融合注意力胶囊的深度因子分解机模型

顾亦然[1,2]，姚朱鹏[1]，杨海根[3]

（1. 南京邮电大学自动化学院、人工智能学院，江苏 南京 210023；
2. 南京邮电大学智慧校园研究中心，江苏 南京 210023；3. 南京邮电大学宽带无线通信技术教育部工程研究中心，江苏 南京 210003）

摘 要：针对深度学习中推荐模型特征组合单一、消解大量有价值特征信息以及过拟合等问题，设计了一种新型的注意力得分机制——注意力胶囊，提出了一种融合注意力胶囊的深度因子分解机模型。基于 DeepFM 模型，将用户历史点击行为与候选物品进行权重计算，降低了无关特征对模型的影响，充分挖掘了不同历史行为对用户兴趣的差异性影响。训练过程中加入自适应正则化式，在不影响训练速度的前提下，有效地减少了过拟合。在 2 个公开数据集上进行对比实验，实验结果表明，所提模型相对于其他模型在损失函数和 GAUC 上均有明显提升。

关键词：推荐模型；深度学习；注意力胶囊；因子分解机；自适应正则化

中图分类号：TP181

文献标识码：A

DOI: 10.11959/j.issn.1000−436x.2021185

Deep factorization machine model based on attention capsule

GU Yiran[1,2], YAO Zhupeng[1], YANG Haigen[3]

1. College of Automation & College of Artificial Intelligence, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
2. Center of Smart Campus Research, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
3. Center of Wider and Wireless Communication Technology, Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

Abstract: Aiming at the problems of single feature combination of recommendation model, resolution of a large amount of valuable feature information, and over-fitting in deep learning, a new attentional scoring mechanism called attention capsule was designed, and a deep factorization machine model based on attention capsule was proposed. Users' historical clicking and candidate items were processed through weight calculation based on the DeepFM model, reducing the impact of irrelevant features on the model, and the differential impact of different historical behaviors on users' interests was fully explored. The adaptive regularization formulation was added to the training, which effectively reduced over-fitting without affecting the training speed. The comparison test on two public data sets shows that the proposed model is significantly enhanced in loss function and GAUC compared to other models.

Keywords: recommendation model, deep learning, attention capsule, factorization machine, adaptive regularization

1 引言

随着互联网时代的到来，计算机技术的高速发展使人们进入了一个信息爆炸的时代。面对海量的信息，用户往往会花费大量的时间和精力去寻找自己所感兴趣的物品，这就产生了信息过载问题，即实际存在的信息远远大于用户所需要的。推荐系统通过用户标签、历史行为、共同好友等因素对用户

收稿日期：2021−03−26；**修回日期**：2021−05−24

基金项目：国防基础科研基金资助项目（No.JCKY2019210B005, No.JCKY2018204B025, No.JCKY2017204B011）；国防重大工程基金资助项目（No.ZQ2019D20401）；装备发展部仿真预研课题（No.41401030301）

Foundation Items: The National Defense Basic Scientific Research Program of China (No.JCKY2019210B005, No.JCKY2018204B025, No.JCKY2017204B011), The Major National Defense Projects of China (No.ZQ2019D20401), The Simulation Pre Research Project of Equipment Development Department of China (No.41401030301)

进行推荐，从而增加用户体验。点击率（CTR, click through rate）预测是推荐系统中最热门的分支。推荐系统通过预测用户点击待推荐物品的概率，对待推荐列表进行排序，将预测概率最高的物品推荐给用户，达到个性化推荐的目的。

随着信息越来越多，系统数据量也越来越大，传统的广义线性推荐模型由于训练开销大、特征交叉能力不足、学习能力弱等劣势，逐渐无法胜任高准确率的推荐任务。为了提高 CTR 模型的预测准确率，深度学习模型开始成为推荐模型的主流，主要以多层感知器（MLP, multi-layer perceptron）为核心。深度学习早期，研究人员主要通过改变神经网络的结构，构建特点各异的推荐模型。Sedhain 等[1]设计了一种单隐层神经网络，将自编码器和协同过滤相结合，利用协同过滤中的共现矩阵，学习用户和物品的低维向量表示，进行预测评分。但是该模型结构较为简单，学习能力不足。He 等[2]提出了将深度神经网络与协同过滤相结合，该模型利用用户向量和物品向量的 Embedding 特征进行特征交叉来代替矩阵分解，解决矩阵分解易欠拟合的问题，但该模型是以协同过滤为核心的，所以特征选取较少，模型表达能力不足。Shan 等[3]提出的 Deep Crossing 模型是 MLP&Embedding 的典型应用，多层残差网络进行多维度的特征组合，但是由于其纯高阶的结构比较单一，无法满足现实中复杂的推荐任务。

CTR 预测任务中主要有 2 种特征交互模式：浅层交互和深层交互[4]。浅层交互指的是那些明显能看出有关联的特征交互，比如下雨和雨伞、饮料和杯子等。而深层交互指的是那些并不容易看出来且需要进行深层次的分析才能找出关联的特征交互，比如下雨和减肥。在现实的推荐系统中，用户特征和物品特征往往十分复杂多样，特征与特征之间的关联也很难做到完美组合，为了提高模型的泛化能力，需要同时考虑浅层交互和深层交互[5]。对此，Guo 等[6]提出了深度因子分解机（DeepFM, deep factorization machine）模型，该模型可自动进行低维特征组合，同时对高维特征进行提取，但该模型所分配的特征权重是固定的，在进行推荐时并未考虑用户的历史行为对用户兴趣的差异性影响，事实上消解了大量有价值的特征信息。例如，应用场景是预测一位 20 岁的女性用户是否购买一款香水，那么“性别=女并且购买历史中包含口红”这一特征远比“性别=女且年龄=20”重要，模型应该赋予前者特征更大的权重，与无关特征的交互会引入噪声甚至降低模型性能。

基于上述分析，本文设计了一种新型的注意力得分机制——注意力胶囊，通过给予不同交叉特征不同的分配权重，解决了不同特征交叉所产生的噪声问题。基于此，本文提出了一种融合注意力胶囊的深度因子分解机（AxDFM, deep factorization machine based on attention capsule）模型。本文的主要工作如下。

1) 设计了一种新型的注意力得分机制，解决了 DeepFM 模型存在的噪声问题，在保证模型泛化能力和训练速度的基础上，充分挖掘了不同历史行为对用户兴趣的差异性影响。

2) 在训练过程中加入自适应正则化式，以减少大规模训练时产生的过拟合影响。

3) 在 Avazu 和 Criteo 这 2 个公开数据集上与主流推荐模型进行比较，验证了所提方法的可行性与有效性。

2　AxDFM 模型介绍

2.1　Embedding 特征表示

CTR 预测的主要任务是给用户推荐其可能感兴趣的物品，用户在进入推荐系统前，并没有表明自己的喜好。所以，在建立 CTR 模型时，需要从用户的个人信息和历史行为中提取用户的兴趣特征[7]。因此，用户的个人信息以及用户历史行为数据的特征表示就显得十分重要，特征表示是 CTR 建模的基本要素。

推荐系统的输入往往具有很多属性特征，其中甚至有部分特征是缺失的，为了能够全面地表示这些特征，one-hot 编码可对其进行表示，但 one-hot 编码极其稀疏，直接进行训练产生的开销太大。因此，Embedding 层被用于对 one-hot 编码进行降维稠密化，由高维稀疏向量转换为低维稠密向量。Embedding 的过程本质上是一层全连接的神经网络。Embedding 网络结构如图 1 所示，输入为一个五维 one-hot 编码向量，接入神经网络与一个三维 Embedding 层连接，虚线所代表的权重即为该 one-hot 编码对应的 Embedding 值。

2.2　DeepFM

DeepFM 是一个典型的并行融合网络结构，由

因子分解机（FM, factorization machine）和深度神经网络（DNN, deep neural network）构成，两者共享用户和物品的 Embedding 层向量。FM 部分负责特征的一阶二阶自动组合，通过学习低阶特征，使模型具有较强的记忆能力。DNN 部分负责高阶特征提取，使模型具有较强的泛化能力[8]。整个模型的输出如式(1)所示。

$$y' = \text{sigmod}(y_{\text{FM}} + y_{\text{DNN}}) \tag{1}$$

其中，$y' \in (0,1)$ 是 CTR 的预测概率；y_{FM} 是 FM 部分的输出；y_{DNN} 是 DNN 部分的输出。

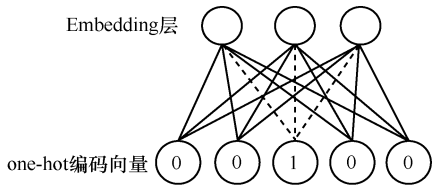


图 1 Embedding 网络结构

FM 模型是 Rendle[9]提出的因子分解机，主要解决了数据稀疏和复杂度上两大缺陷。FM 利用 2 个向量内积取代了单一的权重系数，为每一个特征学习到一个隐向量，特征之间的特征组合权重即为特征的隐向量内积。FM 的提出使即使 2 个特征之间即便没有交互数据，也可以计算两者的相关程度，即

$$\text{FM}(\boldsymbol{V}, \boldsymbol{x}) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \boldsymbol{V}_i \boldsymbol{V}_j \rangle \boldsymbol{x}_i \boldsymbol{x}_j \tag{2}$$

在 DeepFM 中，对于每个特征 i，都有重要程度 w_i 和隐向量 $\boldsymbol{V}_i$ 这 2 个参数，其中，w_i 主要用来衡量特征的一阶重要性；$\boldsymbol{V}_i$ 则用来进行特征组合，用于 FM 的二阶计算和 DNN 的高阶特征组合。FM 模块结构如图 2 所示，其中，Field 为相同性质特征场，是 DeepFM 特征表示的基础。

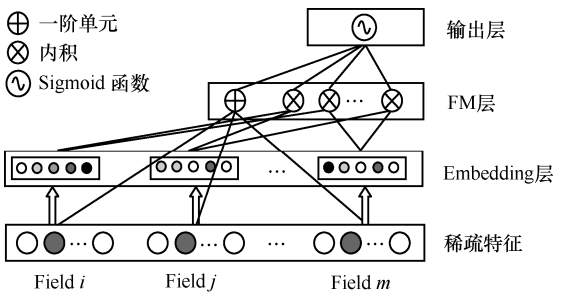


图 2 FM 模块结构

FM 模块的输出为

$$y_{\text{FM}} = \langle w, \boldsymbol{x} \rangle + \sum_{j_1=1}^{d} \sum_{j_2=j_1+1}^{d} \langle \boldsymbol{V}_{j_1} \boldsymbol{V}_{j_2} \rangle \boldsymbol{x}_{j_1} \boldsymbol{x}_{j_2} \tag{3}$$

其中，$\langle w, \boldsymbol{x} \rangle$ 反映了特征之间的一阶重要程度，内积部分反映了特征之间的二阶组合影响。

DNN 部分是一个全连接的前馈神经网络，用来学习用户与物品间的高阶特征组合，DNN 模块结构如图 3 所示。

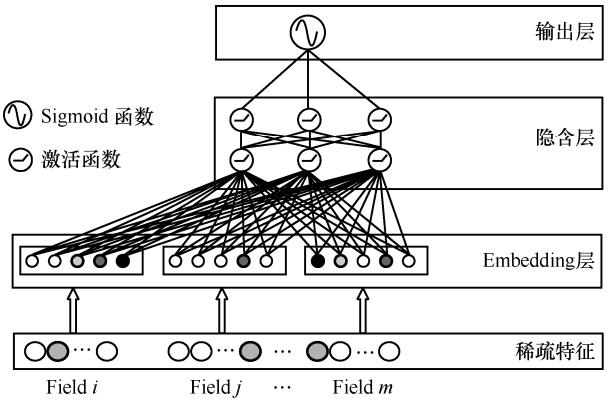


图 3 DNN 模块结构

网络原始输入是高维稀疏的 one-hot 编码，经 Embedding 层转化为低维稠密向量，使网络能够训练。Embedding 层输出为

$$\boldsymbol{a}^{(0)} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_m] \tag{4}$$

其中，m 表示特征域的个数，$\boldsymbol{e}_i$ 代表第 i 个特征域的 Embedding 向量。将 $\boldsymbol{a}^{(0)}$ 输入 DNN 中，则 DNN 的正向传播过程为

$$\boldsymbol{a}^{(l+1)} = \sigma(\boldsymbol{W}^{(l)} \boldsymbol{a}^{(l)} + \boldsymbol{b}^{(l)}) \tag{5}$$

其中，l 是层数，σ 是激活函数，$\boldsymbol{a}^{(l+1)}$ 是第 l 层的输出，$\boldsymbol{W}^{(l)}$ 是模型权重，$\boldsymbol{b}^{(l)}$ 是偏置。DNN 模块的最终输出为

$$y_{\text{DNN}} = \sigma(\boldsymbol{W}^{|H|+1} \boldsymbol{a}^{|H|} + \boldsymbol{b}^{|H|+1}) \tag{6}$$

其中，$|H|$ 是隐含层的层数。

2.3 AxDFM 模型

用户的历史行为在 CTR 预测中起着至关重要的作用。DeepFM 在对用户进行兴趣表示时，将用户的历史行为特征组上的所有 Embedding 向量连接起来，得到一个固定长度的表示向量，如式(4)所示。对于一个给定的用户，由于采用了平均池化，使用户兴趣表示具有一致性与不变性，无论候选物品是

什么，该表示向量均不会变化，即缺乏兴趣表达能力，无法挖掘历史行为对用户兴趣的差异性影响，消解了大量有价值的信息。例如，男生喜欢买球衣球鞋，也喜欢买鼠标耳机，甚至还为自己女朋友购买过香水口红。在实际生活中，当男生在购买键盘的时候，并不需要考虑香水口红这个偏好特征，而男生购买键盘的行为受鼠标耳机的影响远比其余两组特征大。此时，香水口红特征不仅没有对推荐结果产生正向影响，反而消解了鼠标耳机特征的正向影响，变成了推荐系统中的噪声，降低了模型性能。

在上述例子中，整个购买过程如下：候选商品键盘通过对该用户的购买行为进行软搜索，发现该用户购买过鼠标耳机，从而触及了他相关的兴趣。换而言之，与候选物品相关的历史行为对于用户的点击与否有着很大的贡献。考虑到注意力机制可以提升模型的重点内容的学习能力和降低无关特征影响的特性，本文针对用户行为与候选物品的关系程度设计了一种新型的注意力得分机制——注意力胶囊。将注意力胶囊引入 DeepFM 模型中，AxDFM 模型可以在表示向量维度有限的情况下，产生一个可变的、动态的表示向量来对用户兴趣进行表示，即利用候选物品在历史行为中的不同激活程度自适应地改变 DNN 的输入 Embedding 向量。

用户的每一个历史行为都会与候选物品进行权重计算，以自适应地计算候选物品的用户兴趣表示向量，注意力胶囊的网络结构如图 4 所示，具体计算式如式(7)所示。

$$v_U(A)=f(\boldsymbol{v}_A,\boldsymbol{e}_1,\boldsymbol{e}_2,\cdots,\boldsymbol{e}_m)=\sum_{j=1}^{m}g(\boldsymbol{e}_j,\boldsymbol{v}_A)\boldsymbol{e}_j=\sum_{j=1}^{m}\text{weight}_j\boldsymbol{e}_j \tag{7}$$

其中，$\boldsymbol{v}_A$ 是候选物品 A 的 Embedding 向量，$g()$ 是一个前馈网络，weight_j 是激活权重。

注意力胶囊的输入为历史行为和候选物品的 Embedding 向量。引入两者的外积（有助于相关性建模），将三者进行组合拼接，利用一个 3×3 卷积核对其进行卷积，将得到的输出连接单全连接层得到权重大小。本文提出的注意力胶囊抛弃了传统注意力机制中的 Softmax 层[10]，使得到的权重和并不为 1，即 $\sum_i \text{weight}_i \neq 1$。通过放弃 Softmax 的规范化来保留物品的激活程度，即权重和越大，物品与历史行为相关程度就越大，增强了模型的兴趣表达能力。

AxDFM 网络结构如图 5 所示，通过在 Embedding 层后加入注意力胶囊，使用户的历史点击行为与候选物品进行权重计算，得到每个点击行为与候选物品的权重，在形成表示向量时能够更加突出候选物品与历史行为中所相关的物品，可以自适应地生成动态表示向量，从而达到在有限的维度下，增强模型兴趣表达能力的目的。

2.4　自适应正则化

在模型训练过程中，由于训练数据、权重参数过多，过拟合不可避免。过拟合是指模型在训练集上表现良好，在测试集上却表现一般，甚至会随着时间的推移模型效果越来越差，使模型泛化能力较弱[11]，如图 6 所示。

为了减小过拟合的影响，最简单有效的方法是在损失函数后添加正则化项，对高阶权重部分进行惩罚，即

$$\tilde{L}(\boldsymbol{w};\boldsymbol{X},y)=L(\boldsymbol{w};\boldsymbol{X},y)+\lambda f(\boldsymbol{w}) \tag{8}$$

其中，$\boldsymbol{X}$ 为输入样本，y 为对应标签，w 为权重系数，$L()$ 为损失函数，λ 为正则化系数，$f(w)$ 为惩罚项。

CTR 预测中往往具有输入稀疏且维度高的特

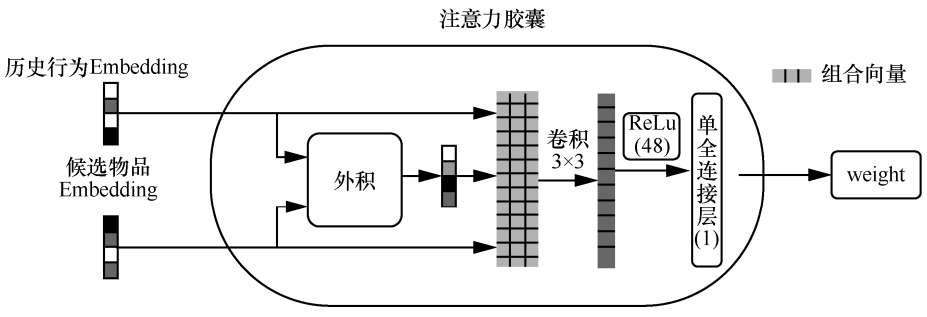


图 4　注意力胶囊网络结构

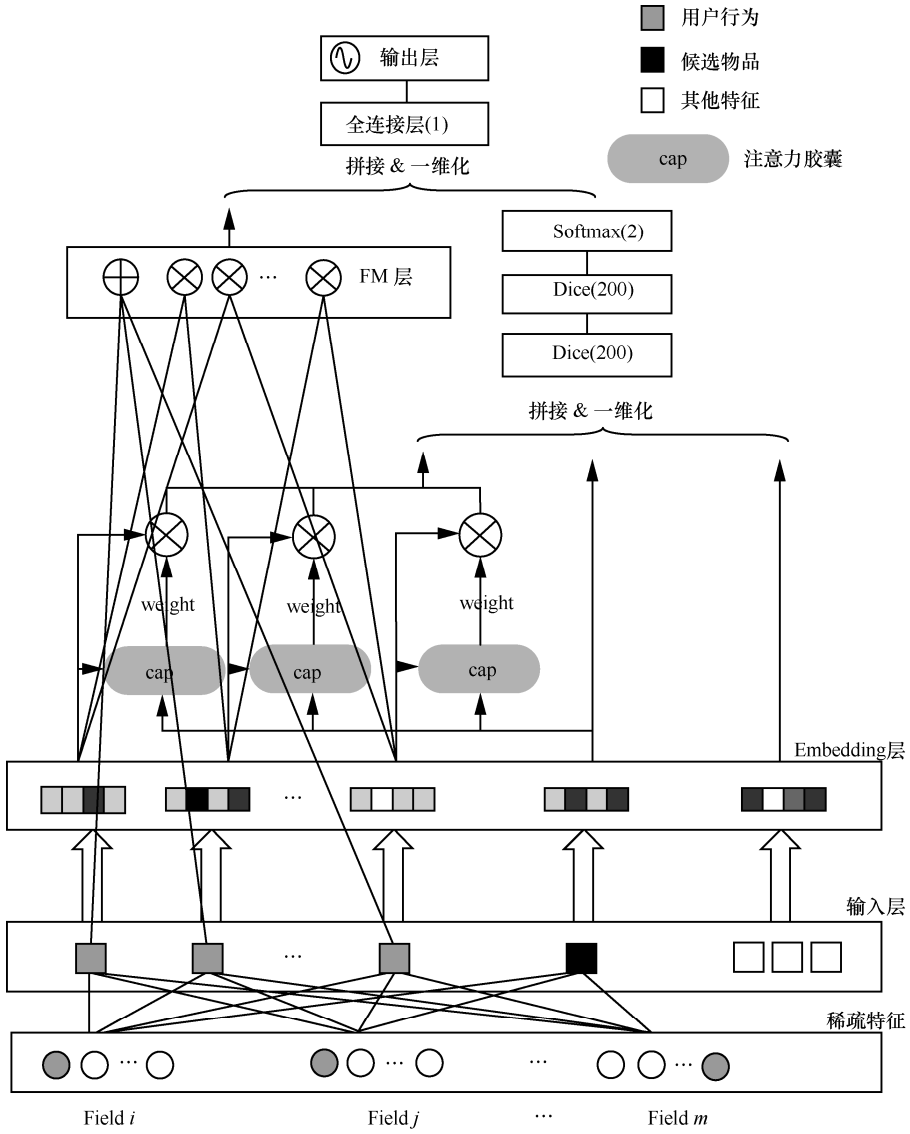


图 5　AxDFM 网络结构

点，在这样一个庞大的网络上直接应用传统的正则化方法显然不现实。以随机梯度下降法（SGD, stochastic gradient descent）为例，在没有进行正则化前，只需更新输入特征中不为 0 的特征所对应的参数。然而，当增加了 L2 正则化后，需要计算全部参数的 L2 范数，这极大地增加了训练的开销，降低了模型的效率。对此，本文使用了一种自适应的正则化式，只计算不为 0 的输入特征所对应参数的 L2 范数，如式(9)所示，判断函数如式(10)所示。

$$L_2(\boldsymbol{W}) = \| \boldsymbol{W} \|_2^2 = \sum_{j=1}^{K} \| \boldsymbol{w}_j \|_2^2 = \sum_{(x,y)\in S} \sum_{j=1}^{K} \frac{I(\boldsymbol{x}_j \neq 0)}{n_j} \| \boldsymbol{w}_j \|_2^2 \tag{9}$$

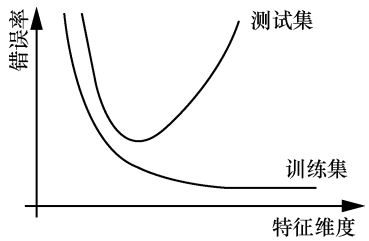


图 6　过拟合

$$I = \begin{cases} 1, \exists (\boldsymbol{x}, y) \in B, \text{s.t.} [\boldsymbol{x}_j] \neq 0 \\ 0, \text{其他} \end{cases} \tag{10}$$

其中，$\boldsymbol{W}$ 为整个输入字典，K 为特征维度，S 为大小为 N 的训练集，$\boldsymbol{x}$ 为网络的输入，$y \in \{0,1\}$ 为点击标签（0 代表未点击，1 代表点击），$\boldsymbol{w}_j$ 为第 j 个向量，I 表示第 i 个样本是否有 j 这个特征，n_j 为整

个样本中 j 特征的出现次数，B 表示样本分割的小批量数。

本文采用的优化算法为小批量梯度下降（MBGD, mini batch gradient descent）法[12]，总样本数可以拆分为多个小批量样本，于是式(9)可以转换为

$$L_2(\boldsymbol{W})=\sum_{j=1}^{K}\sum_{m=1}^{B}\sum_{(x,y)\in B_m}\frac{I}{n_j}\|\boldsymbol{w}_j\|_2^2\approx\sum_{j=1}^{K}\sum_{m=1}^{B}\frac{I}{n_j}\|\boldsymbol{w}_j\|_2^2 \quad (11)$$

其中，B_m 表示第 m 个小批量。

训练的损失函数采用对数似然损失函数，如式(12)所示。

$$L=-\frac{1}{N}\sum_{(\boldsymbol{x},y)\in S}(y\log p(\boldsymbol{x})+(1-y)\log(1-p(\boldsymbol{x}))) \quad (12)$$

其中，y 为样本的真实标签，$p(\boldsymbol{x})$ 为预测输入 $\boldsymbol{x}$ 被点击的概率。

将式(11)和式(12)代入式(8)，可以得到本文最终采用的正则化后的损失函数为

$$\tilde{L}=-\frac{1}{N}\sum_{(\boldsymbol{x},y)\in S}(y\log p(\boldsymbol{x})+(1-y)\log(1-p(\boldsymbol{x})))+\frac{\lambda}{2N}\sum_{j=1}^{K}\sum_{m=1}^{B}\frac{I}{n_j}\|\boldsymbol{w}_j\|_2^2 \quad (13)$$

其中，损失函数的输出区间为 $[0,+\infty)$，其值越小，代表模型的分类工作越好。由于 y 是每个样本的标签（0、1 标签），$p(\boldsymbol{x})$（$p(\boldsymbol{x})\in[0,1]$）是模型对其的预测概率，因此对于每个样本而言，预测值越接近样本标签，损失函数值越接近于 0，即模型预测越准确。

3　实验设计

3.1　实验环境

本文在 Window10 环境下进行实验，代码语言为 Python3.7，深度学习框架为 TensorFlow-GPU 2.1.2，CUDA 版本为 11.2.152，cuDNN 版本为 7.6.5，运行内存为 16 GB，GPU 为 NVIDIA RTX 3070，处理器为 Intel(R) Core(TM) i5-10600KF CPU。

3.2　数据集

本文使用的数据集为 Kaggle CTR 大赛上所使用的两个公开数据集，即 Avazu 和 Criteo。

Avazu 包含了真实的用户点击行为数据，按时间顺序排列，其中训练集是 10 天的点击数据，测试集是一天的点击数据。数据集拥有 4 000 万行数据，23 个特征域（包含用户属性特征、设备特征、广告属性特征以及匿名特征）。

Criteo 是 Criteo 公司的真实数据，按时间顺序排列，其中训练集是 7 天的点击数据，测试集是紧跟着训练集后一天的点击数据。数据集拥有 4 500 万行数据（包含点击标签）、13 个数值特征和 26 个匿名分类特征。

3.3　评价标准

CTR 预测本质上来说是一个二分类问题，即判定用户是否会点击。针对二分类问题，机器学习有一个应用非常广泛的指标——AUC（area under the curve）。AUC 是 ROC（receiver operating characteristic）曲线所围成的面积，范围为[0,1]。对于随机抽取的一对正负样本，本质上来说 AUC 是把正样本预测为 1 的概率大于把负样本预测为 1 的概率的概率，即

$$\mathrm{AUC}=P(P_{\mathrm{ture}}>P_{\mathrm{false}}) \quad (14)$$

其中，P_{ture} 是将正样本预测为 1 的概率，P_{false} 是将负样本预测为 1 的概率。

AUC 值是一个概率值，AUC>0.5 时，将正样本预测为 1 的概率比把负样本预测为 1 的概率大，说明模型有一定的分类能力。在[0,1]的范围内，AUC 越大代表模型性能越好。AUC 计算式为

$$\mathrm{AUC}=\frac{\sum_{\mathrm{ins}_i\in\mathrm{positiveclass}}\mathrm{rank}_{\mathrm{ins}_i}-\frac{M(M+1)}{2}}{MN} \quad (15)$$

其中，$\mathrm{rank}_{\mathrm{ins}_i}$ 为第 i 条样本的序号（概率从小到大排列），M 和 N 分别为正样本的个数和负样本的个数，$\sum_{\mathrm{ins}_i\in\mathrm{positiveclass}}\mathrm{rank}_{\mathrm{ins}_i}$ 为正样本的序号和。

然而，在实际 CTR 预测中，由于用户的个性化程度较高，不同用户间的排序结果对于评价模型性能的意义不大。对此，本文采用了 GAUC（group area under the curve）[13]，对每个用户的 AUC 进行加权平均，可以减小不同用户间的排序结果失真的影响，具体如下

$$\mathrm{GAUC}=\frac{\sum_{i=1}^{n}\mathrm{time}_i\mathrm{AUC}_i}{\sum_{i=1}^{n}\mathrm{time}_i} \quad (16)$$

其中，time_i 表示给用户 i 展示物品的次数。

采用 RelaImpr 衡量模型性能提升百分比，即

$$\mathrm{RelaImpr}=\left(\frac{\mathrm{GAUC}_{\mathrm{measured\_model}}-0.5}{\mathrm{GAUC}_{\mathrm{base\_mdoel}}-0.5}-1\right)\times100\% \quad (17)$$

其中，$GAUC_{measured\_model}$ 为对比模型的 GAUC 值，$GAUC_{base\_mdoel}$ 为基准模型的 GAUC 值。

采用浮点运算数（FLOPS, floating-point operations per second）表示 GPU 计算量，来衡量算法/模型的复杂度。

此外，为了准确评估及对比模型性能，本文采用对数似然损失函数值 Loss 这一指标，如式(13)所示。一般而言，Loss 接近于 0，模型的分类性能越好。

3.4 实验结果

3.4.1 模型性能对比

为了验证本文所提的融合注意力胶囊的深度因子分解机模型的可靠性，本节在 GAUC、RelaImpr 和 Loss 这 3 个指标上，将所提模型和以下模型进行了比较。

LR（logistic regression）[14]：传统线性模型。

DeepCrossing[3]：采用多层残差网络实现 MLP，利用带残差连接的多层全连接神经网络捕捉到更多的非线性特征和组合特征。

DeepFM[6]：由 FM 和 DNN 两部分构成，分别进行低阶与高阶特征组合。

AFM（attentional factorization machines）[15]：在 NFM 基础上引入注意力机制，在 NFM 的特征交叉池化层与输出层之间加入一层基于注意力机制的池化层，用以区分特征之间的不同重要性。

DeepFM_Multi-head：在 DeepFM 模型中加入多头注意力机制[16]，将本文提出的注意力胶囊与多头注意力机制进行对比。

xDeepFM（extreme deep factorization machine）[17]：提出了一种新的压缩交叉网络，以显示方式进行向量级的特征交互，可以隐式学习任意的低阶与高阶特征组合。

所提 AxDFM 模型的主要参数设置如下：深度神经网络部分采用三层全连接层，网络结构为 200-200-2；优化器为 MBGD；batch-size 设置为 512；激活函数选取 Dice，可根据数据分布灵活调整阶跃变化点；Embedding-size 设置为 40；学习率设置为 0.001；Epoch 设置为 10；注意力胶囊层维度设置为 48；正则化式采用自适应正则化函数，正则化系数为 0.01。

为减小过拟合的影响，实验中剔除一些无关标签（Avazu 中的 device_ip 和 device_type，Criteo 中的 C20 和 C22）。表 1 显示了在数据集 Criteo 和 Avazu 上，选取前 100 万份数据，本文提出的 AxDFM 模型和其他 6 种模型的对比（其中 LR 是线性模型，其余均为深度学习模型），实验重复 10 次，GAUC 取 10 次的平均值，RelaImpr 反映了模型相较于 DeepFM 的提升。从表 1 可以得到以下结论。

1) 所有的深度学习模型的结果均优于 LR 模型。LR 模型是这 7 种模型里唯一不考虑特征组合的模型，其性能表现最差，由此可以证明学习特征组合可以提高 CTR 预测模型的性能，也证明了深度学习的可行性。

2) 纯高阶特征组合模型不如低阶−高阶特征组合模型。DeepCrossing 模型是经典的高阶特征组合的深度模型，在性能表现上不如低阶−高阶特征组合模型。

3) 注意力机制的加入可以提高模型性能。AFM、DeepFM_Multi-head 和本文的提出的 AxDFM 引入了注意力机制，三者表现均优于其基础模型。

不同模型训练过程中的损失函数曲线如图 7 所示。从图 7 中可以看出，LR 特征学习能力较弱，故数值较大；加入多头注意力机制的 DeepFM_Multi-head 在高阶特征映射时易导致学习

表 1 **不同模型在数据集 Criteo 和 Avazu 上的对比**

模型	Criteo		Avazu	
	GAUC	RelaImpr	GAUC	RelaImpr
LR	0.768 3	−6.48%	0.731 4	−9.68%
DeepCrossing	0.774 1	−4.46%	0.735 1	−8.24%
DeepFM	0.786 9	0	0.745 2	0
AFM	0.778 5	−2.93%	0.740 3	−2.01%
DeepFM_Multi-head	0.790 3	1.19%	0.748 7	1.43%
xDeepFM	0.800 2	4.64%	0.762 1	6.89%
AxDFM	**0.809 3**	**7.81**%	**0.761 2**	**6.52**%

精度误差，故产生了一个较大波动；AxDFM 收敛速度较快，Loss 较其余 6 种模型保持着较低的水平，收敛值约为 0.446 4，整体表现最优。

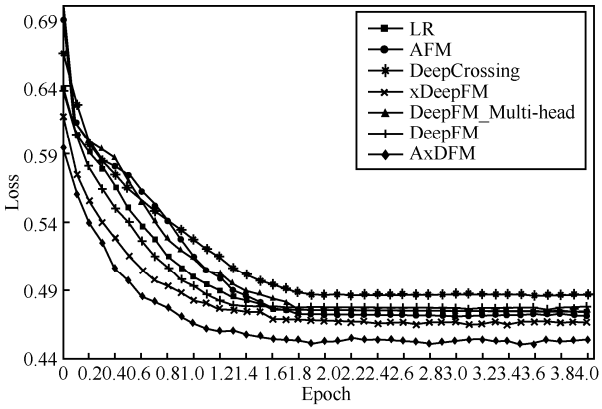


图 7　损失函数曲线

综上所述，AxDFM 通过引入注意力胶囊计算候选物品和用户历史行为的权重，突出了用户历史行为对候选物品的影响，增强了模型的兴趣表达能力，提高了 CTR 预测的准确性和可靠性。

3.4.2　模型复杂度对比

由于本文提出的 AxDFM 模型的时间成本主要在深度神经网络中，因此，本节实验主要对比包含深度神经网络的模型。MFLOPS 为百万次的浮点运算，Time 为不同模型训练 100 万条数据的时间，具体实验结果如表 2 所示。

综合表 1 和表 2 可以看出，AxDFM 在增加 5.2% 的训练开销后，获得了最大 7.81%的模型性能提升。xDeepFM 由于引入了压缩交叉单元，使模型复杂度大大提升，训练开销也随之增大。虽然 xDeepFM 在 Avazu 数据集上的 GAUC 略优于 AxDFM，但前者复杂度过高。综合考虑模型复杂度和性能提升，AxDFM 在这 5 种模型中表现最优。

3.4.3　正则化式对比

在实际 CTR 中，模型的输入是极高维与极稀疏的，且样本数是亿级的，如果不经过正则化处理，模型性能将在一次完整迭代后迅速下降。因此，针对正则化式，本文在完整的 Criteo 数据集上进行了实验，选取 AxDFM 作为基准实验模型，正则化参数设置为 0.01，并且与以下几种正则化式进行对比，证明所提出自适应正则化式的可行性。

L1 正则化：L1 正则化式为权值绝对值之和。

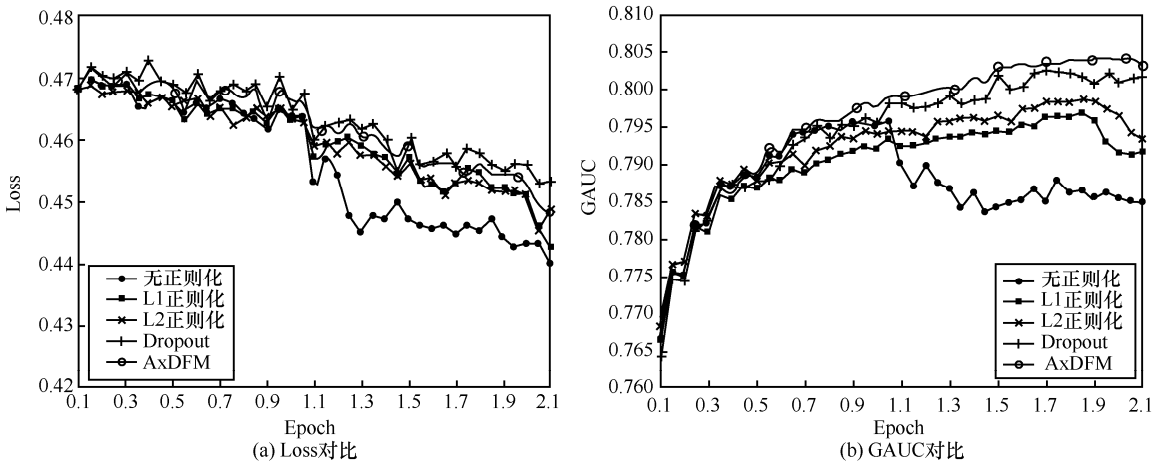
L2 正则化：L2 正则化式为权值绝对值平方和。

Dropout[18]：随机丢弃样本中 50%的特征。

图 8 为不同正则化式在 Criteo 数据集上的 Loss

表 2　　**不同模型的复杂度和运行时间对比**

模型	MFLOPS	Time/s
DeepCrossing	1.93	161
DeepFM	2.09	173
DeepFM_Multi-head	2.33	187
xDeepFM	3.27	249
AxDFM	**2.21**	**182**



(a) Loss对比　　(b) GAUC对比

图 8　不同正则化式在 Criteo 上的 AxDFM 模型性能对比

和 GAUC 对比。不加正则化式的模型在每迭代一次之后，模型的 Loss 和 GAUC 迅速下降，过拟合发生。L1 和 L2 正则化虽然能在一定程度上缓解过拟合，随着迭代次数的增加，模型的性能受过拟合的影响程度增大。Dropout 虽然可以防止快速过拟合，但是 Dropout 收敛速度较慢。本文的自适应正则化方法表现最好，在有效防止过拟合的同时，还保持着一定的收敛速率。

图 9 为不同正则化式在 Criteo 数据集中前 1 000 万份数据的训练时间对比。从图 9 可以看出，不加正则化式的训练时间最短，AxDFM 次之。L1 和 L2 正则化都需要对所有权重进行计算，训练时间显著增加。Dropout 虽然随机丢弃了 50%的样本，但是只是让神经元失活，即变为 0，并且由于训练网络的每个单元要添加一道概率流程，收敛到全局最优的时间变长，因此训练时间大大增加。

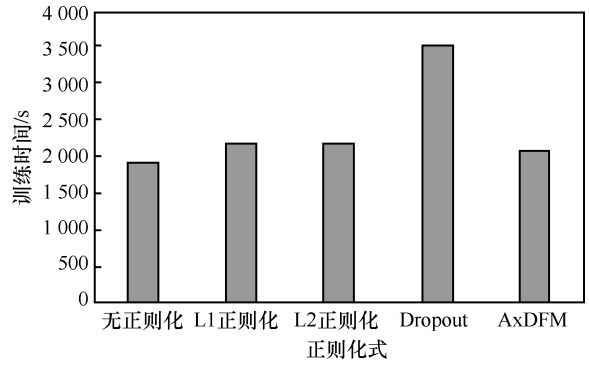


图 9 不同正则化式训练时间对比

综上所述，自适应正则化模型相比于无正则化模型，在增加 8.4%的时间基础上（其余分别为 13.9%、13.4%和 83.5%），极大地减少了过拟合的影响，提升了模型的分类能力。与其他正则式相比，AxDFM 在有效防止过拟合的同时，还保持着较快的收敛速率。

4 结束语

本文设计了一种新型的注意力得分机制——注意力胶囊，提出了一种融合注意力胶囊的深度因子分解机模型。注意力胶囊的引入使该模型不仅可以对输入特征同时进行低阶与高阶组合，还可以根据不同的候选物品生成不同的兴趣表示向量，在保证了模型的记忆与泛化能力的同时，大大提高了模型的兴趣表达能力，挖掘了不同历史行为对兴趣的差异性影响。此外，利用自适应正则化式，使模型在训练过程中有效地减少了过拟合的影响，并保证了训练效率。在 2 个公开数据集上进行了对比实验，验证了 AxDFM 的可行性与有效性。在未来的研究中，考虑不必将用户所有的行为记录压缩进一个向量，只选取部分行为记录从而进一步减少模型训练时间。

参考文献：

[1] SEDHAIN S, MENON A K, SANNER S, et al. AutoRec: autoencoders meet collaborative filtering[C]//Proceedings of the 24th International Conference on World Wide Web. New York: ACM Press, 2015: 111-112.

[2] HE X N, LIAO L Z, ZHANG H W, et al. Neural collaborative filtering[C]//Proceedings of the 26th International Conference on World Wide Web. New York: ACM Press, 2017: 173-182.

[3] SHAN Y, HOENS T R, JIAO J, et al. Deep crossing: Web-scale modeling without manually crafted combinatorial features[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2016: 255-262.

[4] QU Y R, CAI H, REN K, et al. Product-based neural networks for user response prediction[C]//Proceedings of 2016 IEEE 16th International Conference on Data Mining (ICDM). Piscataway: IEEE Press, 2016: 1149-1154.

[5] WANG R X, FU B, FU G, et al. Deep & cross network for ad click predictions[C]//Proceedings of the 23nd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM Press 2017: 1-7.

[6] GUO H, TANG R, YE Y, et al. DeepFM: a factorization-machine based neural network for CTR prediction[C]//Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. New York: ACM Press, 2017:1725-1731.

[7] ZHU H, LI X, ZHANG P Y, et al. Learning tree-based deep model for recommender systems[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM Press, 2018: 1079-1088.

[8] CHEN Q W, ZHAO H, LI W, et al. Behavior sequence transformer for e-commerce recommendation in Alibaba[C]//Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data. New York: ACM Press, 2019: 1-4.

[9] RENDLE S. Factorization machines[C]//Proceedings of 2010 IEEE International Conference on Data Mining. Piscataway: IEEE Press, 2010: 995-1000.

[10] YUAN W H, WANG H, YU X M, et al. Attention-based context-aware sequential recommendation model[J]. Information Sciences, 2020, 510: 122-134.

[11] GHASEMIAN A, HOSSEINMARDI H, CLAUSET A. Evaluating overfit and underfit in models of network community structure[J]. IEEE Transactions on Knowledge and Data Engineering, 2020, 32(9): 1722-1735.

[12] WU D R, YUAN Y, HUANG J, et al. Optimize TSK fuzzy systems for

regression problems: minibatch gradient descent with regularization, DropRule, and AdaBound (MBGD-RDA)[J]. IEEE Transactions on Fuzzy Systems, 2020, 28(5): 1003-1015.

[13] ZHU H, JIN J Q, TAN C, et al. Optimized cost per click in Taobao display advertising[C]//Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2017: 2191-2200.

[14] QUEVEDO J R, MONTAÑÉS E, RANILLA J, et al. Ranked tag recommendation systems based on logistic regression[M]. Berlin: Springer, 2010.

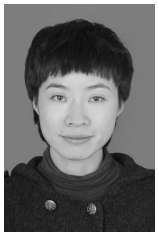
[15] XIAO J, YE H, HE X N, et al. Attentional factorization machines: learning the weight of feature interactions via attention networks[C]// Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. New York: ACM Press, 2017: 3119-3125.

[16] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems. Massachusetts: MIT Press, 2017:5998-6008.

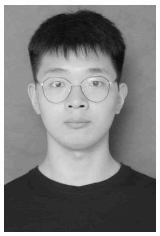
[17] LIAN J X, ZHOU X H, ZHANG F Z, et al. xDeepFM: combining explicit and implicit feature interactions for recommender systems[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM Press, 2018: 1754-1763.

[18] SRIVASTAVA N, HINTON G, KRIZHEVSKY A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

[作者简介]



顾亦然（1972− ），女，江苏南京人，博士，南京邮电大学教授、硕士生导师，主要研究方向为复杂网络、大数据处理等。



姚朱鹏（1997− ），男，江苏苏州人，南京邮电大学硕士生，主要研究方向为推荐算法、自然语言处理等。

杨海根（1983− ），男，江苏南京人，博士，南京邮电大学副教授、硕士生导师，主要研究方向为无线通信、虚拟论证、虚拟设计等。