DOI: 10.3785/j.issn.1008-973X.2023.07.006

特征融合与分发的多专家并行推荐算法框架

杨哲[1,2]，葛洪伟[1,2]，李婷[1,2]
(1. 江南大学 人工智能与计算机学院，江苏 无锡 214122；2. 江苏省模式识别与计算智能工程实验室，江苏 无锡 214122)

摘 要：为了解决点击率预测任务中现存的参数共享和计算耗费较高的问题，提出特征融合与分发的多专家并行推荐算法框架. 利用该方法不仅可以提高并行架构对不同类型特征的分辨能力，学习表现力更强的特征输入，还能够在显式特征和隐式特征之间进行参数共享，缓和反向传播期间的梯度，提高模型的性能. 该框架是轻量级而且与模型无关的，可以泛化应用在众多主流并行架构的推荐算法上. 在 3 个公共数据集上的大量实验结果表明，利用该算法框架，能够有效地提高 SOTA 模型的性能.

关键词：推荐系统；点击率预测；深度学习；多专家模型
中图分类号：TP 391　　**文献标志码：**A　　**文章编号：**1008−973X(2023)07−1317−09

Framework of feature fusion and distribution with mixture of experts for parallel recommendation algorithm

YANG Zhe[1,2], GE Hong-wei[1,2], LI Ting[1,2]
(1. *School of Artificial Intelligence and Computer Science*, *Jiangnan University*, *Wuxi* 214122, *China*;
2. *Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence*, *Wuxi* 214122, *China*)

Abstract: A mixture of experts parallel recommendation algorithm framework which combined feature fusion and distribution was proposed in order to address the issues of parameter sharing and high computational costs in click-through rate prediction. The ability of parallel architecture can be improved to distinguish different types of features and learn more expressive feature inputs, and parameters between explicit and implicit features can be shared. The gradients during backpropagation were mitigated and the performance of the model was improved. The framework is lightweight and model-agnostic, and can be generalized to a variety of mainstream parallel recommendation algorithms. Extensive experimental results on three public datasets demonstrate that the algorithm framework can be used to effectively improve the performance of SOTA models.

Key words: recommender system; click-through rate prediction; deep learning; mixture of experts

点击率预测 (click-through rate, CTR) 任务是预测用户点击广告的概率，在工业应用中十分重要，比如推荐系统或在线广告. 模型的性能和预测结果与广告商利润有着最直接的关联，对后续下游任务比如推荐排序算法、重排算法和广告替换等决策有着重要的参考意义.

当前 CTR 模型中存在以下 3 个问题. 1)Embedding 占用资源及计算耗费较高，Embedding 可以将原始高度稀疏的输入数据映射到低维密集空间中，大型数据集中每个特征的非重复值数量为千万级别，Embedding 维度设置过高会导致占用很大的内存或显存资源，导致计算耗费昂贵. 2)并行架构 Embedding 输入部分参数过度共享，导致输入到并行架构中的特征信息无任何可分辨性. 对于不同的特征建模方式，关注的特征信息不同，因此不是所有特征对该建模方式有意义[1]. 并行架构的 Embedding 输入部分应该有所区分，要训练出更匹配建模方式的特征输入. 3)并行架

收稿日期：2022−07−25.　　**网址：**www.zjujournals.com/eng/article/2023/1008-973X/202307006.shtml
基金项目：国家自然科学基金资助项目 (61806006)；江苏高校优势学科建设工程资助项目；111 引智计划资助项目 (B12018).
作者简介：杨哲(1998—)，男，硕士生，从事推荐系统的研究. orcid.org/0000-0002-8252-6625. E-mail: yz9909@qq.com
通信联系人：葛洪伟，男，教授，博导. orcid.org/0000-0002-1413-0303. E-mail: ghw8601@163.com

构子网络部分参数共享不足. 显式建模和隐式建模部分独立计算，这 2 个部分只有在最后计算结束的时候才会进行信息融合. Hu 等[2] 的研究表明，并行架构计算部分因缺乏共享参数而无法捕捉不同特征语义的相关性，在反向传播期间容易出现梯度较陡的情况.

本研究提出轻量级且高性能的多专家并行推荐算法框架 (mixture of experts for parallel recommendation algorithm framework, ME-PRAF)，其核心组件为 Fusion 模块和 Broker 模块. Fusion 模块用于在显式建模层和隐式建模层之间建立连接，融合显式特征和隐式特征的关联信息，解决参数共享不足的问题. Broker 模块用于学习表现力更强的低维度 Embedding 输入，分别为显式建模层和隐式建模层训练具有分辨性和个性化的特征信息，解决参数过度共享的问题. 由于 Fusion 模块与 Broker 模块的轻量级和高性能特性，在 3 个公共数据集上的大量实验结果表明，利用该算法框架，能够有效地提高 SOTA 并行架构算法模型的性能.

1 相关工作

1.1 并行架构与串行架构

在研究早期，学者们通常手工刻画所有特征，导致模型过拟合很难泛化[3]. 使用线性模型、支持向量机及因子分解机[4] 等方法训练 CTR 模型，但是都只能建模低阶特征信息. 大规模数据集都隐含用户和用户、用户和物品以及物品与物品之间的高阶特征关联[5]，因此有必要对数据集中的高阶特征关联建模[6]. 近年来，学者提出众多深度神经网络来建模高阶特征关联，以端到端的方式捕捉特征信息，无须繁琐地手动刻画特征. 大部分模型使用多层感知机 (multilayer perception, MLP) 建模隐式高阶特征关联. Beutel 等[7] 的研究表明，MLP 在建模 2 阶或 3 阶特征时的交叉效果较差，且隐式建模的方式导致模型的可解释性较差，因此大部分 CTR 算法将显式建模和隐式建模 2 个模块搭配使用. 根据 2 个模块不同的组织方式，可以分为串行架构和并行架构. 如图 1 所示，串行架构是显式建模网络后连接隐式建模网络，PIN[8]、DIN[9] 和 DIEN[10] 等算法属于这种架构；并行架构中，两者独立进行计算，最终将两者输出融合，比如算法模型 DCN[11]、AutoInt+[12] 和 DCN-v2[6] 等. 在实际的工业生产环境中，通常使用多 GPU 进行训练，并行架构能够充分利用多 GPU 资源，相比于串行架构可以节约训练时间，因此本文主要关注对并行架构的优化.

1.2 特征关联

如何有效建模特征关联是 CTR 任务的关键，同时利用显式特征和隐式特征是当前主流 CTR 模型的核心思想. 根据处理显式特征和隐式特征的模块组织方式不同，分为串行架构和并行架构. 本文只关注并行架构，众多 CTR 模型中都是使用 MLP 来建模隐式特征关联，因此不作过多详述. Cheng 等[3] 提出 DeepFM 算法，通过因子分解机学习低阶显式特征关联，但只能学习二阶显式特征关联，无法捕捉更高阶信息. DCN[11] 算法使用特征交叉网络显式建模有限阶特征关联，计算更高效. DeepFM[13] 使用压缩感知层，以 vector-wise 的方式进行特征交叉，但参数量大且计算复杂度高.

AutoInt+[12] 使用多头自注意力机制构建显式特征关联，训练后的注意力权重矩阵具有较好的模型可解释性. DCN-v2[6] 使用权重矩阵替换 DCN 中的权重向量，可以捕捉不同语义子空间下的特征关联.

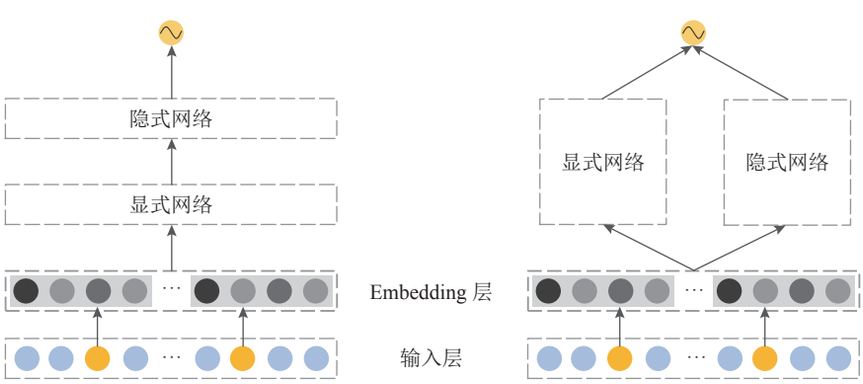


图 1 串行架构和并行架构的示意图

Fig.1 Illustration of sequential and parallel architecture

1.3 并行架构的优化

学者们对并行架构提出很多优化方案. 在多模态训练任务中, 针对模型只对浅层和输出层进行特征融合的问题, DMF[2] 算法使用并行架构中的每一层都进行特征融合, 用于捕捉不同模态任务之间的关联程度, 充分挖掘不同任务之间的特征关联信息. 对于并行架构中只能手工选取输入特征的问题, AutoFeature[1] 使用自动寻找重要特征关联的方法, 为模型输入选取具有侧重点的特征信息, 忽略次要冗余的特征信息. GateNet[14] 使用 Embedding Gate 选取重要潜在特征信息, 通过使用 Hidden Gate, 可以使 MLP 自适应选取隐式特征传给下一层, 但对并行架构输入是无差别的. EDCN[15] 使用 bridge 和 regulation 模块解决参数共享的问题, regulation 模块使用门控网络为并行架构学习不同特征的输入, 但是只提供一种解决方案, 无法捕捉单一特征在不同情况下的多语义信息, 因此实验效果不理想. 在多任务模型中, 多门多专家系统 (multi-gate mixture of experts, MMoE)[16] 通过学习不同任务之间的联系和差异来提高模型质量, 使用门控网络学习多个任务之间的关联, 最大化各种策略对模型的提升价值. 本文使用 MMoE 对 CTR 任务进行更细粒度的划分, 提出 ME-PRAF 框架来学习不同建模任务之间的关联, 训练性能更高的推荐算法模型, ME-PRAF 整体网络架构如图 2 所示.

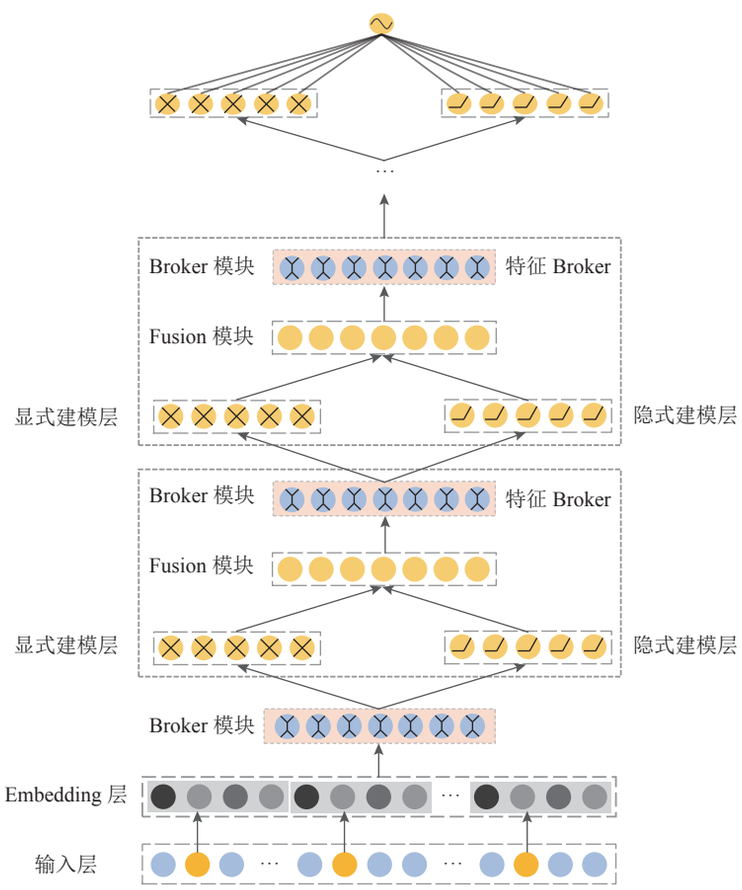


图 2 多专家并行推荐算法框架的整体示意图

Fig.2 Illustration overall architecture diagram of ME-PRAF

2 ME-PRAF 框架

2.1 输入层和 Embedding 层

输入层将用户属性和物品属性聚合, 把所有特征拼接后组成高维稀疏向量:

$$\boldsymbol{e}=[\boldsymbol{e}_1,\boldsymbol{e}_2,\cdots,\boldsymbol{e}_h]. \tag{1}$$

式中: h为特征的数量, $\boldsymbol{e}_i\in\mathbf{R}^{v_i}$表示第$i$个特征. 如果$\boldsymbol{e}_i$是类别型数据, 则为 one-hot 向量; 如果是数值型数据, 则为标量.

由于类别型特征非常稀疏而且维度较高, 常见的处理方式是使用 Embedding, 将高维稀疏的特征映射到低维密集的空间中. 对于输入数据中的每个类别型特征, 使用低维向量进行表示:

$$\boldsymbol{x}_{\text{embed},i}=\boldsymbol{W}_{\text{embed},i}\boldsymbol{e}_i. \tag{2}$$

对于数值型特征，直接取原数值，将所有的特征拼接起来得到：

$$\boldsymbol{x}_0=[\boldsymbol{x}_{\text{embed},1},\cdots,\boldsymbol{x}_{\text{embed},m},\boldsymbol{x}_{\text{num},1},\cdots,\boldsymbol{x}_{\text{num},n}]. \tag{3}$$

式中：$\boldsymbol{x}_{\text{embed},i}\in\mathbf{R}^{u_i}$为第$i$个类别型特征对应的低维Embedding向量，$\boldsymbol{x}_{\text{num},j}$为第$j$个数值型特征标量；$\boldsymbol{W}_{\text{embed},i}\in\mathbf{R}^{u_i\times v_i}$为可训练的映射权重矩阵，其中$u_i\ll v_i$；最终Embedding层输出为$\boldsymbol{x}_0\in\mathbf{R}^d$. 若类别型特征是多值变量，则取所有对应Embedding向量的平均值作为最终向量.

对于基于注意力机制的模型，由于需要训练不同特征之间的注意力权重矩阵，须对数值型特征进行进一步的处理，将其从标量转为与类别型特征相同维度的向量：

$$\boldsymbol{x}_{\text{num},j}=\boldsymbol{v}_{\text{num},j}e_j. \tag{4}$$

式中：$\boldsymbol{v}_{\text{num},j}$为对第$j$个数值型特征的可训练映射权重向量，$e_j$为第$j$个数值型特征标量.

2.2 Fusion 模块

在当前现存的推荐算法并行架构中，主流深度CTR模型使用2个子网络，分别对显式特征关联和隐式特征关联进行建模. 2个网络之间独立进行训练，只在2个子网络输出层进行特征融合. 这种特征融合策略只能捕捉语义级别的关联，无法捕捉中间层显式特征和隐式特征之间的关联. 在2个独立子网络反向传播期间，会存在梯度较高、导致模型过拟合的问题，这是导致模型性能变差的原因之一. 在人体大脑结构中，生物认知科学家发现多器官感知不仅存在于大脑颞叶，而且存在于额叶和顶叶中[17]. 这意味着信息融合应该在信息处理中间阶段开展，用于捕捉不同特征类型之间更复杂的关联.

为了解决上述问题，使用密集融合(dense fusion)的策略构建Fusion模块. 对2个独立子网络中的每一层输出进行信息融合，充分捕捉显式特征和隐式特征之间的关联，缓和反向传播期间的梯度.

在ME-PRAF中，令$\boldsymbol{x}_l$和$\boldsymbol{h}_l$分别表示第l层显式建模层和隐式建模层的输出，使用$\boldsymbol{\alpha}_l=f(\boldsymbol{x}_l,\boldsymbol{h}_l)$表示Fusion模块的输出，其中$f(\cdot):\mathbf{R}^d\times\mathbf{R}^d\to\mathbf{R}^{d'}$表示对于显式特征和隐式特征融合方式，对输入的要求是两者维度相同. 提出使用以下3种融合方式.

1）拼接. 使用最简单的融合方式，将显式建模层和隐式建模层每一层的输出直接进行拼接：

$$\boldsymbol{\alpha}_l=[\boldsymbol{x}_l,\boldsymbol{h}_l]. \tag{5}$$

2）按位加. 将2个相同维度的向量进行加法计算：

$$\boldsymbol{\alpha}_l=\boldsymbol{x}_l\oplus\boldsymbol{h}_l. \tag{6}$$

3）Hardmard积. 将2个相同维度的向量对应元素进行乘法计算：

$$\boldsymbol{\alpha}_l=\boldsymbol{x}_l\otimes\boldsymbol{h}_l. \tag{7}$$

Fusion模块用于融合同一层显式特征和隐式特征之间的层级关联，当多个Fusion模块叠加时能够融合不同层之间更复杂的关联信息，极大改善了并行架构中参数共享不足的问题. 3种融合方式的对比在3.6节的实验中给出.

2.3 Broker 模块

在现存的并行架构CTR模型中，使用完全一致的Embedding作为输入进行计算，然而不同建模方式对特征信息的关注点不同，应该采取因地制宜的策略. DCN-v2中交叉网络是通过显式建模的方式来高效捕捉有限阶特征关联，MLP网络是用来建模高阶隐式特征. 2种方式对特征建模的角度不同，为不同的子网络学习具有可分辨性的特征输入.

受到MMoE中多任务学习的启发，将CTR任务进行更细粒度、更精细化的划分，提出使用Broker模块对模型中的子网络训练专有的特征输入. 如图3所示为Broker模块的内部结构. 根据使用场景的不同，Broker模块分为Embedding Broker和Feature Broker. 前者用于解决模型输入参数过度共享的问题，为并行架构中不同子网络学习更具有分辨性的、个性化的特征输入. 后者用于配合Fusion模块，对融合后的数据进行训练并且拆分为2个数据流，为子网络下一层提供个性化的输入，捕捉显式特征和隐式特征之间的关

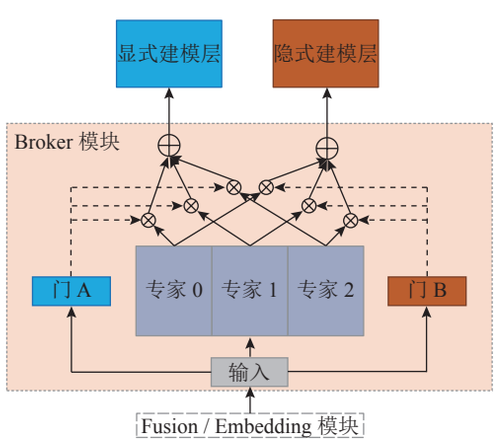


图 3 Broker 模块的内部结构

Fig.3 Internal structure of Broker module

联, 多层叠加还可以学习高阶和低阶特征之间的关联.

设置 2 个独立的门控网络, 分别对应 2 个细粒度的任务: 建模显式特征关联和建模隐式特征关联. 对于任务 k, 输出为

$$y^k = \sum_{i=1}^{n} g^k(\boldsymbol{x})_i f_i(\boldsymbol{x}). \tag{8}$$

式中: $f_i(\cdot)$表示第i个专家的输出; $g^k(\cdot)_i$表示对于任务k对应门控网络输出的第i个分量, 用于表示选取第i个特性的概率, 有$\sum_{i=1}^{n} g^k(\boldsymbol{x})_i = 1$; y^k为对应任务k的输出结果; n为专家的数量. 每个门控网络都是由相同的线性模型组成, 使用 softmax 得到选择对应专家的概率:

$$g^k(\boldsymbol{x}) = \text{softmax}\,(\boldsymbol{W}_{gk}\boldsymbol{x}). \tag{9}$$

式中: $\boldsymbol{W}_{gk} \in \mathbf{R}^{n\times d}$为任务$i$的可训练矩阵. 对专家函数的定义可以是线性模型、MLP 或者是自定义函数, 本文定义为线性模型, 经过 Batch Normalization 处理, 可得

$$f_i(\boldsymbol{x}) = \text{BatchNorm}\,(\boldsymbol{W}_{ei}\boldsymbol{x} + \boldsymbol{b}_{ei}). \tag{10}$$

式中: $\boldsymbol{W}_{ei} \in \mathbf{R}^{d'\times d}$为第$i$个专家的可训练权重矩阵, $\boldsymbol{b}_{ei}$为可训练的偏置向量.

对于现实生活中人或者物品的属性来说, 都可能由多个标签组成. 比如 Movielens-1M 中电影《Toy Story》, 所属类别是动画片、儿童片及喜剧, 人或物品的类别型属性可能有一个或多个标签. EDCN 中的 Regulation Module 可以看作单个 Experts, 因此只能捕捉特征中的单个语义, 忽略了其他大量关键的语义信息, 这是 EDCN 效果更差的原因. Broker 模块中有多个专家, 因此可以将特征的不同语义映射到多个子空间中, 每个专家对应一个子空间, 从而达到增强 Embedding 中特征表现力的效果. 每个门可以选取所有专家的一个子集, 根据各种建模方式为每个专家学习不同侧重点的权重. 当显式特征和隐式特征之间的关联较多时, Broker 模块会为某个专家分配较高的权重; 当关联较少时, Broker 模块会惩罚对应的专家, 尽量使用多个专家. 对于并行架构中存在的参数共享不足问题来说, 这是非常灵活的解决方案. Broker 模块参数数量是常数级别, 在整个模型中是可以忽略不计的, 因此在并行架构添加 Broker 模块后, 可以在不增加计算复杂度的情况下, 显著提高模型性能, 这是 Broker 模块的好处之一.

2.4 输出层

输出层将 2 个网络的输出拼接起来, 最终输出点击率预测结果:

$$\hat{y} = \sigma\,(\boldsymbol{W}[\boldsymbol{x}_l, \boldsymbol{h}_l]). \tag{11}$$

式中: 在 ME-PRAF 中$\boldsymbol{x}_l$为显式建模层的输出, $\boldsymbol{h}_l$为 MLP 层的输出, $\boldsymbol{W}$为可训练权重矩阵, σ为最终的激活函数. 该模型使用 sigmoid 函数作为激活函数, 即$\sigma(x) = 1/(1+\exp(-x))$.

损失函数使用 LogLoss 进行评估:

$$\text{LogLoss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \ln \hat{y}_i + (1-y_i)\ln(1-\hat{y}_i). \tag{12}$$

式中: y_i为真实标签, $\hat{y}_i$为模型的预测结果, N为输入数据的数量.

2.5 CowClip 训练加速

通常情况下, 在训练过程中, 增大训练 batch 虽然会缩短训练时间, 但是会带来模型性能的下降. 使用 CowClip[18] 模型来提高模型的训练速度, 由于推荐系统的大部分数据集中存在特征频次数量级相差较大的问题, 若增大训练 batch 但是不相应调整其他超参数, 比如学习速率和正则化系数, 则会导致模型训练造成偏差. 利用 CowClip 算法, 可以在不损耗模型性能的基础上增大训练的批次大小, 从而达到大幅度缩减训练时间的目的.

3 实验与分析

由于该算法框架是与模型无关的框架, 对比在各 SOTA 模型上使用 ME-PRAF 框架的效果.

3.1 数据集

使用以下 3 个数据集进行实验: Criteo 数据集、Avazu 数据集、MovieLens-1M 数据集. 具体数据如表 1 所示. 表中, M为数据集样本量, F为特征数量, C为词汇量.

表 1 3 个实验数据集的参数

Tab.1 Parameters of three datasets in experiment

数据集	$M/10^6$	F	$C/10^6$
Criteo	45	39	33
Avazu	40	23	9.4
Movielens-1M	0.74	7	0.013

Criteo 数据集是当前最流行的 CTR 基准数据集, 该数据集包含用户 7 天内点击广告的数据日志信息. 遵循先前 SOTA 工作中的处理操作, 将

前 6 天的用户数据作为训练集，将最后一天的用户数据平分作为验证集和测试集. 对于数值型数据，将所有数据放缩到 [0, 1.0].

Avazu 数据集是流行的 CTR 基准数据集，数据中包含了用户 11 d 内在移动端点击广告的信息，将 80% 的数据作为训练集，10% 的数据作为验证集，最终剩余 10% 的数据作为测试集.

MovieLens-1M 是十分知名流行的数据集，其中包含 3 个文件：评分数据、用户数据和电影数据. 将 3 个文件聚合成 1 个文件，其中每行数据对应的组织形式为：[用户属性，电影属性，评分]. 与先前的工作处理方式相同[6]，将评分等级为 1 或 2 设置为 0，将等级为 4 或 5 设置为 1，移除等级为 3 的数据. 将 80% 的数据作为训练集，10% 的数据作为验证集，最终剩余 10% 的数据作为测试集.

3.2 实现细节

使用以下 2 个指标对模型性能进行评估. 1）AUC (area under ROC curve)，用于衡量模型对随机选取的正标签样本较随机选取的负标签样本给出更高分值的概率，AUC 越高表示模型性能越好. 2）LogLoss，所有 CTR 模型都是为了最小化式 (12) 中的 LogLoss，LogLoss 越小表示模型性能越好. 对于 CTR 任务来说，若 AUC 增大 0.001 或 LogLoss 减小 0.001，则表示模型性能有了较大的提升[6,8-12].

将 ME-PRAF 框架应用到 DCN-v2 算法上，在 3 个数据集上的性能可以达到最优，以这个具有代表性的并行架构 CTR 模型作为演示，本文称为 ME-DCN(mixture of experts for DCN-v2) 算法. 若将 ME-DCN 中的 Broker 模块和 Fusion 模块删除，则会退化为 DCN-v2 算法.

ME-DCN 模型超参数的设置. 由于 Embedding Broker 可以训练学习表现力更强的 Embedding，只需要设置 Embedding 在所有数据集上的维度为 10. 优化器使用 Adam[19]，batch 大小默认设置为 8 192，MovieLens-1M 设置为 1 024，所有权重矩阵使用 He Normal[20] 进行初始化. 交叉层和 MLP 的层数都为 4，由于每一层交叉层和 MLP 需要进行 Fusion 操作，须保证 MLP 每一层输出维度与交叉层数据维度完全一致.

3.3 模型性能比较

参与对比的 SOTA 基准模型有 DeepFM、DCN、xDeepFM、AutoInt+、DCN-v2、CowClip 及 EDCN. 所有基准算法和本文算法都使用 TensorFlow[21] 进行实现. 如表 2 所示为 ME-DCN 与主流 SOTA 并行架构算法的对比，在 Criteo 数据集和 Avazu 数据集上 ME-DCN 算法优于其他算法，在 MovieLens-1M 数据集上 AUC 指标领先其他算法. 这说明 ME-DCN 较主流 SOTA 算法更能胜任 CTR 任务.

如表 3 所示为 ME-DCN 与主流 SOTA 并行架构模型参数量 N_p 的对比，表明 ME-DCN 算法的参数量较主流 SOTA 算法相对适中. 相比于参数较少的算法，参数较多的原因取决于该框架应用的原型算法，原型算法 DCN-v2 是在 Google 大规模商业数据集上取得优秀成绩的算法，与 DCN-v2 相比，ME-DCN 算法的参数量减少了 20%. 这表明 ME-DCN 的参数量处于可接受的范围之内，证明 Fusion 模块和 Broker 模块是轻量级的，可以部署到其他并行算法中，在工业级应用上是可行的.

分析 ME-DCN 的算法时间度可知，与 DCN-

表 2 ME-DCN 与其他 SOTA 模型在 3 个数据集上的性能比较

Tab.2 Performance comparisons between ME-DCN and other SOTA models in three datasets

模型	Criteo		Avazu		MovieLens-1M	
	AUC	LogLoss	AUC	LogLoss	AUC	LogLoss
DeepFM	0.8007	0.4508	0.7852	0.3780	0.8932	0.3202
DCN	0.8099	0.4419	0.7905	0.3744	0.8935	0.3197
xDeepFM	0.8052	0.4418	0.7894	0.3794	0.8923	0.3251
AutoInt+	0.8083	0.4434	0.7774	0.3811	0.8488	0.3753
DCN-v2	<u>0.8115</u>	<u>0.4406</u>	<u>0.7907</u>	0.3742	<u>0.8964</u>	**0.3160**
EDCN	0.8001	0.5415	0.7793	0.3803	0.8722	0.3469
CowClip	0.8097	0.4420	0.7906	<u>0.3740</u>	0.8961	0.3174
本文方法	**0.8122**	**0.4398**	**0.7928**	**0.3732**	**0.8970**	<u>0.3163</u>

表 3 ME-DCN 与其他模型参数量的对比 (Criteo)

Tab.3 Number of parameters comparison between ME-DCN and other models (Criteo)

模型	$N_p/10^6$
DeepFM	1.4
DCN	3.1
xDeepFM	4.2
AutoInt+	3.7
DCN-v2	7.2
EDCN	11
本文方法	5.7

v2 模型相比，增加时间复杂度的部分是 Broker 模块，专家部分和门控网络使用的是线性模型，因此时间复杂度为 $O(n)$. 并行网络中的每一层都对应一个 Broker 模块，ME-DCN 中的交叉层和 MLP 层数设置为 4，以累加的形式进行计算，因此时间复杂度为 $O(n)$.

3.4 ME-PRAF 框架的鲁棒性

为了证明 ME-PRAF 框架的鲁棒性，在其他 CTR 并行算法的基础上，融合 ME-PRAF 框架进行实验检验. 由于 DeepFM 显式建模部分只能有一层不能进行叠加，xDeepFM 在压缩感知层计算耗费十分昂贵，因此工业界很少使用. EDCN 模型中由于 regulation 模块的存在无法添加 Broker 模块，使用以下 3 种流行的 CTR 模型进行对比：DCN、AutoInt+、DCN-v2. 3 个数据集上的实验结果如表 4 所示.

表 4 SOTA 并行架构模型使用 ME-PRAF 后在 3 个数据集上的性能比较

Tab.4 Performance comparison of SOTA parallel architecture models after using ME-PRAF on three datasets

模型	Criteo		Avazu		MovieLens-1M	
	AUC	LogLoss	AUC	LogLoss	AUC	LogLoss
DCN	0.8099	0.4419	0.7905	0.3744	0.8935	0.3197
DCN_{ME}	0.8116	0.4403	0.7919	0.3731	0.8962	0.3174
AutoInt+	0.8083	0.4434	0.7774	0.3811	0.8488	0.3753
$\text{AutoInt+}_{\text{ME}}$	0.8104	0.4414	0.7899	0.3737	0.8928	0.3250
DCN-v2	0.8115	0.4406	0.7907	0.3742	0.8964	0.3160
$\text{DCN-v2}_{\text{ME}}$	0.8122	0.4398	0.7928	0.3732	0.8970	0.3163

从表 4 可知，ME-PRAF 算法框架对并行 CTR 算法模型具有很好的鲁棒性，在 AUC 和 LogLoss 2 个基准上都有有效的提升. 这表明 ME-PRAF 框架可以有效地提高并行 CTR 模型的性能，其中表 4 中的 $\text{DCN-v2}_{\text{ME}}$ 为 ME-DCN 模型. 在 Embedding 维度设置方面，DCN 和 DCN-v2 在 Criteo 数据集上的维度设置为 39，AutoInt+设置为 16，在本框架下的所有维度设置为 10. 这表明 ME-PRAF 框架不仅在并行算法上的性能提升较大，而且在 Embedding 维度较小的情况下有较好的性能，由此可以说明 ME-PRAF 框架下训练的 Embedding 表现力更强.

由于 Embedding 在模型中的参数量占据模型参数的很大一部分，利用本文算法可以大幅度减少模型的参数量，节约计算机内存及显存资源，在参与到模型计算时可以更快速地进行运算.

3.5 消融实验

为了进一步了解 ME-PRAF 算法框架中 Broker 模块的效果，对 Broker 模块进行消融实验. 由上文可知，Broker 模块分为 Embedding Broker 及 Feature Broker. 前者用于解决模型参数过度共享的问题，为并行架构训练学习具有可分辨性和个性化的输入；后者用于解决模型参数共享不足的问题，学习显式特征与隐式特征之间的关联. 对 Broker 模块进行消融实验的具体数据如表 5 所示.

表 5 中，w/o FB 表示将 ME-DCN 模型删除 Feature Broker 及 Fusion 模块后的实验结果，w/o EB 表示将 ME-DCN 模型删除 Embedding Broker 后

表 5 ME-DCN 模型上的 Broker 模块消融实验 (Criteo)

Tab.5 Ablation study of Broker modules in ME-DCN(Criteo)

模型	AUC	LogLoss
ME-DCN -w/o FB	0.811 7	0.440 3
ME-DCN -w/o EB	0.811 3	0.440 7
ME-DCN	0.812 2	0.439 8

的实验结果. 结果表明，删除其中一个都会导致模型性能下降，因此 Embedding Broker 和 Feature Broker 在算法模型中都十分重要而且缺一不可. 2 种 Broker 起到相辅相成的作用，为并行模型中存在的参数共享问题提供了解决方案，提高了模型性能.

3.6 Fusion 模块融合方式的对比

Fusion 模块的 3 种融合方式为拼接、按位加及 Hardmard 积. 这 3 种方式都不需要额外的参数，因此计算效率都很高. 为了探索不同 Fusion 方式对模型的影响，分别在 3 种方式下进行实验，实验结果如表 6 所示.

表 6 ME-DCN 模型上 Fusion 模块不同融合方式的性能对比 (Criteo)

Tab.6 Performance comparison of various fusion types in Fusion module in ME-DCN (Criteo)

模型	AUC	LogLoss
ME-DCN -w/ concat	0.812 2	0.439 8
ME-DCN -w/ add	0.810 5	0.441 8
ME-DCN -w/ Hardmard	0.810 7	0.441 6

由表 6 可知，拼接方式的效果比其他方式更好. 按位加方式的效果最差，由于相差较大的 2 对特征进行按位加融合后，最终向量会有较大概率出现结果相似的情况，选择拼接的融合方式更佳. 按照先前学者的研究经验，使用 Hardmard 积应取得较好的实验结果，但是此处的实验效果不理想，因此未来会进一步优化 Hardmard 积的融合方式.

3.7 模型参数调整

对于 ME-PRAF 算法框架来说，模型需要调参的地方如下.

1）在 Fusion 模块中需要调整对比的是特征的融合方式，这在 3.6 节中已进行讨论.

2）Broker 模块中参数的调整是对专家数量的调整. 为了研究专家数量对模型性能的影响，对 Broker 模块中专家数量分别为 2、3、4、5 的情况进行对比实验. 当专家数量小于 4 时，模型的性能会随着专家数量的增加而提高；当专家数量大于 4 时，性能开始变差；当专家数量为 4 时，模型性能最好. 可知，大部分数据集中特征不同语义平均数量为 4，当专家数量大于 4 时会捕捉无用冗余的语义特征，导致模型性能下降.

3.8 模型分析

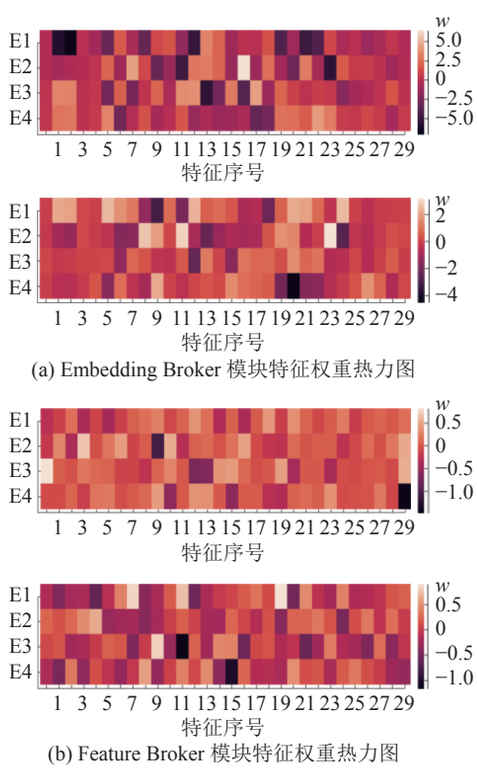
分析模型的关键在于模型是否能够学习到有意义的特征关联，在本框架中表现为以下 2 个方面.

1）Embedding Broker 是否能为不同类型的子网络学习到具有可分辨性和个性化的特征输入.

2）Feature Broker 是否能够学习到显式特征和隐式特征之间的关联信息.

现在大部分公司考虑用户隐私问题，将大部分数据集中的特征部分进行过脱敏处理，特征是加密后的数据. 采用 Avazu 数据集，分析 Broker 模块对特征的处理.

如图 4(a) 所示为 Embedding Broker 对输入特征的权重 w 热力图. 可知，Broker 模块不仅可以学习到输入特征不同语义下的信息，而且可以为不同并行架构子网学习到具有个性化的输入.



(a) Embedding Broker 模块特征权重热力图

(b) Feature Broker 模块特征权重热力图

图 4 Broker 模块特征权重的差异度分析

Fig.4 Analysis of diversity factor of feature weight of Broker module

如图 4(a) 所示，不同门控网络中热力图权重分布有着明显不同，在融合显式特征和隐式特征后，能够为下一层学习到具有可分辨性及个性化的特征信息，证明 Broker 模块的有效性.

若不使用 Broker 模块，则输入到显式特征模块和隐式特征模块的信息完全相同，因此图 4(a) 中 2 个热力图会完全一致. 2 个热力图分布的差异度越高，则表示输入到 2 个模块中的个性化程度越高. 图 4(b) 同理.

将 Fusion 模块和 Broker 模型两者配合，对不同子网络中的特征进行融合. 将融合后的信息分裂成最适合 2 个子网络的输入，显式特征与隐式

特征之间的信息得到有效交互, 提升了模型性能.

4 结 语

ME-PRAF 是轻量级且高性能的并行算法框架, 用于解决目前主流并行 CTR 推荐模型中普遍存在的参数共享问题. 对于并行架构中输入部分参数过度共享及子网络部分参数共享不足的问题, 可以泛化到众多并行 CTR 算法上, 有效提高模型的性能. 在数据集上的大量实验表明, ME-PRAF 框架能够有效地提高 SOTA 并行 CTR 算法模型的性能. 下一步将研究解决推荐系统中常见的冷启动问题以及如何在串行架构中融合显式特征和隐式特征.

参考文献 (References):

[1] KHAWAR F, HANG X, TANG R, et al. Autofeature: searching for feature interactions and their architectures for click-through rate prediction [C]// **Proceedings of the 29th ACM International Conference on Information and Knowledge Management**. [S. l.]: ACM, 2020: 625–634.

[2] HU D, WANG C, NIE F, et al. Dense multimodal fusion for hierarchically joint representation [C]// **2019 IEEE International Conference on Acoustics, Speech and Signal Processing**. Brighton: IEEE, 2019: 3941–3945.

[3] CHENG H T, KOC L, HARMSEN J, et al. Wide and deep learning for recommender systems [C]// **Proceedings of the 1st Workshop on Deep Learning for Recommender Systems**. Boston: ACM, 2016: 7–10.

[4] RENDLE S. Factorization machines [C]// **IEEE International Conference on Data Mining**. Sydney: IEEE, 2010: 995–1000.

[5] GUO H, TANG R, YE Y, et al. DeepFM: a factorization-machine based neural network for CTR prediction [C]// **Proceedings of the 26th International Joint Conference on Artificial Intelligence**. Melbourne: AAAI, 2017: 1725–1731.

[6] WANG R, SHIVANNA R, CHENG D, et al. DCN v2: improved deep & cross network and practical lessons for web-scale learning to rank systems [C]// **Proceedings of the Web Conference**. Ljubljana: ACM, 2021: 1785–1797.

[7] BEUTEL A, COVINGTON P, JAIN S, et al. Latent cross: making use of context in recurrent recommender systems [C]// **Proceedings of the 11th ACM International Conference on Web Search and Data Mining**. Marina Del Rey: ACM, 2018: 46–54.

[8] QU Y, FANG B, ZHANG W, et al. Product-based neural networks for user response prediction over multi-field categorical data [J]. **ACM Transactions on Information Systems**, 2018, 37(1): 1–35.

[9] ZHOU G, ZHU X, SONG C, et al. Deep interest network for click-through rate prediction [C]// **Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. London: ACM, 2018: 1059–1068.

[10] ZHOU G, MOU N, FAN Y, et al. Deep interest evolution network for click-through rate prediction [C]// **Proceedings of the AAAI Conference on Artificial Intelligence**. California: AAAI, 2019: 5941–5948.

[11] WANG R, FU B, FU G, et al. Deep & cross network for ad click predictions [M]// **Proceedings of the ADKDD'17**. Halifax: ACM, 2017: 1–7.

[12] SONG W, SHI C, XIAO Z, et al. Autoint: automatic feature interaction learning via self-attentive neural networks [C]// **Proceedings of the 28th ACM International Conference on Information and Knowledge Management**. Beijing: ACM, 2019: 1161–1170.

[13] LIAN J, ZHOU X, ZHANG F, et al. xdeepfm: combining explicit and implicit feature interactions for recommender systems [C]// **Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. London: ACM, 2018: 1754–1763.

[14] HUANG T, SHE Q, WANG Z, et al. GateNet: gating-enhanced deep network for click-through rate prediction [J]. **ArXiv**, 2020, 7(1): 1–7.

[15] CHEN B, WANG Y, LIU Z, et al. Enhancing explicit and implicit feature interactions via information sharing for parallel deep CTR models [C]// **Proceedings of the 30th ACM International Conference on Information and Knowledge Management**. Queensland: ACM, 2021: 3757–3766.

[16] MA J, ZHAO Z, YI X, et al. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts [C]// **Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. London: ACM, 2018: 1930–1939.

[17] HOLMES N P, SPENCE C. Multisensory integration: space, time and superadditivity [J]. **Current Biology**, 2005, 15(18): R762–R764.

[18] ZHENG Z, XU P, ZOU X, et al. CowClip: reducing CTR prediction model training time from 12 hours to 10 minutes on 1 GPU [J]. **ArXiv**, 2022, 4(1): 1–18.

[19] KINGMA D P, BA J. Adam: a method for stochastic optimization [J]. **ArXiv**, 2014, 12(1): 1–13.

[20] HE K, ZHANG X, REN S, et al. Delving deep into rectifiers: surpassing human-level performance on imagenet classification [C]// **Proceedings of the IEEE International Conference on Computer Vision**. Santiago: IEEE, 2015: 1026–1034.

[21] ABADI M, AGARWAL A, BARHAM P, et al. Tensorflow: large-scale machine learning on heterogeneous distributed systems [J]. **ArXiv**, 2016, 3(1): 1–19.