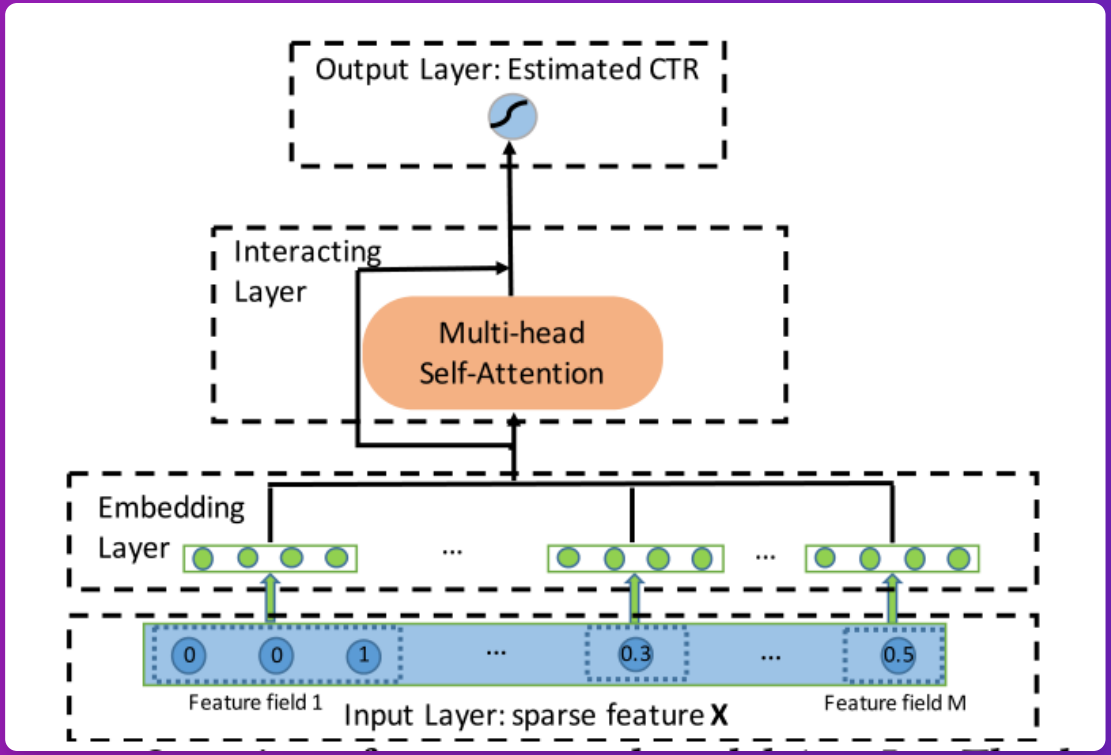
如图1所示，我们提出的方法将稀疏特征向量 $\mathbf{x}$ 作为输入，然后是一个嵌入层，该嵌入层将所有特征（即分类特征和数值特征）投影到同一低维空间。

接下来，我们将所有字段嵌入到一个新的交互层中，该层被实现为一个多头自注意力神经网络。对于每个交互层，通过注意力机制组合高阶特征，并可以使用多头机制评估不同类型的组合，将特征映射到不同的子空间。

通过堆叠多个交互层，可以模拟不同顺序的组合特征。最终交互层的输出是输入特征的低维表示，它对高阶组合特征进行建模，并进一步用于通过 $sigmoid$ 函数估计点击率。

图1：AutoInt 模型概述

嵌入层和交互层的细节分别如图2和图3所示。



输入层

我们首先将用户配置文件和项目属性表示为稀疏向量，这是所有字段的串联。具体来说，

$$\mathbf{x} = [\mathbf{x}_1; \mathbf{x}_2; \ldots; \mathbf{x_M}]$$

其中 M 是总特征字段的数量，$\mathbf{x}_i$ 是第 i 个字段的特征表示。如果第 i 个字段是分类的，则 $\mathbf{x}_i$ 是独热向量。如果第 i 个字段是数值字段，则 $\mathbf{x}_i$ 是标量值。

嵌入层

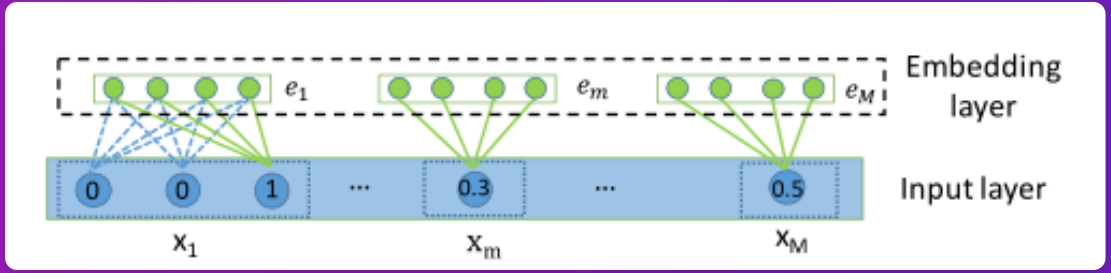
由于分类特征的特征表示非常稀疏和高维，一种常见的方法是将其表示到低维空间（例如，词嵌入）。具体来说，我们用低维向量表示每个分类特征，即 $\mathbf{e}_i = \mathbf{V}_i \mathbf{x}_i$，其中 V_i 是字段 i 的嵌入矩阵，$\mathbf{x}_i$ 是独热向量。通常情况下，分类特征可以是多值的，即 $\mathbf{x}_i$ 是一个多热向量。

以电影观看预测为例，可以有一种特征字段类型来描述电影的类型，并且可能具有多值性（例如，电影《泰坦尼克号》的戏剧和浪漫主义）。为了与多值输入兼容，我们进一步修改了等式2，并将多值特征表示为相应特征嵌入向量的平均值：$\mathbf{e_i} = \frac{1}{q}\mathbf{V}_\mathrm{i}\mathbf{x_i}$，其中 q 是样本对第 i 个字段的值的数量，$\mathbf{x}_i$ 是该字段的多热向量表示。

为了允许分类特征和数值特征之间的交互，我们还将数值特征表示在同一低维特征空间中。具体来说，我们将数值特征表示为 $\mathbf{e}_\mathrm{m} = \mathbf{v}_\mathrm{m} x_m$，其中 $\mathbf{v}_m$ 是字段 m 的嵌入向量，x_m 是标量值。

图2：输入层和嵌入层的图示

其中分类字段和数值字段均由低维密集向量表示。通过这样做，嵌入层的输出将是多个嵌入向量的串联，如图2所示。



交互层

一旦数字和分类特征位于同一低维空间中，我们就开始在该空间中建模高阶组合特征。关键问题是确定应组合哪些特征以形成有意义的高阶特征。

传统上，这是由领域专家完成的，他们根据自己的知识创建有意义的组合。在本文中，我们使用一种新方法，即多头自注意力机制来解决这个问题。

多头自注意力网络最近在建模复杂关系方面取得了显著的性能。例如，它显示了在机器翻译和句子嵌入中建模任意单词依存关系的优越性，并已成功应用于捕捉图嵌入中的节点相似性。在这里，我们扩展了这项最新技术，以建模不同特征字段之间的相关性。

交互层

具体来说，我们采用键值注意力机制来确定哪些特征组合是有意义的。以特征 m 为例，接下来我们解释如何识别涉及特征m 的多个有意义的高阶特征。我们首先定义特征 m 和特征 k 在特定注意头 h 下的相关性，如下所示：

$$\alpha_{\mathbf{m},\mathbf{k}}^{(\mathbf{h})}=\frac{\exp\left(\psi^{(h)}\left(\mathbf{e_m},\mathbf{e_k}\right)\right)}{\sum_{l=1}^{M}\exp\left(\psi^{(h)}\left(\mathbf{e_m},\mathbf{e_1}\right)\right)},$$
$$\psi^{(h)}\left(\mathbf{e_m},\mathbf{e_k}\right)=\left\langle \mathbf{W}_{\text{Query}}^{(\mathbf{h})}\,\mathbf{e_m},\mathbf{W}_{\text{Key}}^{(\mathbf{h})}\,\mathbf{e_k}\right\rangle,$$

交互层

其中 $\psi^{(h)}(\cdot,\cdot)$ 是一个注意力函数，定义了特征 m 和 k 之间的相似性。它可以定义为神经网络或与内积一样简单。在这项工作中，我们使用内积，因为它简单有效。$\mathbf{W}^{(\mathbf{h})}_{\text{Query}}$，$\mathbf{W}^{(\mathbf{h})}_{\text{Key}} \in \mathbb{R}^{d' \times d}$ 是将原始嵌入空间 $\mathbb{R}^{d}$ 映射到新空间 $\mathbb{R}^{d'}$ 的变换矩阵。接下来，我们通过组合由系数 $\alpha^{(\mathbf{h})}_{\mathbf{m},\mathbf{k}}$ 引导的所有相关特征来更新子空间 h 中特征 m的表示：

$$\tilde{\mathbf{e}}^{(\mathbf{h})}_{\mathbf{m}} = \sum_{k=1}^{M} \alpha^{(\mathbf{h})}_{\mathbf{m},\mathbf{k}} \left(\mathbf{W}^{(\mathbf{h})}_{\text{Value}} \, \mathbf{e}_{\mathbf{k}} \right)$$

其中 $\mathbf{W}^{(\mathbf{h})}_{\text{Value}} \in \mathbb{R}^{d' \times d}$。

交互层

$\tilde{\mathbf{e}}_{\mathbf{m}}^{(\mathbf{h})} \in \mathbb{R}^{d'}$ 是特征 m 及其相关特征（在头 h 下）的组合，它代表了通过我们的方法学习的一种新的组合特征。此外，一个特征也可能涉及不同的组合特征，我们通过使用多个头来实现这一点，这些头创建不同的子空间，并分别学习不同的特征交互。我们收集了在所有子空间中学习的组合特征，如下所示：

$$\tilde{\mathbf{e}}_{\mathbf{m}} = \tilde{\mathbf{e}}_{\mathbf{m}}^{(1)} \oplus \tilde{\mathbf{e}}_{\mathrm{m}}^{(2)} \oplus \cdots \oplus \tilde{\mathbf{e}}_{\mathrm{m}}^{(\mathrm{H})}$$

为了保留先前学习的组合特征，包括原始（即一阶）特征，我们在网络中添加了标准残差连接。形式上，

$$\mathbf{e}_{\mathrm{m}}^{\mathrm{Res}} = \mathrm{ReLU}\left(\tilde{\mathbf{e}}_{\mathrm{m}} + \mathbf{W}_{\mathrm{Res}}\mathbf{e}_{\mathrm{m}}\right)$$

其中 $\mathbf{W}_{\mathrm{Res}} \in \mathbb{R}^{d'H \times d}$ 是维数不匹配情况下的投影矩阵。

通过这样一个交互层，每个特征 $\mathbf{e}_m$ 的表示将更新为新的特征表示 $\mathbf{e}_{\mathrm{m}}^{\mathrm{Res}}$，即高阶特征的表示。我们可以将多个这样的层与前一个交互层的输出堆叠起来，作为下一个交互层的输入。通过这样做，我们可以对任意顺序的组合特征进行建模。

输出层

交互层的输出是一组特征向量 $\left\{\mathrm{e}_{\mathrm{m}}^{\mathrm{Res}}\right\}_{m=1}^{M}$，其中包括残差块保留的原始个体特征和通过多头自注意力机制学习的组合特征。对于最终的CTR预测，我们只需将所有这些连接起来，然后应用非线性投影，如下所示：

$$\hat{y}=\sigma\left(\mathbf{w}^{\mathrm{T}}\left(\mathbf{e}_{1}^{\mathrm{Res}} \oplus \mathbf{e}_{2}^{\mathrm{Res}} \oplus \cdots \oplus \mathbf{e}_{\mathbf{M}}^{\mathrm{Res}}\right)+b\right)$$

其中 $\mathbf{w} \in \mathbb{R}^{d^{\prime} H M}$ 是线性组合串联特征的列投影向量，b 是偏差。

训练

我们的损失函数是对数损失，其定义如下：

$$\text{Logloss} = -\frac{1}{N}\sum_{j=1}^{N}\left(y_j \log\left(\hat{y}_j\right) + \left(1-y_j\right)\log\left(1-\hat{y}_j\right)\right)$$

其中 y_j 和 $\hat{y}_j$ 分别是用户点击的真实情况和估计的点击率，j 对训练样本进行索引，N 是训练样本总数。在我们的模型中要学习的参数是 $\left\{\mathbf{V_i}, \mathbf{v_m}, \mathbf{W}_{\text{Query}}^{(\text{h})}, \mathbf{W}_{\text{Key}}^{(\text{h})}, \mathbf{W}_{\text{Value}}^{(\text{h})}, \mathbf{W}_{\text{Res}}, \mathbf{w}, b\right\}$，这些参数通过使用梯度下降最小化总对数损失来更新。

建模任意顺序的组合特征

给定由等式5-8定义的特征交互算子，我们现在分析如何在我们提出的模型中建模低阶和高阶组合特征。

为了简单起见，假设有四个特征字段（即 $M = 4$），分别由x_1、 x_2、 x_3、 x_4表示。在第一个交互层内，每个单独的特征通过注意力机制与任何其他特征交互，因此，用不同的相关权重捕捉一组二阶特征组合，例如 $g(x_1, x_2)$，$g(x_2, x_3)$ 和 $g(x_3, x_4)$，其中，可以通过激活函数 $ReLu$ 的非线性来确保相互作用函数 $g(\cdot)$ 的非加性。

理想情况下，涉及 x_1 的组合特征可以编码到第一个特征字段 $\mathbf{e}_1^{\text{Res}}$ 的更新表示中。由于可以为其他特征字段推导出相同的表示，所有二阶特征交互可以编码在第一交互层的输出中，其中注意力权重提取有用的特征组合。

建模任意顺序的组合特征

接下来，我们证明了高阶特征交互可以在第二个交互层内建模。给定第一个特征字段 $\mathbf{e}_1^{\text{Res}}$ 的表示和由第一个交互层生成的第三个特征字段 $\mathbf{e}_3^{\text{Res}}$ 的表示，可以通过允许 $\mathbf{e}_1^{\text{Res}}$ 参与 $\mathbf{e}_3^{\text{Res}}$ 来建模涉及 x_1、x_2 和 x_3的三阶组合特征，因为 $\mathbf{e}_1^{\text{Res}}$ 包含交互 $g(x_1, x_2)$，$\mathbf{e}_3^{\text{Res}}$ 包含单个特征 x_3（来自残差连接）。此外，组合特征的最大阶数随交互层的数量呈指数增长。例如，可以通过 $\mathbf{e}_1^{\text{Res}}$ 和 $\mathbf{e}_3^{\text{Res}}$ 的组合来捕捉四阶特征交互作用$(x_1$、x_2、x_3、$x_4)$，其分别包含二阶交互作用 (x_1, x_2) 和 (x_3, x_4)。因此，几个交互层足以模拟高阶特征交互。

基于上述分析，我们可以看到 AutoInt 以分层的方式学习具有注意力机制的特征交互，即从低阶到高阶，并且所有低阶特征交互都由残差连接进行。这是有希望和合理的，因为学习层次表示在计算机视觉和深度神经网络语音处理中已证明相当有效。

五、实验

实验

我们的目标是回答以下问题：

1. 我们提出的AutoInt如何处理CTR预测问题？它对大规模稀疏和高维数据有效吗？
2. 不同模型配置的影响是什么？
3. 不同功能之间的依赖结构是什么？我们提出的模型可以解释吗？
4. 集成隐式特征交互是否会进一步提高性能？

RQ1

有效性评估：我们将10次不同运行的平均结果总结到表2中。我们有以下观察结果：

1. 探索二阶特征相互作用的FM和AFM在所有数据集上始终大幅度优于LR，这表明单个特征在CTR预测中不足。
2. 一个有趣的观察结果是，一些捕捉高阶特征交互的模型较差。例如，尽管 Deep Crossing 和 NFM 使用深度神经网络作为学习高阶特征交互的核心组件，但它们不能保证比 FM 和 AFM 更好。这可能归因于他们以隐式方式学习特征交互。相反，CIN 明确地做到了这一点，并且始终优于低阶模型。
3. 在 Criteo 和 MovieLens-1M 数据集上，HOFM 显著优于 FM，这表明建模三阶特征交互有助于预测性能。
4. AutoInt 在四个真实数据集中的三个上实现了最佳性能的总体基线方法。在Avazu数据集上，CIN 在AUC评估中的表现略优于AutoInt，但我们得到了更低的对数损失。注意，除了特征交互层外，我们提出的 AutoInt 与 Deep Crossing 共享相同的结构，这表明使用注意力机制学习显式组合特征至关重要。

表2：不同算法的有效性比较

Model Class	Model	Criteo		Avazu		KDD12		MovieLens-1M	
		AUC	Logloss	AUC	Logloss	AUC	Logloss	AUC	Logloss
First-order	LR	0.7820	0.4695	0.7560	0.3964	0.7361	0.1684	0.7716	0.4424
Second-order	FM [26]	0.7836	0.4700	0.7706	0.3856	0.7759	0.1573	0.8252	0.3998
	AFM[40]	0.7938	0.4584	0.7718	0.3854	0.7659	0.1591	0.8227	0.4048
High-order	DeepCrossing [32]	0.8009	0.4513	0.7643	0.3889	0.7715	0.1591	0.8448	0.3814
	NFM [13]	0.7957	0.4562	0.7708	0.3864	0.7515	0.1631	0.8357	0.3883
	CrossNet [38]	0.7907	0.4591	0.7667	0.3868	0.7773	0.1572	0.7968	0.4266
	CIN [19]	0.8009	0.4517	**0.7758**	0.3829	0.7799	0.1566	0.8286	0.4108
	HOFM [5]	0.8005	0.4508	0.7701	0.3854	0.7707	0.1586	0.8304	0.4013
	AutoInt (ours)	**0.8061****	**0.4455****	0.7752	**0.3824**	**0.7883****	**0.1546****	**0.8456***	**0.3797****

RQ1

模型效率评估：我们在图4中给出了四个数据集上不同算法的运行时结果。 不出所料，LR 由于其简单性是最有效的算法。FM和NFM在运行时方面表现类似，因为NFM仅在二阶交互层的顶部堆叠一个前馈隐藏层。在所有列出的方法中，在所有基线中实现最佳预测性能的 CIN 由于其复杂的交叉层而更加耗时。这可能会使其难以部署在工业场景。注意，AutoInt 足够高效，与高效算法 DeepCrossing 和 NFM 相当。

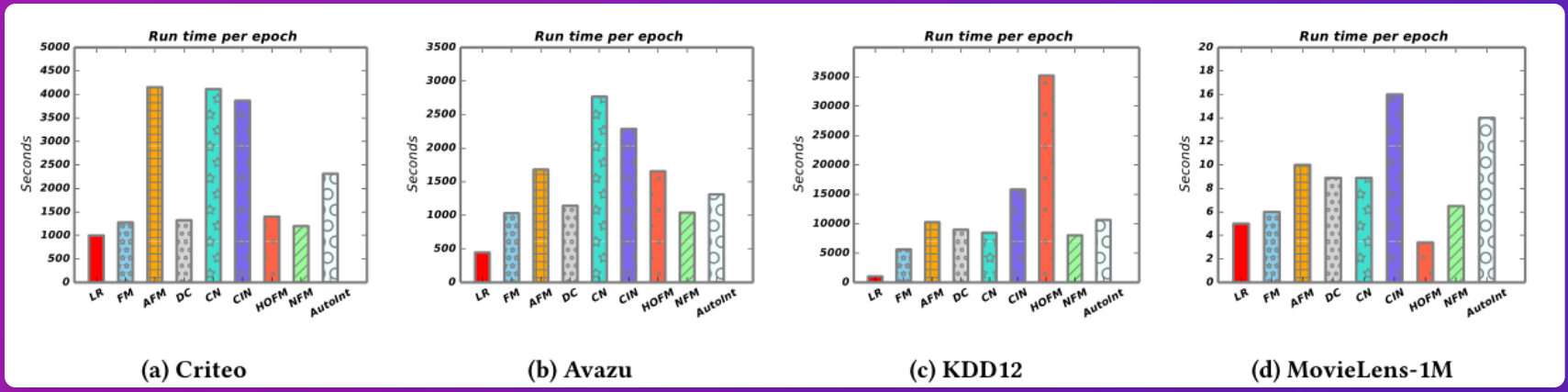
我们还比较了不同模型的大小（即参数数量），作为效率评估的另一个标准。如表3所示，与基线模型中的最佳模型 CIN 相比，AutoInt中的参数数量要小得多。

表3:Criteo数据集上不同算法在模型大小方面的效率比较。

Model	DC	CN	CIN	NFM	AutoInt
#Params	1.6×10^5	3×10^3	1.9×10^6	4×10^3	3.9×10^4

总之，在所有比较模型中，我们提出的 AutoInt 实现了最佳性能。与最具竞争力的基线模型 CIN 相比，AutoInt需要更少的参数，在线推理效率更高。

图4：在运行时间方面，不同算法的效率比较



RQ2

为了进一步验证并深入了解所提出的模型，我们进行了消融研究，并比较了AutoInt的几种变体。

残差结构的影响：标准的 AutoInt 利用残差连接，这些连接承载所有学习到的组合特征，因此允许建模非常高阶的组合。为了证明残差单元的贡献，我们将其从标准模型中分离出来，并保持其他结构不变。如表4所示，我们观察到，如果删除残差连接，所有数据集的性能都会下降。具体来说，在 KDD12 和 MovieLens-1M 数据上，完整模型的性能大大优于变体，这表明在我们提出的方法中，残差连接对于建模高阶特征交互至关重要。

表4：消融研究比较了带和不带残差连接的 AutoInt 的性能。AutoIntw/ 是完整的模型，而 AutoIntw/o 是没有残差连接的模型

Data Sets	Models	AUC	Logloss
Criteo	AutoInt $_{w/}$	0.8061	0.4454
	AutoInt $_{w/o}$	0.8033	0.4478
Avazu	AutoInt $_{w/}$	0.7752	0.3823
	AutoInt $_{w/o}$	0.7729	0.3836
KDD12	AutoInt $_{w/}$	0.7888	0.1545
	AutoInt $_{w/o}$	0.7831	0.1557
MovieLens-1M	AutoInt $_{w/}$	0.8460	0.3784
	AutoInt $_{w/o}$	0.8299	0.3959

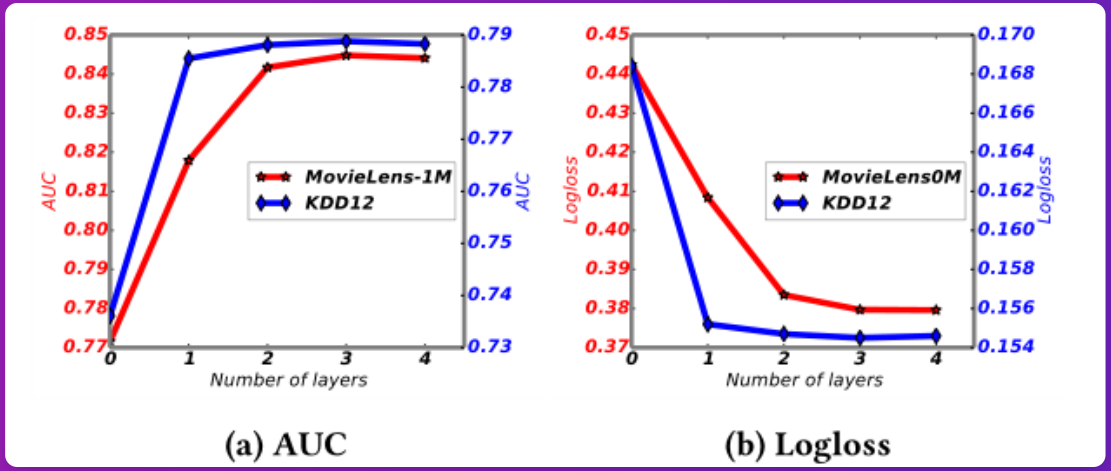
RQ2

网络深度的影响：我们的模型通过堆叠多个交互层来学习高阶特征组合。因此，我们感兴趣的是性能如何随交互层的数量而变化，即组合特征的顺序。注意，当没有交互层（即层数等于零）时，我们的模型将原始单个特征的加权和作为输入，即不考虑组合特征。

结果总结在图5中。我们可以看到，如果使用一个交互层，即考虑特征交互，两个数据集的性能都会显著提高，这表明组合特征对于预测非常有用。随着交互层的数量进一步增加，即考虑高阶组合特征，模型的性能进一步提高。当层数达到三层时，性能变得稳定，这表明添加极高阶特征对预测没有帮助。

图5：交互层的数量

Criteo和Avazu数据集的结果相似，因此省略。

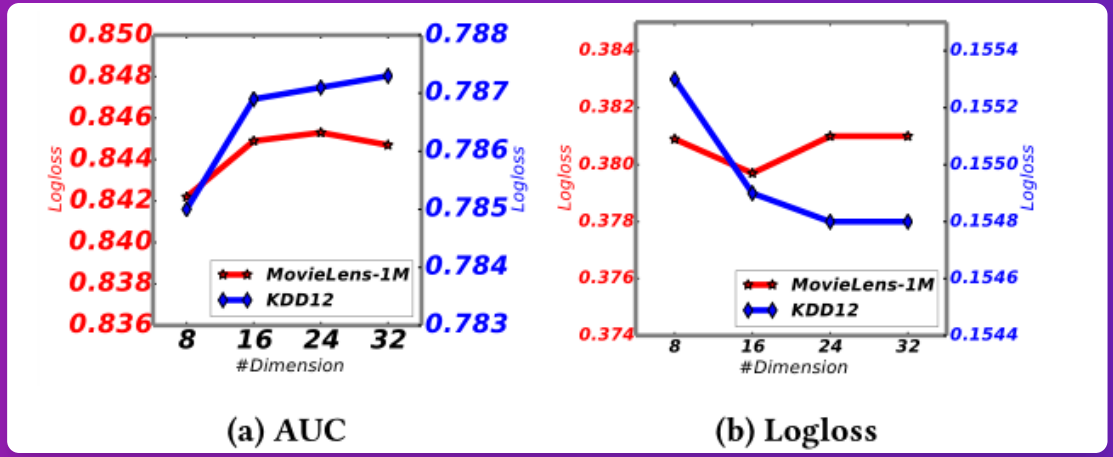


(a) AUC　　(b) Logloss

RQ2

不同尺寸的影响：接下来，我们研究了嵌入层的输出维参数 d 的性能。在KDD12数据集上，我们可以看到，由于使用较大的模型进行预测，随着维数的增加，性能不断提高。在MovieLens-1M数据集上的结果不同。当维度大小达到24时，性能开始下降。原因是该数据集较小，当使用太多参数时，模型拟合过度。

图6：嵌入层维度的影响



(a) AUC (b) Logloss

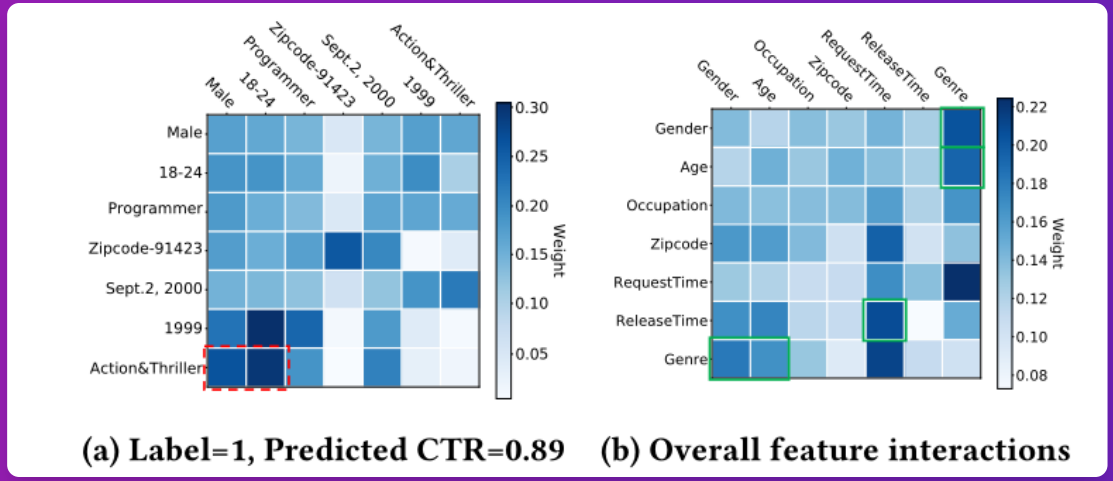
RQ3

一个好的推荐系统不仅可以提供好的推荐，而且可以提供良好的解释性。因此，在这一部分中，我们将介绍我们的AutoInt如何能够解释推荐结果。我们以MovieLens-1M数据集为例。

让我们看看我们的算法建议的推荐结果，即用户喜欢一个项目。图7（a）显示了通过注意力评分获得的输入特征不同字段之间的相关性。我们可以看到，AutoInt 能够识别有意义的组合特征（即红点矩形）。这是很合理的，因为年轻人很可能更喜欢动作片和颤音片。

我们还对数据中不同特征字段之间的相关性感兴趣。因此，我们根据特征字段在整个数据中的平均注意力分数来衡量特征字段之间的相关性。图7（b）总结了不同字段之间的相关性。我们可以看到，和（即实心绿色区域）是强相关的，这是该领域推荐的可解释规则。

图7:MovieLens-1M上某个样本和全局级特征交互的注意力权重热图



(a) Label=1, Predicted CTR=0.89 (b) Overall feature interactions

RQ4

集成隐式交互前馈神经网络能够建模隐式特征交互，并已广泛集成到现有的CTR预测方法中。为了研究集成隐式特征交互是否进一步提高性能，我们通过联合训练将 AutoInt 与两层前馈神经网络相结合。我们将联合模型命名为 AutoInt+，并将其与以下算法进行比较：

- Wide&Deep。Wide&Deep整合了 logistic 回归和前馈神经网络的输出。
- DeepFM。DeepFM 结合了传统的二阶分解机和前馈神经网络，具有共享的嵌入层。
- Deep&Cross。Deep&Cross 是通过集成前馈神经网络对交叉网络的扩展。
- xDeepFM。xDeepFM 是通过集成前馈神经网络对CIN的扩展。

RQ4

表5给出了联合训练模型的平均结果（超过10次）。我们有以下观察结果：

1. 通过在所有数据集上与前馈神经网络联合训练，我们的方法的性能得到了提高。这表明，整合隐式特征交互确实提高了我们提出的模型的预测能力。然而，从最后两列中可以看出，与其他模型相比，性能改进的幅度相当小，这表明我们的单个模型 AutoInt 相当强大。
2. 在集成隐式特征交互后，AutoInt+ 优于所有竞争方法，并在使用的CTR预测数据集上实现了最新的性能。

表5：整合隐式特征交互的结果。

Model	Criteo		Avazu		KDD12		MovieLens-1M		Avg. Changes	
	AUC	Logloss	AUC	Logloss	AUC	Logloss	AUC	Logloss	AUC	Logloss
Wide&Deep (LR)	0.8026	0.4494	0.7749	0.3824	0.7549	0.1619	0.8300	0.3976	+0.0292	-0.0213
DeepFM (FM)	0.8066	0.4449	0.7751	0.3829	0.7867	0.1549	0.8437	0.3846	+0.0142	-0.0113
Deep&Cross (CN)	0.8067	0.4447	0.7731	0.3836	0.7872	0.1549	0.8446	0.3809	+0.0200	-0.0164
xDeepFM (CIN)	0.8070	0.4447	0.7770	0.3823	0.7820	0.1560	0.8463	0.3808	+0.0068	-0.0096
AutoInt+ (ours)	**0.8083****	**0.4434****	**0.7774***	**0.3811****	**0.7898****	**0.1543****	**0.8488****	**0.3753****	+0.0023	-0.0020

六、总结与展望

总结与展望

在这项工作中，我们提出了一种新的基于自注意力机制的CTR预测模型，该模型可以以显式方式自动学习高阶特征交互。我们方法的关键是新引入的交互层，它允许每个特征与其他特征交互，并通过学习确定相关性。在四个真实数据集上的实验结果证明了我们提出的模型的有效性和效率。

此外，通过可视化学习的组合特征，我们提供了良好的模型解释能力。与以前最先进的方法相比，当与前馈神经网络捕获的隐式特征交互进行集成时，我们获得了更好的离线AUC和对数损失分数。

对于未来的工作，我们感兴趣的是将上下文信息合并到我们的方法中，并改进其在在线推荐系统中的性能。此外，我们还计划将 AutoInt 扩展到一般的机器学习任务，如回归、分类和排序。

Thank you