基于改进 Transformer 的广告点击率预估模型*

周 菲，徐洪珍†
(东华理工大学 信息工程学院，南昌 330013)

摘 要：针对现有的广告点击率预估模型未能精准挖掘用户历史兴趣及历史兴趣对目标广告点击与否的影响，提出了一种基于改进 Transformer 的广告点击率预估模型。该模型采用 Transformer 网络捕捉隐藏在用户点击序列背后的潜在历史兴趣；同时针对 Transformer 建模用户历史兴趣无法有效关联目标广告的问题，提出了一种改进的 Transformer 网络。改进后的 Transformer 不但有效建模用户历史兴趣，而且考虑了跟目标广告的关联。新模型采用辅助损失函数来监督改进的 Transformer 对用户历史兴趣的抽取过程，然后采用注意力机制进一步建模用户的历史兴趣和目标广告的相关性以提升模型的预估性能。实验结果表明新模型有效提升了广告点击率的预估效果。

关键词：广告点击率；Transformer；点击序列；注意力机制
中图分类号：TP10 **文献标志码**：A **文章编号**：1001-3695(2021)08-025-2386-04
doi:10.19734/j.issn.1001-3695.2020.09.0355

Improved Transformer based model for click-through rate prediction

Zhou Fei, Xu Hongzhen†
(*School of Information Engineering, East China University of Technology, Nanchang* 330013, *China*)

Abstract: Aiming at the problem that the existing click-through rate prediction models fail to accurately dig out the historical interest of users and the influence of historical interest on whether or not the target advertisement is clicked, this paper proposed a click-through rate prediction model based on the improved Transformer. The new model used Transformer network to capture the potential historical interest hidden behind the user's click sequence. At the same time, in response to the problem that the Transformer modeling user's historical interest sequence could not effectively associate the target advertisement, it proposed an improved Transformer network. The new model not only effectively captured the user's historical interest, but also considered the association with the target advertisement to enhance estimated performance. Experimental results demonstrate that the new model shows better performance than other models.

Key words: click-through rate; Transformer; click sequence; attention mechanism

0 引言

广告点击率(click-through rate,CTR)指的是给定用户和网页内容,广告被点击的次数占总展示次数的比例[1]。广告精准投放,依赖于预估目标受众对相应广告的 CTR[2]。CTR 预估是互联网公司的重要研究课题,CTR 的准确度不仅影响公司广告产品的收入,同时也影响用户的体验度。

传统的 CTR 预估主要采用逻辑回归(logistic regression,LR)作为 CTR 预测模型[3,4],LR 模型的优点是简单、可解释性强,缺点是作为一个线性模型,无法进行特征的交叉。POLY2 模型通过二阶特征交叉在一定程度上解决了特征的组合问题[3],但是该模型的特征向量极度稀疏,导致权重难以训练,模型收敛困难。随后,非线性模型在 CTR 预估中得到了广泛关注,代表模型为因子分解机模型(factorization machine,FM)[5]。FM 将高维稀疏向量转换为低维稠密向量从而大幅减少了权重参数的数量,FM 的缺点是只能进行二阶特征交叉,无法学习高阶特征交叉信息。随后,文献[6]基于 FM 提出了特征域相关的因子分解机模型(field-aware factorization machine,FFM),相比 FM,FFM 引入了特征域感知的概念,每个特征针对其他特征域学习不同的特征隐向量。

随着深度学习的兴起,基于深度学习的 CTR 预估模型在探索特征之间的高阶组合方面取得了大幅进展,Qu 等人[7]设计乘积层,通过对特征的乘积操作从而更有针对性地获取特征之间的交叉信息。Wide&Deep 模型[8]于 2016 年由谷歌提出,Wide 侧采用逻辑回归,从而让模型具有较强的记忆能力,Deep 侧使用多层感知机使得模型具有较强的泛化能力。Guo 等人[9]通过在 Wide&Deep 基础上使用 FM 替代原来的逻辑回归,从而加强了特征之间的两两交叉;Wang 等人[10]借鉴深度残差网络[11]结构设计 Cross 网络替代 Wide&Deep 的 Wide 部分,可以显式学习特征的高阶交互。微软亚洲研究院提出 xDeepFM 模型[12],该网络不仅能同时显式和隐式学习特征的交叉,而且兼具记忆和泛化能力。Song 等人[13]进一步提出 AUTOINT模型,采用多头自注意力机制进行特征的高阶组合。Zhou 等人[14]提出将注意力机制应用于推荐模型,从而在一定程度上改善了预估的准确性。文献[15]提出将 CNN 用于 CTR 预估,通过 CNN 进行特征的生成。文献[2]将深度置信网络引入 CTR 预估,提出了基于深度置信网络作为构造模型的融合模型,并基于梯度下降算法和改进型粒子群算法对融合模型进行优化,使得 CTR 预估精度相比传统模型得到一定的改进。文献[3]提出了一种基于模型融合的点击率预估模型,通过融合不同结构的深度神经网络模型,特征得到不同阶数的交叉,使得融合后的模型学习到更充分的特征交叉信息。但是文献

收稿日期：2020-09-21；**修回日期**：2020-11-14 **基金项目**：江西省青年科学家培养对象计划项目(20142BCB23017)；江西省教育厅科技计划项目(GJJ151538,GJJ160554)；江西省放射性地学大数据技术工程实验室开放项目(JELRGBDT201802)

作者简介：周菲(1992-)，女，江西九江人，硕士研究生，主要研究方向为机器学习、推荐系统、计算广告；徐洪珍(通信作者)，男，教授，博士，主要研究方向为机器学习、云计算(26909204@qq.com)。

[2,3]都没有考虑用户历史点击序列对目标广告点击与否的影响。在非搜索电商场景中,用户并不会显式地表达自身的兴趣偏好,同时用户的兴趣是动态变化的,因此用户历史点击序列对于预估用户当前的点击行为是至关重要的。举例来说: 某位用户上周购买一台笔记本电脑,那么当他完成购买后,下次购买鼠标或机械键盘的概率则远远大于笔记本电脑的概率。点击序列的作用在于: a) 用户的下次点击行为是受近期点击影响的; b) 序列模型可以捕捉用户兴趣的变化趋势。然而这些模型忽略了精准建模用户的潜在历史兴趣及历史兴趣跟目标广告之间的关联。因此,本文提出一种基于改进 Transformer 的广告点击率预估模型,称为 SACSN(self-attentive click sequence network) 。SACSN 结构采用 Transformer[16] 建模用户的历史点击序列,从而捕捉用户的动态兴趣。Transformer 基于一种自注意力机制,有效缓解了 RNN 对于长序列的时间依赖问题。SASRec[17] 和 BERT4Rec[18] 已经证明 Transformer 网络对于序列问题建模的有效性,但是 Transformer 只能建模点击序列背后的用户历史兴趣,无法将用户历史兴趣和目标广告相关联。本文在原 Transformer 网络中增加目标广告的多头注意力层,改进后的 Transformer 不但有效建模用户历史兴趣,而且考虑了与目标广告的关联,并采用辅助损失函数用于监督兴趣的抽取。通过实验对比,SACSN 在 CTR 预估任务中是有效的。SACSN 的主要贡献总结如下:

a) 不同于之前的模型将用户的点击行为直接当成用户的兴趣,本文将 Transformer 网络用于建模用户的点击序列,以捕捉用户点击序列背后的兴趣; 同时针对 Transformer 建模用户历史兴趣序列无法有效关联目标广告的问题,提出了在原 Transformer 网络中增加目标广告的多头注意力层。改进后的 Transformer 不但有效建模用户历史兴趣,而且考虑了跟目标广告的关联。

b) 采用辅助损失函数监督 Transformer 每一步的兴趣输出,使得模型的输出和用户的真实兴趣差距得以缩小,提升了模型的准确度。

c) 进一步将序列建模和注意力机制[14] 相结合,经Transformer 建模后的兴趣和目标广告根据相似度进行显式加权,使得相关性高的兴趣得到增强,低相关度兴趣得到削弱,从而对兴趣的建模更加有效。

1　基于改进 Transformer 的广告点击率预估模型

首先介绍模型的输入,然后自底向上地逐一介绍模型的各个组件,模型结构如图 1 所示,包括: a) 嵌入层,用于将从原始特征转换为对应的嵌入向量; b) 改进的 Transformer 层,改进点是在原 Transformer 中增加了与目标广告的多头注意力层,改进后的 Transformer 不但有效建模用户历史兴趣,而且考虑了与目标广告的关联; c) 注意力层,将用户历史兴趣和目标广告的 embedding 向量通过注意力机制[14] ,得到经过目标广告加权过后的用户历史兴趣; d) 拼接层,将加权过后的用户历史兴趣和目标广告、上下文、用户画像等特征的 embedding 向量进行拼接,然后输入到后续的多层感知机层; e) 多层感知机层,通过多层神经网络进一步加强特征之间的交叉,从而学习出交叉后的高阶特征表达; f) 得到模型的输出,计算模型的损失函数。

1.1　模型的输入

模型的输入主要包括: 用户行为序列的 one-hot 向量 $\boldsymbol{S}_u$、目标广告的 one-hot 向量 $\boldsymbol{x}_a$、上下文特征的 one-hot 向量 $\boldsymbol{x}_c$、用户画像特征的 one-hot 向量 $\boldsymbol{x}_p$。原始特征首先需要经过 one-hot 编码[19] 转换成 one-hot 向量,用户行为序列表示为: $\boldsymbol{S}_u = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_T\}$,其中 T 为用户点击的个数,$\boldsymbol{b}_i$ 为用户第 i 个点击物品的 one-hot 向量。在 CTR 预估中,上下文特征为用户点击或购买的时间、设备信息等,用户画像特征包括用户的 ID、年龄、性别等。

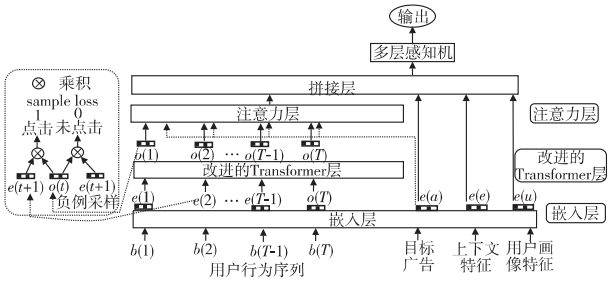


图 1　SACSN 模型的结构

Fig. 1　Structure of SACSN model

1.2　嵌入层

嵌入层主要用于将 one-hot 编码后的高维稀疏向量转换为低维稠密向量。将用户点击序列的 one-hot 向量 $\boldsymbol{S}_u$、目标广告特征的 one-hot 向量 $\boldsymbol{x}_a$、上下文特征的 one-hot 向量 $\boldsymbol{x}_c$ 以及用户画像特征的 one-hot 向量 $\boldsymbol{x}_p$ 经过 embedding 层的 embedding 技术得到用户点击序列嵌入向量: $\boldsymbol{E}_u = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_T\}$,$\boldsymbol{E}_u \in \mathbb{R}^{T \times d_{model}}$,其中 T 为用户点击序列的长度,d_{model} 为 Embedding 的维度。目标广告嵌入向量 $\boldsymbol{E}_a \in \mathbb{R}^{N_a \times d_{model}}$、上下文特征嵌入向量 $\boldsymbol{E}_c \in \mathbb{R}^{N_c \times d_{model}}$、用户画像嵌入向量 $\boldsymbol{E}_p \in \mathbb{R}^{N_p \times d_{model}}$,$N_a$、$N_c$、$N_p$ 分别为目标广告 $\boldsymbol{x}_a$、上下文特征 $\boldsymbol{x}_c$、用户画像 $\boldsymbol{x}_p$ 的特征域个数,$\boldsymbol{e}_1 \sim \boldsymbol{e}_T$ 即用户点击序列中第 1 ~ T 位置的物品嵌入向量。

1.3　改进的 Transformer 层

Transformer 层由改进的 Transformer 网络和 sample loss 组成。其中改进的 Transformer 网络主要由位置编码、多头自注意力、归一化和残差层、跟目标广告的多头注意力、前向全连接层等组成,其中一层 Transformer 网络称为一个 block,多层 Transformer 的叠加则称为多个 block。其结构如图 2 所示。

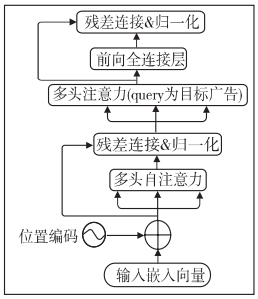


图 2　改进的 Transformer 的网络结构

Fig. 2　Structure of improved Transformer

1.3.1　Transformer 网络的位置编码

为了表示序列中物品之间的位置关系,对序列中的每个物品向量均分配一个位置向量,公式表示为

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d}) \tag{1}$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{(2i+1)/d}) \tag{2}$$

其中: PE 表示位置编码(position encoding) ; pos 为物品在序列中的位置; d 表示位置编码的维度,这里和嵌入层的维度相同; $2i$ 表示偶数位置,$2i+1$ 表示奇数位置。融合了位置信息的序列嵌入向量 $\boldsymbol{Z}$ 的公式表示为

$$\boldsymbol{Z} = \boldsymbol{E}_u + PE \tag{3}$$

1.3.2　Transformer 网络的多头自注意力层

将融合了位置信息的点击序列嵌入向量 $\boldsymbol{Z}$ 进行多头自注意力计算,过程如下: $\boldsymbol{W}_i^Q$ 为查询变换矩阵权重向量,$\boldsymbol{W}_i^K$ 为关键字变换矩阵权重向量,$\boldsymbol{W}_i^V$ 为值变换矩阵权重向量,通过 $\boldsymbol{W}_i^Q$、$\boldsymbol{W}_i^K$、$\boldsymbol{W}_i^V$ 将 $\boldsymbol{Z}$ 分别转换为查询向量 $\boldsymbol{Q} = \boldsymbol{Z}\boldsymbol{W}_i^Q$、关键字向量 $\boldsymbol{K} = \boldsymbol{Z}\boldsymbol{W}_i^K$、值向量 $\boldsymbol{V} = \boldsymbol{Z}\boldsymbol{W}_i^V$,$\boldsymbol{Q}$ 和 $\boldsymbol{K}$ 进行点积计算,结果除以缩放因子$\sqrt{d}$,经过 softmax 函数后再和 $\boldsymbol{V}$ 相乘得到自注意力后的结果,多头自注意力为 H 个自注意力拼接,即并行将自注意力进行 H 次的结果拼接后得到多头自注意力层的输出 $\boldsymbol{S}$,多头自

注意力的公式为

$$head_i = \text{multihead}(\boldsymbol{Z}) = \text{attention}(\boldsymbol{Z}\boldsymbol{W}_i^Q, \boldsymbol{Z}\boldsymbol{W}_i^K, \boldsymbol{Z}\boldsymbol{W}_i^V) \tag{4}$$

$$\text{attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}}\right)\boldsymbol{V} \tag{5}$$

$$\boldsymbol{S} = \text{concat}(head_1, head_2, \cdots, head_H)\,\boldsymbol{W}^C \tag{6}$$

其中: $head_i$ 表示第 i 个 $head$ 自注意力($1 \leqslant i \leqslant H$) ,多头注意力 S 为 H 个注意力的拼接。

1.3.3 Transformer 网络的归一化和残差层

LayerNorm[20] 表示层归一化,主要作用在于加快模型的收敛速度; dropout[21] 为随机失活,用于在参数较多的模型中防止过拟合,同时采用残差网络结构,减少模型的学习负荷,公式如下:

$$\boldsymbol{S}' = \text{LayerNorm}(\boldsymbol{Z} + \text{dropout}(\boldsymbol{S})) \tag{7}$$

1.3.4 Transformer 网络目标广告的多头注意力层

在 1.3.2 节中,多头自注意力的查询向量 $\boldsymbol{Q}$、关键字向量 $\boldsymbol{K}$、值向量 $\boldsymbol{V}$ 其输入均为点击序列嵌入向量 $\boldsymbol{Z}$,区分于多头自注意力,目标广告的多头注意力的查询向量 $\boldsymbol{Q}'$的输入为目标广告嵌入向量 $\boldsymbol{E}_a$,关键字向量 $\boldsymbol{K}'$、值向量 $\boldsymbol{V}'$其输入为归一化和残差层的输出 $\boldsymbol{S}'$,计算公式同多头自注意力。此改进使得 Transformer 不仅可以关注点击序列自身的不同位置,学习到点击序列背后的用户历史兴趣,同时将用户历史兴趣与目标广告相关联,从而增强了预测用户下一次点击的准确性。

1.3.5 通过前向全连接层加强模型的非线性能力

由于之前的多头注意力层本质还是线性变换,前向全连接层通过两层神经网络加强模型的非线性能力,公式如下:

$$\boldsymbol{O} = \text{LayerNorm}(\boldsymbol{S}' + \text{dropout}(\text{ReLU}(\boldsymbol{S}'\boldsymbol{W}^{(1)} + b^{(1)})\,\boldsymbol{W}^{(2)} + b^{(2)})) \tag{8}$$

其中: $\boldsymbol{O}$ 为前向全连接层的输出向量, $\boldsymbol{S}'$为归一化过后的向量。由于原输入是一个长度为 T 的序列,所以将 $\boldsymbol{O}$ 进一步表示为 $\boldsymbol{O} = \{o_1, o_2, \cdots, o_t, \cdots, o_T\}$,为方便后续描述,将 $\boldsymbol{O}$ 称为用户点击兴趣向量。$W^{(1)}$ 、$W^{(2)}$ 和 $b^{(1)}$ 、$b^{(2)}$ 分别为两层前向全连接层的权重系数和偏置。

1.3.6 sample loss 监督兴趣的抽取

本文利用 sample loss 监督改进后 Transformer 的每一步输出,具体方法如下: 采用用户的 $t+1$ 步点击的物品 b_{t+1}来监督模型第 t 步的输出 o_t, b_{t+1}为用户第 $t+1$ 步点击的物品,表示为正样本,通过负例采样随机选择总物品中未被该用户点击过的物品为负样本,于是产生$\{\boldsymbol{e}^i, \hat{\boldsymbol{e}}^i\}$,$i = 1, 2, \cdots, T$, $\boldsymbol{e}^i \in \mathbb{R}^{T \times d_{model}}$, T 为用户历史点击序列的个数, d_{model} 表示 embedding 的维度。$\boldsymbol{e}^i[t+1]$表示用户点击序列的第 $t+1$ 个物品的 embedding 向量, $\hat{\boldsymbol{e}}^i[t+1]$表示经过负例采样得到序列的第 $t+1$ 个物品的 embedding 向量, N 为样本的总数, sample loss 的表示向量 $\boldsymbol{L}_{Sample}$公式为

$$\boldsymbol{L}_{Sample} = -\frac{1}{N}\left(\sum_{i=1}^{N}\sum_t \log \sigma(o_t^i \times \boldsymbol{e}^i[t+1]) + \log(1 - \sigma(o_t^i \times \hat{\boldsymbol{e}}^i[t+1]))\right) \tag{9}$$

1.4 注意力层

Transformer 网络的多头自注意力和目标广告的多头注意力应用于用户历史点击序列中,其用途分别为捕捉序列中不同位置的物品之间的联系及隐式建模序列的物品和目标广告之间的关联。与之相区别,本层的注意力进一步显式建模用户历史点击序列和目标广告之间的相关程度,公式如下:

$$\boldsymbol{a}_t = \frac{\exp(\boldsymbol{o}_t \cdot \boldsymbol{W} \cdot \boldsymbol{E}_a)}{\sum_{j=1}^{T}\exp(o_j \cdot \boldsymbol{W} \cdot \boldsymbol{E}_a)} \tag{10}$$

$$A = \sum_{j=1}^{T} a_j / T \tag{11}$$

其中: $\boldsymbol{a}_t$ 为经目标广告嵌入向量 $\boldsymbol{E}_a$ 加权后的用户点击兴趣向量; $\boldsymbol{o}_t$ 为序列中第 t 个位置的点击兴趣向量; $\boldsymbol{E}_a$ 为目标广告的嵌入向量; $\boldsymbol{W}$ 为权重向量, $\boldsymbol{W} \in \mathbb{R}^{d_h \times d_{model}}$, d_h 为用户点击兴趣向量的维度, d_{model}为目标广告 embedding 的维度。$\boldsymbol{a}_t$ 值越大表明输入 $\boldsymbol{o}_t$ 和目标广告 $\boldsymbol{E}_a$ 的相似度越高。A 表示经注意力机制加权过后的用户历史兴趣。

1.5 拼接层

拼接层用于拼接加权后的用户历史兴趣 A 和目标广告、上下文、用户画像的嵌入向量 $\boldsymbol{E}(a)$ 、$\boldsymbol{E}(c)$ 、$\boldsymbol{E}(p)$,将拼接后的结果作为后续网络的输入,公式表示为

$$H = \text{concat}(A, \boldsymbol{E}(a) + \boldsymbol{E}(c) + \boldsymbol{E}(p)) \tag{12}$$

1.6 多层感知机层

激活函数为 softmax,最后一层的隐向量个数为 2,即表示 CTR 预估的二分类,通过 softmax 激活函数将二分类的输出向量转换为概率 p:

$$H^{(l)} = \sigma(\boldsymbol{W}_f^l \boldsymbol{H}^{(l-1)} + b_f^l) \tag{13}$$

$$p = \sigma(\boldsymbol{W}^s \boldsymbol{H}^{(l)} + b^s) \tag{14}$$

其中: σ 为激活函数; l 为多层感知机的层数($l \geqslant 1$) ; $\boldsymbol{W}_f^l$、b_f^l 分别为多层感知机第 $l-1$ 层隐节点到第 l 层隐节点的连接权重和偏置, $\boldsymbol{W}_f^l \in \mathbb{R}^{n_l - 1 \times n_l}$, n_{l-1} 和 n_l 分别为第 $l-1$、l 层隐节点的个数; $\boldsymbol{H}^{(l)}$ 为第 l 层的输出隐向量; $\boldsymbol{W}^s$ 、b^s 分别为多层感知机倒数第二层到最后一层的权重和偏置, $\boldsymbol{W}^s \in \mathbb{R}^{n_l \times 2}$; n_l 为多层感知机的倒数第二层隐节点个数。

1.7 计算模型的损失函数

为了对模型的权重和参数进行学习,本文将对数损失函数作为模型的目标函数,公式表示为

$$L_{target} = -\frac{1}{N}\sum_{(x,y)\in D}^{N}(y \log p(x) + (1-y)\log(1 - p(x))) \tag{15}$$

$$L = L_{target} + \alpha * L_{sample} \tag{16}$$

其中: L_{target}为模型的输出和样本的偏差; L 表示模型的总损失函数; α 为 sample loss 加入到总损失函数的比例; N 为样本的总数; x 为模型的输入; y 为真实样本的标签; $p(x)$ 为模型的预估概率。

2 广告点击率预估实验

为了对 SACSN 模型的预测结果进行评价,本文基于亚马逊图书和电子数据集[22] 这两个公开数据集进行实验。

2.1 数据集及评价指标

本文采用的亚马逊数据集包括亚马逊原始数据集文件和用户对商品的评分文件,数据的统计如表 1 所示,本文将用户对物品的评分作为用户的点击行为。本次实验采用的评估指标为 AUC[23] 。AUC 的计算步骤为: a) 通过混淆矩阵求解真阳率(TPR) 和假阳率(FPR) 的值,得到坐标点对; b) 由不同的坐标点对形成的曲线为 ROC(receiver operating characteristic) 曲线; c) AUC为 ROC 曲线下方的面积。AUC 被认为是 CTR 预估问题的一个重要指标,AUC 越大说明 CTR 预估模型的性能越好。

表 1 亚马逊图书数据集和电子数据集统计

Tab. 1 Statistics of Amazon book and electronic datasets

数据集	用户数	商品个数	商品的类别数	样本总数
亚马逊图书	543 060	367 983	1 601	603 669
亚马逊电子	173 166	63 002	705	192 403

2.2 数据的处理及关键参数设置

用户点击序列主要由样本的物品 ID 和物品的类别 ID 组成。通过将原始样本按照用户和用户的点击时间排序得到用户的点击序列。假设序列长度为 T,将前 $T-1$ 次点击作为历史点击序列,将第 T 次点击的物品作为目标广告,即模型的预测目标。针对每一条样本,从用户未点击的物品序列中按正负比例 1 ∶ 1 随机抽取,从而构成正负样本对。历史点击序列的长度设置为 100,对超出部分进行截取,对不足部分按 0 填充,目标广告特征同样是由物品 ID 和物品的类别 ID 组成,上下文特征为用户点击或购买的时间,用户画像特征为用户的 ID 等。

实验中,将 embedding 的大小设置为 16,Transformer 的 block 设置为 1,多头注意力头数设置为 2,dropout 为 0.2,Transformer 学习率设置为 0.001,优化器为 Adam 优化器[24],Transformer 前向连接层激活函数为 ReLU[25]。

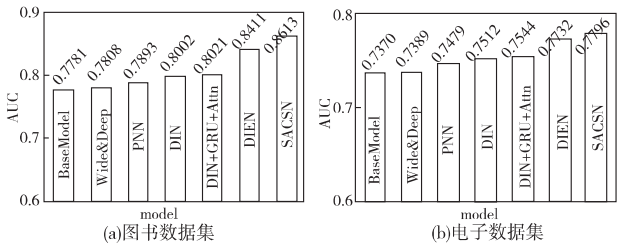
2.3　与其他 CTR 模型的指标对比

从表 2 可以看出,SACSN 在两个数据集上 AUC 得分最高,其中 DIN 对比 BaseModel 在图书和电子数据集中 AUC 分别提升了 2.21%、1.60%,其主要区别为 DIN 比 BaseModel 多了一个注意力机制,通过注意力机制建模用户历史点击和目标广告的相关性。DIN + GRU 比 DIN 分别在图书和电子数据集 AUC 提升了 0.19% 和 0.32%,其主要区别是 DIN + GRU 中使用 GRU 对用户历史点击序列建模,由此可见注意力机制和用户历史点击序列建模是有效的,如图 3 所示。

表 2　典型 CTR 预估模型在两个数据集上的表现

Tab. 2　The overall performance of typical CTR prediction models on two public datasets

模型	AUC	
	图书数据集	电子数据集
BaseModel[14]	0.778 1	0.737 0
Wide&Deep[8]	0.780 8	0.738 9
PNN[7]	0.789 3	0.747 9
DIN[14]	0.800 2	0.751 2
DIN + GRU + Attn	0.802 1	0.754 4
DIEN[26]	0.841 1	0.773 2
SACSN	0.861 3	0.779 6



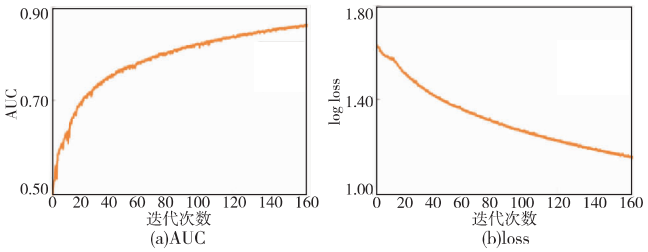
(a)图书数据集　　(b)电子数据集

图 3　典型 CTR 预估模型在亚马逊的图书数据集和电子数据集的表现

Fig. 3　Performance of typical CTR prediction models on Amazon book and electronic dataset

2.4　SACSN 的 AUC 和 loss 的曲线

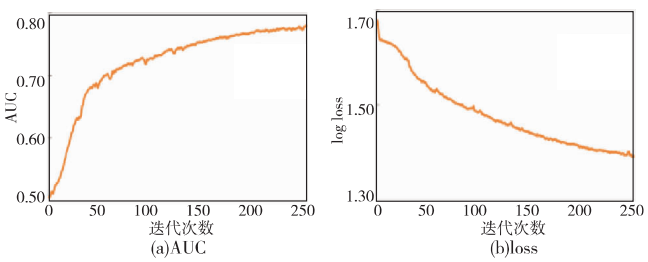
从图 4、5 可以看出,随着迭代次数的增加,AUC 稳步提升、loss 不断下降,其中亚马逊图书数据集和电子数据集分别在前 30、50 次迭代中上升明显,在分别迭代 100、150 次后上升较为缓慢。



(a)AUC　　(b)loss

图 4　SACSN 在亚马逊图书数据集的 AUC 和 loss

Fig. 4　AUC and loss for SACSN on Amazon book dataset



(a)AUC　　(b)loss

图 5　SACSN 在亚马逊电子数据集的 AUC 和 loss

Fig. 5　AUC and loss for SACSN on Amazon electronic dataset

2.5　模型的关键组件分析

从表 3 可以看出,去掉各个组件后指标均有不同幅度的下降,其中去掉 Transformer 和 sample loss 后指标下降最为明显,分别是图书类 AUC 下降 7.61% 和电子类下降 3.30%,说明了 Transformer 结合 sample loss 对于建模用户的历史兴趣是有效的。注意力机制的作用为根据目标广告对用户的历史兴趣序列根据相似度进行了加权,去除后图书类 AUC 下降 0.53%、电子类下降 0.64%。全连接网络主要是增加模型的非线性能力,同时隐式地对特征进行交叉,去除后图书类 AUC 下降 2.23%、电子类下降 1.87%。

表 3　SACSN 关键组件分析

Tab. 3　The key components analysis of SACSN

模型	图书数据集		电子数据集	
	AUC	下降幅度	AUC	下降幅度
SACSN	0.861 3	–	0.779 6	–
去掉注意力层	0.856 0	↓0.53%	0.773 2	↓0.64%
去掉全连接网络	0.839 0	↓2.23%	0.760 9	↓1.87
去掉 Transformer 和 sample loss	0.785 2	↓7.61%	0.746 6	↓3.30

3　结束语

本文针对 CTR 预估精准度不高的问题,提出一个全新的 CTR 预估模型 SACSN。该模型将 Transformer 网络应用于 CTR 预估任务,同时针对 Transformer 建模用户历史兴趣无法有效关联目标广告的问题,提出了在原 Transformer 网络中增加目标广告的多头注意力层,使得 Transformer 不但有效建模用户历史兴趣,而且考虑了与目标广告的关联。同时本文加入辅助损失函数来监督改进后的 Transformer 的训练过程,并加入注意力机制,进一步显式地根据目标广告来反向激活用户历史点击序列,实验证明 SACSN 是有效的。

参考文献:

[1] 周傲英,周敏奇,宫学庆. 计算广告: 以数据为核心的 Web 综合应用[J]. 计算机学报,2011,34(10):1805-1819. (Zhou Aoying, Zhou Minqi, Gong Xueqing. Computation advertising: a data centric comprehensive Web application[J]. Chinese Journal of Computers,2011,34(10):1805-1819.)

[2] 陈杰浩,张钦,王树良,等. 基于深度置信网络的广告点击率预估的优化[J]. 软件学报,2019,30(12):3665-3682. (Chen Jiehao, Zhang Qin, Wang Shuliang, *et al*. Click-through rate prediction based on deep belief nets and its optimization[J]. Journal of Software, 2019,30(12):3665-3682.)

[3] 刘梦娟,曾贵川,岳威,等. 基于融合结构的在线广告点击率预测模型[J]. 计算机学报,2019,42(7):1570-1587. (Liu Mengjuan, Zeng Guichuan, Yue Wei, *et al*. A hybrid network based CTR prediction model for online advertising[J]. Chinese Journal of Computers,2019,42(7):1570-1587.)

[4] Chapelle O, Manavoglu E, Rosales R. Simple and scalable response prediction for display advertising[J]. ACM Trans on Intelligent Systems and Technology,2014,5(4):1-34.

[5] Rendle S. Factorization machines[C]//Proc of IEEE International Conference on Data Mining. Piscataway, NJ: IEEE Press,2010:995-1000.

[6] Juan Yuchin, Zhuang Yong, Chin Weisheng, *et al*. Field-aware factorization machines for CTR prediction[C]//Proc of the 10th ACM Conference on Recommender Systems. New York: ACM Press,2016:43-50.

[7] Qu Yanru, Cai Han, Ren Kan, *et al*. Product-based neural networks for user response prediction[C]//Proc of the 16th IEEE International Conference on Data Mining. Piscataway, NJ: IEEE Press,2016:1149-1154.

[8] Cheng H T, Koc L, Harmsen J, *et al*. Wide & deep learning for recommender systems[C]//Proc of the 1st Workshop on Deep Learning for Recommender Systems. 2016:7-10.

[9] Guo Huifeng, Tang Ruiming, Ye Yunming, *et al*. DeepFM: a factorization-machine based neural network for CTR prediction[C]//Proc of International Joint Conference on Artificial Intelligence. 2017:1-7.

(下转第 2400 页)

clones in entire OS distributions [C]//Proc of IEEE Symposium on Security and Privacy. Piscataway, NJ: IEEE Press, 2012: 48-62.
[8] Sajnani H, Saini V, Svajlenko J, *et al*. SourcererCC: scaling code clone detection to bigcode [C]//Proc of the 38th International Conference on Software Engineering. 2016: 1157-1168.
[9] Kamiya T, Kusumoto S, Inoue K. CCFinder: a multilinguistic token-based code clone detection system for large scale source code [J]. IEEE Trans on Software Engineering, 2002, 28(7): 654-670.
[10] Kim S, Woo S, Lee H, *et al*. VUDDY: a scalable approach for vulnerable code clone discovery [C]//Proc of IEEE Symposium on Security and Privacy. Piscataway, NJ: IEEE Press, 2017: 595-614.
[11] Jiang Lingxiao, Misherghi G, Su Zhendong, *et al*. Deckard: scalable and accurate tree-based detection of code clones [C]//Proc of the 29th International Conference on Software Engineering. Piscataway, NJ: IEEE Press, 2007: 96-105.
[12] Pham N H, Nguyen T T, Nguyen H A, *et al*. Detection of recurring software vulnerabilities [C]//Proc of IEEE/ACM International Conference on Automated Software Engineering. New York: ACM Press, 2010: 447-456.
[13] Li Jingyue, Ernst M D. CBCD: cloned buggy code detector [C]//Proc of the 34th International Conference on Software Engineering. Piscataway, NJ: IEEE Press, 2012: 310-320.
[14] Rattan D, Bhatia R, Singh M. Software clone detection: a systematic review [J]. Information and Software Technology, 2013, 55(7): 1165-1199.
[15] Li Zhen, Zou Deqing, Xu Shouhuai, *et al*. VulPecker: an automated vulnerability detection system based on code similarity analysis [C]//Proc of the 32nd Annual Conference on Computer Security Applications. 2016: 201-213.
[16] Scandariato R, Walden J, Hovsepyan A, *et al*. Predicting vulnerable software components via text mining [J]. IEEE Trans on Software Engineering, 2014, 40(10): 993-1006.
[17] Pang Yulei, Xue Xiaozhen, Namin A S. Predicting vulnerable software components through *n*-gram analysis and statistical feature selection [C]//Proc of the 14th International Conference on Machine Learning and Applications. Piscataway, NJ: IEEE Press, 2015: 543-548.
[18] Russell R, Kim L, Hamilton L, *et al*. Automated vulnerability detection in source code using deep representation learning [C]//Proc of the 17th International Conference on Machine Learning and Applications. Piscataway, NJ: IEEE Press, 2018: 757-762.
[19] Xu Xiaojun, Liu Chang, Feng Qian, *et al*. Neural network-based graph embedding for cross-platform binary code similarity detection [C]//Proc of ACM SIGSAC Conference on Computer and Communications Security. 2017: 363-376.
[20] 夏之阳,易平,杨涛. 基于神经网络与代码相似性的静态漏洞检测[J]. 计算机工程,2019,45(12): 141-146. (Xia Zhiyang, Yi Ping, Yang Tao. Static vulnerability detection based on neural network and code similarity [J]. Computer Engineering, 2019, 45(12): 141-146.)
[21] 李元诚,黄戎,来风刚,等. 基于深度聚类的开源软件漏洞检测方法[J]. 计算机应用研究,2020,37(4): 1107-1110, 1114. (Li Yuancheng, Huang Rong, Lai Fenggang, *et al*. Open source software vulnerability detection method based on deep clustering [J]. Application Research of Computers, 2020, 37(4): 1107-1110, 1114.)
[22] Harer J A, Kim L Y, Russell R L, *et al*. Automated software vulnerability detection with machine learning [EB/OL]. (2018-08-02). https://arxiv.org/abs/1803.04497.
[23] Li Zhen, Zou Deqing, Xu Shouhuai, *et al*. VulDeePecker: a deep learning-based system for vulnerability detection [EB/OL]. (2018-01-05). http://doi.org/10.14722/ndss.2018.23158.
[24] Parr T. ANTLR4 [EB/OL]. [2020-09-05]. https://github.com/antlr/antlr4.
[25] Homeland Security Systems Engineering and Development Institute. Common weakness enumeration [EB/OL]. [2020-09-05]. https://cwe.mitre.org.
[26] Ullah F, Jabbar S, Al-Turjman F. Programmers' de-anonymization using a hybrid approach of abstract syntax tree and deep learning [J]. Technological Forecasting and Social Change, 2020, 159: 120186.
[27] U. S. Commerce Department. Software assurance reference dataset of national institute of standards and technology [EB/OL]. [2020-09-05]. https://samate.nist.gov/SARD.

(上接第2389页)

[10] Wang Ruoxi, Fu Bin, Fu Gang, *et al*. Deep & cross network for Ad click predictions [C]//Proc of the ADKDD. 2017: 1-7.
[11] He Kaiming, Zhang Xiangyu, Ren Shaoqing, *et al*. Deep residual learning for image recognition [C]//Proc of IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE Press, 2016: 770-778.
[12] Lian Jianxun, Zhou Xiaohuan, Zhang Fuzheng, *et al*. xDeepFM: combining explicit and implicit feature interactions for recommender systems [C]//Proc of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM Press, 2018: 1754-1763.
[13] Song Weiping, Shi Chence, Xiao Zhiping, *et al*. Autoint: automatic feature interaction learning via self-attentive neural networks [C]//Proc of the 28th ACM International Conference on Information and Knowledge Management. New York: ACM Press, 2019: 1161-1170.
[14] Zhou Guorui, Zhu Xiaoqiang, Song Chenru, *et al*. Deep interest network for click-through rate prediction [C]//Proc of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM Press, 2018: 1059-1068.
[15] Liu Bin, Tang Ruiming, Chen Yingzhi, *et al*. Feature generation by convolutional neural network for click-through rate prediction [C]//Proc of World Wide Web Conference. 2019: 1119-1129.
[16] Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need [C]//Advances in Neural Information Processing Systems. 2017: 5998-6008.
[17] Kang W C, McAuley J. Self-attentive sequential recommendation [C]//Proc of IEEE International Conference on Data Mining. Piscataway, NJ: IEEE Press, 2018: 197-206.
[18] Sun Fei, Liu Jun, Wu Jian, *et al*. BERT4Rec: sequential recommendation with bidirectional encoder representations from transformer [C]//Proc of the 28th ACM International Conference on Information and Knowledge Management. New York: ACM Press, 2019: 1441-1450.
[19] Zhang Weinan, Du Tianming, Wang Jun. Deep learning over multi-field categorical data a case study on user response prediction [C]//Proc of Conference on Information Retrieval. 2016: 45-57.
[20] Ba J L, Kiros J R, Hinton G E. Layer normalization [EB/OL]. (2016-07-21). https://arxiv.org/abs/1607.06450v1.
[21] Srivastava N, Hinton G, Krizhevsky A, *et al*. Dropout: a simple way to prevent neural networks from overfitting [J]. Journal of Machine Learning Research, 2014, 15(1): 1929-1958.
[22] McAuley J, Targett C, Shi Qinfeng, *et al*. Image-based recommendations on styles and substitutes [C]//Proc of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2015: 43-52.
[23] Lobo J M, Jiménez-Valverde A, Real R. AUC: a misleading measure of the performance of predictive distribution models [J]. Global Ecology and Biogeography, 2008, 17(2): 145-151.
[24] Kingma D P, Ba J. Adam: a method for stochastic optimization [EB/OL]. (2014-12-22). [2017-01-30]. https://arxiv.org/abs/1412.6980.
[25] Nair V, Hinton G E. Rectified linear units improve restricted Boltzmann machines [C]//Proc of International Conference on Machine Learning. 2010: 807-814.
[26] Zhou Guorui, Mou Na, Fan Ying, *et al*. Deep interest evolution network for click-through rate prediction [C]//Proc of AAAI Conference on Artificial Intelligence. 2019: 5941-5948.