山东大学学报(理学版) 2024 年 7 月 第 59 卷 第 7 期
Journal of Shandong University(Natural Science), Vol.59, No.7, 2024

E-mail: xblxb@sdu.edu.cn
http://lxbwk.njournal.sdu.edu.cn
Tel: +86-531-88366917

文章编号: 1671-9352(2024)07-0044-09 DOI: 10.6040/j.issn.1671-9352.1.2023.042

基于矩阵乘积算符表示的序列化推荐模型

刘沛羽[1],姚博文[2],高泽峰[1,2*],赵鑫[1*]
(1.中国人民大学高瓴人工智能学院,北京 100872; 2.中国人民大学物理系,北京 100872)

摘要: 推荐系统中的序列化推荐任务面临着高度复杂和多样性大的挑战,基于序列化数据的商品表示学习中广泛采用预训练和微调的方法,现有方法通常忽略了在新领域中模型微调可能会遇到的欠拟合和过拟合问题。为了应对这一问题,构建一种基于矩阵乘积算符(matrix product operator, MPO)表示的神经网络结构,并实现2种灵活的微调策略。首先,通过仅更新部分参数的轻量化微调策略,有效地缓解微调过程中的过拟合问题;其次,通过增加可微调参数的过参数化微调策略,有力地应对微调中的欠拟合问题。经过实验验证,该方法在现有开源数据集上均实现显著的性能提升,充分展示在实现通用的物品表示问题上的有效性。

关键词: 推荐模型; 序列化数据; 矩阵乘积算符; 过拟合; 欠拟合
中图分类号: TP391 **文献标志码:** A
引用格式: 刘沛羽,姚博文,高泽峰,等.基于矩阵乘积算符表示的序列化推荐模型[J].山东大学学报(理学版),2024,59(7):44-52,104.

Matrix product operator based sequential recommendation model

LIU Peiyu[1], YAO Bowen[2], GAO Zefeng[1,2*], ZHAO Wayne Xin[1*]
(1. Gaoling School of Artificial Intelligence, Renmin University of China, Beijing 100872, China; 2. Department of Physics, Renmin University of China, Beijing 100872, China)

Abstract: The task of sequential recommendation confronts challenges characterized by high complexity and substantial diversity. The paradigm of pre-training and fine-tuning is extensively employed for learning item representations based on sequential data in recommendation scenarios. However, prevalent approaches tend to disregard the potential underfitting and overfitting issues that may arise during model fine-tuning in new domains. To address this concern, a novel neural network architecture grounded in the framework of matrix product operator (MPO) is introduced, and two versatile fine-tuning strategies are presented. Firstly, a lightweight fine-tuning approach that involves updating only a subset of parameters is proposed to effectively mitigate the problem of overfitting during the fine-tuning process. Secondly, an over-parameterization fine-tuning strategy is introduced by augmenting the number of trainable parameters, robustly addressing the issue of underfitting during fine-tuning. Through extensive experimentation on well-established open-source datasets, the efficacy of the proposed approach is demonstrated by achieving performance achievements. This serves as a compelling testament to the effectiveness of the proposed approach in addressing the challenge of general item representation in recommendation systems.

Key words: recommendation model; sequential data; matrix product operator; overfitting; underfitting

0 引言

序列化推荐任务是推荐系统领域中一个备受关注的议题,旨在通过用户与物品之间的交互序列对用户的偏好建模。目前,已经有多种方法用于解决这一问题,包括基于循环神经网络(recurrent neural network, RNN)[1]以及基于 Transformer 模型的方法[2]等。尽管这些方法在很多情况下都表现出色,但是在面对新的

收稿日期: 2023-11-24; 网络出版时间: 2024-05-29 17:00:10
网络出版地址: https://link.cnki.net/urlid/37.1389.N.20240528.0945.006
基金项目: 国家自然科学基金资助项目(62206299,62222215)
第一作者: 刘沛羽(1992—),男,博士研究生,研究方向为自然语言处理和模型压缩. E-mail: liupeiyustu@ruc.edu.cn
* 通信作者: 高泽峰(1995—),男,博士后,博士,研究方向为张量网络、预训练语言模型. E-mail: zfgao@ruc.edu.cn;
赵鑫(1985—),男,教授,博士,研究方向为自然语言处理以及推荐系统. E-mail: batmanfly@qq.com

推荐场景，尤其是当交互历史中缺乏足够序列信息时，如何有效地捕捉用户兴趣成为一个主要的挑战。

在自然语言处理领域解决这个问题的一种典型方法是采用预训练与微调的策略，首先在大规模的语料库上进行自监督学习预训练，然后在特定领域的少量数据上微调，即可在新任务上取得显著性能提升。受到这一思路的启发，最近的研究表明，通过在丰富的用户交互数据中预训练物品表示嵌入，在新场景中只需要进行领域特定的微调，就能够应对用户兴趣的冷启动问题[3]。微调所需的数据量相对较少，因此能够有效缓解新场景下数据不足的问题，然而，类似于自然语言处理领域的研究指出，预训练模型在微调时仍然可能遭遇效果下降的风险，主要原因包括潜在的过拟合和欠拟合问题[4]。当微调数据与预训练数据差异较大且微调数据量较少时，容易导致欠拟合; 相反，过多的微调数据则可能导致过拟合。基于预训练与微调范式的推荐模型的研究中忽略了这个问题，因此，一个直观的想法是，通过灵活控制微调过程中的参数量，采用少量参数的微调来实现模型的正则化，以应对过拟合问题; 或者通过更多参数的微调辅助模型的优化，从而应对欠拟合问题，这也是本文研究的出发点。

受矩阵乘积算符(matrix product operator，MPO) 方法的启发，本文首次在序列化推荐问题中引入基于矩阵乘积算符的神经网络表示结构，并设计一种灵活的微调策略，以应对微调中可能出现的过拟合和欠拟合问题。具体来说，本文利用权重矩阵的矩阵乘积算符表示，创造性地构建一种新的神经网络权重表示结构，将权重矩阵分解为多个局部张量的乘积形式，其具备 2 个关键特点: 1) 大部分局部张量的参数量小于原始矩阵，而局部张量的参数量总和略大于原始矩阵的参数量; 2) 通过更新部分局部张量，可以有效地更新整个权重表示。基于这些特点，本文提出 2 种灵活的微调策略: 一种是针对部分局部张量进行微调，以实现少参数微调; 另一种是针对所有局部张量进行微调，以实现过参数微调。值得注意的是，经过微调后的权重表示可以通过矩阵乘积算符的合并操作恢复为原始矩阵形式，确保模型在下游任务的推理过程中不会引入额外的参数或计算开销。通过一系列下游任务的实验，证明本文的方法可以有效地缓解模型的过拟合和欠拟合问题。

综上，本文针对序列化推荐问题，通过引入基于矩阵乘积算符的神经网络表示结构，提出灵活的微调策略，以应对微调过程中可能存在的过拟合和欠拟合问题。该方法在各类下游任务中取得了显著的性能提升，为基于预训练的推荐模型的进一步研究和应用提供新的思路和方法。

1　相关工作

1.1　序列化推荐系统

用户表示问题在推荐系统领域、序列化推荐任务中具有重要意义。用户在一段时间内的交互历史形成了一个包含多个商品的序列化数据，是反映用户偏好的关键信息，基于序列化数据的推荐系统在实际应用中尤为关键和典型。随着深度学习的蓬勃发展，许多研究开始采用深度神经网络来直接对商品序列化信息进行建模。例如，文献[5]首次使用了门控循环单元(gated recurrent unit，GRU) 来捕捉序列化商品信息，此后出现了许多基于 RNN 模型[1]、Transformer 模型[3]、多层感知机[6]以及图神经网络[7]等的方法。另外，还有一些研究关注于丰富商品的表示信息，而不仅仅局限于商品 ID 的序列，然而，这些方法往往在特定领域限制较大，难以在新领域中表现出良好的迁移性能。

1.2　预训练和微调方法的应用

预训练和微调作为一种在自然语言处理领域广泛应用的学习范式，近年来也在其他领域取得了显著成功。这一范式首先在大量的自监督信号中对模型进行预训练，然后通过在特定的下游任务上微调，即可在仅有少量标注数据的情况下实现显著性能提升。近期，一些研究将预训练-微调范式应用于序列化推荐建模问题中，旨在获得通用的用户或物品表示。例如，文献[8]重新学习通用的用户表示以应对面向用户的下游任务，而文献[8]从物品序列中学习通用的物品表示。这些方法通常采用全参数微调策略，然而在自然语言处理领域的研究已经指出，全参数微调在某些情况下可能不适用。文献[4]发现在微调过程中模型容易过拟合，导致在微调数据有限的情况下性能下降，而轻量化微调策略可以起到正则化的作用。类似地，文献[9]表明在某些任务下，通过过参数化微调可以进一步提升模型性能。这些研究在基于预训练的推荐系统中的应用，特别是在序列化推荐领域中的微调问题，尚未得到充分探讨。

1.3　基于张量分解的神经网络模型

张量分解方法是一种将矩阵分解为多个张量的技术。在深度学习领域，MPO 最早是由 Gao 等[10]引入，

作为神经网络中线性层更高效表示的方法,主要被应用于压缩网络结构,包括对深度神经网络[11]、卷积神经网络[12]以及长短期记忆网络(long short term memory networks, LSTM[13])的压缩。此外,基于 MPO 分解的特性,还有研究通过微调部分分解的张量来更新整个网络,实现轻量化微调[14],以及与混合专家结构(mixture-of-expert, MoE)结合,通过扩展部分张量作为额外的专家模块,构建参数高效的大型 MoE 网络结构[15]。虽然在其他领域有关于 MPO 的研究已经涌现,但在基于预训练的推荐系统领域尚未有类似的探索。本文的创新之处在于首次将 MPO 分解引入序列化推荐问题中,以填补这一领域的研究空白。

2 预备知识

为了方便起见,首先定义本章需要用到的数学符号。本文一般将标量表示为小写字母(例如,a),向量表示为粗体小写字母(例如,$\boldsymbol{v}$),矩阵表示为大写粗体字母(例如,$\boldsymbol{M}$),高阶的张量表示为大写粗体的字母(例如,$\boldsymbol{T}$)。矩阵乘积算符方法是一种源自量子物理领域解决多体问题的张量分解方法,在深度神经网络中常被用于模型压缩[9-11,14-17]。本章主要介绍矩阵乘积算符方法的分解过程和矩阵重建过程。

2.1 矩阵乘积算符的分解和重建过程

MPO 分解可以表示为一个过程:输入矩阵 $\boldsymbol{M}\in\mathbf{R}^{I\times O}$,通过 MPO 分解可以得到 n 个张量乘积的形式

$$\mathrm{MPO}(\boldsymbol{M})=\prod_{k=1}^{n}\boldsymbol{T}_k[d_{k-1},i_k,o_k,d_k], \tag{1}$$

其中:$\boldsymbol{T}_k[d_{k-1},i_k,o_k,d_k]$ 是一个四阶张量;$\prod_{k=1}^{n} i_k=I$,$\prod_{k=1}^{n} o_k=O$,其中 n 表示分解的张量的个数,n 越大则会引入更多的参数量。文献[14]中讨论了关于 n 的选择方法,并分析参数 n 的影响,对于 $n=2,3,5,7$ 没有特别大的影响,一般选择 $n=5$。符号 d_k 表示张量之间的连接键,计算方法为

$$d_k=\min\left(\prod_{m=1}^{k} i_m\times o_m,\ \prod_{m=k+1}^{n} i_m\times o_m\right)。 \tag{2}$$

2.2 矩阵乘积算符方法的特点

矩阵乘积算符方法主要有 2 个特点。首先,通过分解得到的局部张量的参数分布主要集中在中间位置的张量。特别地,如果 n 是奇数,则主要集中在中间的 1 个张量;如果 n 是偶数,则集中在中间的 2 个张量,这 2 个张量参数量相等。其中第 k 个张量的参数量 count_k 以及总参数量 C 可以通过下列方法计算:

$$\mathrm{count}_k=d_{k-1}i_k o_k d_k, \tag{3}$$

$$C=\sum_{k=1}^{n}\mathrm{count}_k。 \tag{4}$$

另外,更新矩阵乘积算符方法得到的所有张量中任意一个张量,都可以实现对重建矩阵的更新。换句话说,通过矩阵乘积算符表示的权重,可以优化分解后的部分张量来实现对权重整体的学习。由于部分张量参数量远远小于总参数量,可以极大降低可训练的参数量,模型的轻量化微调成为可能。

3 矩阵乘积算符方法

3.1 基于矩阵乘积算符方法的神经网络层设计

在本节中,基于矩阵乘积算符方法设计了一种全新的神经网络层设计 LinearMPO,旨在构建可高效微调的模型结构。通常,神经网络的线性层结构由权重矩阵 $\boldsymbol{W}$ 和偏置项 bias 组成,在输入 $\boldsymbol{x}$ 的情况下,其工作原理可以用以下方程表示:

$$\mathrm{hidden\_states}=\boldsymbol{W}\boldsymbol{x}+\mathrm{bias}。 \tag{5}$$

在传统的线性层结构中,矩阵 $\boldsymbol{W}$ 包含了整个网络层的关键信息,很难进行部分参数微调或引入额外结构进行过参数微调。受到矩阵乘积算符方法的启发,对线性层的矩阵结构进行了创新性的改进,将其进行张量分解,得到 n 个不同尺寸的张量结构,如在 2.1 节中所述。在这个新的结构下,基于矩阵乘积算符表示的神经网络层的工作原理可以表示为

$$\mathrm{hidden\_states}=\mathrm{MPO}(\{\boldsymbol{T}_1,\boldsymbol{T}_2,\cdots,\boldsymbol{T}_n\})\boldsymbol{x}+\mathrm{bias} \tag{6}$$

其中$\{T_1,T_2,\cdots,T_n\}$代表矩阵 W 经过 MPO 分解得到的 n 个张量,MPO$(\cdot)$ 表示将多个张量重构为矩阵的过程。为了应用这一新的结构,将模型中所有的线性层替换为基于矩阵乘积算符表示的线性层 LinearMPO,从而得到了一种经过改进的模型结构,能够更加灵活地进行微调操作,以应对不同情况下可能出现的欠拟合和过拟合问题。

3.1.1　训练过程

为了更加详细地阐述训练过程,本文考虑一个包含 L 层全连接层的简化模型。在这个模型中,每一层都包含一个权重矩阵 W_l,这一设定同样适用于其他包含全连接层的结构。

模型的典型训练过程涉及 3 个关键阶段: 参数初始化、前向传播和反向传播。首先,在参数初始化阶段,采用标准的线性层初始化方法,例如 Xavier 初始化,来初始化权重矩阵 W_l。本文的创新之处在于,将初始化后的 W_l 进行 MPO 分解,得到一组张量$\{T_1^{(l)},T_2^{(l)},T_3^{(l)},T_4^{(l)},T_5^{(l)}\}$,作为张量集合$\{T_i\}_{i=1}^n$的初始状态。在前向传播阶段,首先将张量集合$\{T_i\}_{i=1}^n$合并为矩阵 W_l,并将其与输入进行计算。值得注意的是,合并过程遵循 2.1节所述的方式,不会引入额外的误差。在反向传播阶段,使用常见的优化方法(例如,AdamW)对张量集合进行梯度计算和迭代更新。通过这一训练过程,模型可以逐步优化,使得基于矩阵乘积算符方法的神经网络层能够更好地适应于特定任务,同时也能够灵活地应对微调过程中可能出现的过拟合和欠拟合问题。

3.1.2　推理过程

在模型的推理过程中,由于 LinearMPO 相对于标准的全连接层包含了额外的矩阵重建过程,直接进行模型推理会导致额外的计算负担。为了有效减轻这部分附加计算开销,可以采取一种优化方法,即将所有 LinearMPO 层中的张量重构为矩阵,并基于这些重构矩阵以及偏置项来初始化新的全连接层,以替代原有的 LinearMPO 结构。通过这种优化方法,经过微调后的 LinearMPO 结构与原始的全连接结构模型在推理模式上保持完全一致,同时不会引入任何额外的计算和存储开销。这种操作可以被视为一种有效的推理优化,在维持模型性能的同时减少了重建过程所产生的计算开销。在实际应用中,这一推理优化方法可以显著降低模型推理的计算负担,从而提升模型的实际应用性能。通过这样的策略,能够在推理过程中充分发挥 LinearMPO 结构的优势,而无需为了额外的计算而牺牲性能。

算法 1 总结了模型训练和推理相关的所有过程。在 LinearMPO 的基础上,模型的权重矩阵可以被表示为多个张量的乘积,为本文后续实现灵活的微调策略打好了基础。

算法 1　MPO 线性层的训练过程

输入: $W^{(l)}$ 为第 l 层的全连接层矩阵; η 为学习率; CE 为损失函数; L 为模型层数。

初始化过程

1: for $0<l\leqslant L$ do

2:　　Xavier 初始化 $W(l)$

3:　　$\{T_1^{(l)},T_2^{(l)},T_3^{(l)},T_4^{(l)},T_5^{(l)}\}\leftarrow$MPO$(W^{(l)})$

4: end for

训练过程

5: while 未收敛 do

(前向传播)

6:　　$W^{(l)}=T_1^{(l)}T_2^{(l)}T_3^{(l)}T_4^{(l)}T_5^{(l)}$

7:　　$L=\mathrm{CE}(x,y;W^{(l)})$

(反向传播)

8:　　$t\leftarrow t+1$

9:　　$g_t\leftarrow\dfrac{\partial L}{\partial(T_t^{(l)})}$

10: $T_t^{(l)}\leftarrow T_t^{(l)}-\eta\cdot g_t$

11: end while

推理过程

(初始化全连接层代替 LinearMPO)

12: $W^{(l)}\leftarrow T_1^{(l)}T_2^{(l)}T_3^{(l)}T_4^{(l)}T_5^{(l)}$

3.2 基于多种微调策略的物品表示学习

典型的预训练-微调方法通常可以支持在与预训练完全不同的领域中进行迁移学习,以将预训练的知识应用于新领域。现有研究表明,直接在新领域微调预训练模型可能不一定是最佳策略。在某些领域中,采用轻量化微调[14]或过参数化微调[9]等方法,可以有效提升模型性能,因此,迫切需要一种方法,既能够支持轻量化微调,又能够支持过参数化微调。接下来将详细介绍如何基于 LinearMPO 实现灵活的轻量化微调和过参数化微调策略。

3.2.1 轻量化微调方法

轻量化微调是一种只训练部分参数的策略,其目的是在新领域实现预训练模型的迁移,并在尽可能保持新领域模型性能与全参数微调相近的同时,减少训练开销,然而,现有方法通过引入额外的可训练参数模块[18],增加了额外的计算成本。受到文献[14]的启发,本文将权重 W_l 进行 MPO 分解为 $\{T_i^{(l)}\}_{i=1}^{n}$,分解后的张量矩阵具有特殊的参数分布特点,其中大部分参数主要集中在 1 个或 2 个中间位置的张量上,中间位置的张量在 n 为奇数时为 1 个张量,在 n 为偶数时为 2 个张量。其他位置的张量只包含少量的参数。此外,微调这些张量中的任意一个都可以实现模型的轻量化微调。在微调过程中,通过仅训练其他位置的张量,可以有效减少可训练参数的数量,抑制在特定领域下的过拟合问题(如图 1 中轻量化微调方法所示)。

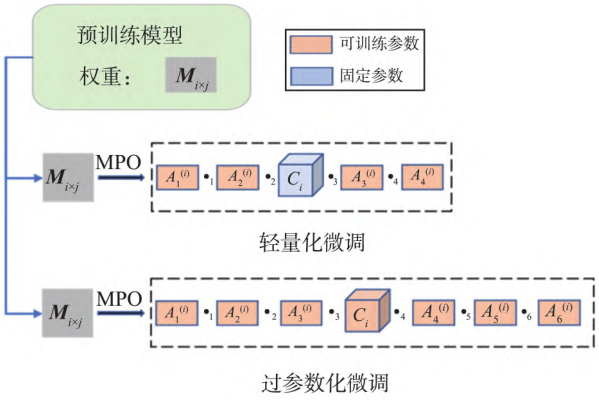

图 1 轻量化微调和过参数化微调

Fig.1 Lightweight & over-parameter fine-tuning

3.2.2 过参数化微调方法

过参数化微调意味着将模型的现有权重过度参数化,以获得一个包含更多可训练参数的新权重,可以辅助模型的优化,从而实现更好的性能。为了实现过参数化微调,主要的挑战是在引入额外参数的过程中,尽量少的修改模型结构以及降低参数在模型推理中额外的计算开销。首先,本文借助权重的矩阵乘积算符表示,可以通过调整 MPO 分解的长度,灵活控制模型可训练的参数量(如图 1 中过参数化微调所示)。具体来说,当输入权重 W_l 被表示为 $\{T_i^{(l)}\}_{i=1}^{n}$,通过在分解后的张量中插入 n 个额外的形状为 $T^{d_{k-1}\times 1\times 1\times d_k}$ 的张量,模型总参数量扩大 ρ 倍,其中

$$\rho=\frac{t\times d_k\times d_{k-1}+\sum_{k=1}^{m} i_k o_k d_k d_{k-1}}{\prod_{i=1}^{m} i_k o_k}。\tag{7}$$

另外,过参数化微调完成后,张量 $\{T_i^{(l)}\}_{i=1}^{n}$ 可以通过合并重建为权重矩阵 W_l 的形式,这样一来,模型的推理过程不会引入任何额外的存储和计算开销。这一策略在一些情况下可能非常有效,尤其是在面临欠拟合问题时。通过引入更多的可训练参数,模型可以更充分地利用训练数据,从而取得更好的优化效果。

通过以上 2 种微调策略,在不同情况下都能够实现模型的有效优化。这种灵活的微调方法可以根据特定领域的需求,灵活选择合适的微调策略,从而在实际应用中取得更好的性能。

4 实验

4.1 实验数据

为了评估该文方法的有效性,文献[3]从亚马逊评论数据集(Amazon Review Dataset[19])中选择 5 个数据集进行测试,包括 Scientific、Pantry、Instruments、Arts 和 Office 作为该文的评测数据集。表 1 列出了数据集的详细统计信息。

表 1　数据集评测信息

Table 1　Details of datasets for evaluation

数据集	用户个数	产品个数	交互个数	Avg. n	Avg. c
Scientific	8 842	4 385	52 427	7.04	182.87
Pantry	13 101	4 898	126 962	9.69	83.17
Instruments	24 962	9 964	208 926	8.37	165.18
Arts	45 486	21 019	395 150	8.69	155.57
Office	87 436	25 986	684 837	7.84	193.22

4.2　对比模型

本文考虑 2 种典型的序列化推荐场景，分别是将物品序号作为特征进行学习，以及不考虑物品序号的情况。在每一种情况下，本文分别实现了 3 个方法变种：

1) MPORecLight：轻量化微调方法。将模型所有的全连接层权重均表示为 $n=5$ 的矩阵乘积算符形式，并且微调除 $n=2$ 以外的所有张量。

2) MPORec：过参数化微调方法。将模型所有的全连接层权重均表示为 $n=5$ 的矩阵乘积算符形式，并且微调所有张量，这时可训练参数量略微比原始的全连接层多一些。

3) MPORec(Expand_t)：过参数化微调方法。首先将模型所有的全连接层权重均表示为 $n=4+t$ 的矩阵乘积算符形式，其中位置 i，$i\in\{1<i<4+t\}$ 的张量的连接键均为 1。

同时本文也考虑了几种典型的基线模型。具体来说，基线模型可以被分为两大类：一种是基于预训练技术的序列化推荐模型，是本文方法主要对比的基线方法；另一种是直接基于已有预训练模型的方法。

对比的方法主要包括以下几种。

1) 基于预训练技术的序列化推荐模型。这里主要考虑 UniSRec [3] 模型，这是一种基于混合专家结构的序列化推荐模型，其中 UniSRec_F 表示固定编码器结构的轻量化微调结果。S3Rec [20] 模型，是一种基于最大化互信息的自监督学习框架，结合 4 个预训练目标，以增强属性、项目和序列的表征。BERT4Rec [21] 采用预训练好的 BERT 用于建模用户序列表示。

2) 其他序列化推荐方法。CCDR [22] 提出领域内和领域间的基于对比学习的目标函数，用于解决跨域推荐中的匹配问题。

4.3　实验结果

对比不同的基线模型。主要对比在新领域的迁移中，基于矩阵乘积算符方法的序列化推荐模型和基于预训练技术的方法效果的差异，结果展示在表 2 中。首先对比本文方法和基线模型，发现在测试的 5 个数据集上，本文的方法基本都达到最佳的的效果，而且本文方法在不同数据集上的提升幅度也不同。值得注意的是在 Pantry 数据集上，本文方法的提升效果显著，在 hit@ 50 以及 ndcg@ 50 上取得了相比最佳基线方法 9.11%和 7.97%的效果提升。而在 Office 以及 Arts 上只有略微提升，原因可能是在这 2 个数据集中与预训练阶段的数据存在差异导致的，将会在未来的研究中继续跟进这个问题。最后，相比于其他的序列化推荐方法 CCDR，从表格中的结果看出本文提出的方法以及其他基于预训练的方法均具有明显的优势。该结果表明，在新领域的迁移任务中，基于预训练技术的方法可能依然是最有竞争力的。

表 2　对比不同基线模型的评测结果

Table 2　Comparison with other baseline models

Dataset	Metric	S3Rec	BERT4Rec	CCDR	UniSRec	MPORec	MPORecLight
Scientific	hit@ 10	0.052 5	0.048 8	0.069 5	0.109 5	0.110 3	**0.111 6**
	hit@ 50	0.141 8	0.118 5	0.164 7	0.211 9	0.205 6	**0.222 2**
	ndcg@ 10	0.027 5	0.024 3	0.034 0	0.059 8	0.059 6	**0.059 9**
	ndcg@ 50	0.046 8	0.039 3	0.054 6	0.083 5	0.083 5	**0.083 7**
Pantry	hit@ 10	0.044 4	0.030 8	0.048 0	0.062 7	**0.066 4**	0.060 5
	hit@ 50	0.131 5	0.103 0	0.126 2	0.171 1	**0.179 0**	0.170 1
	ndcg@ 10	0.021 4	0.015 2	0.020 3	0.030 8	**0.032 4**	0.030 5
	ndcg@ 50	0.040 0	0.030 5	0.038 5	0.054 2	**0.056 8**	0.054 1

续表

Dataset	Metric	S3Rec	BERT4Rec	CCDR	UniSRec	MPORec	MPORecLight
Instruments	hit@10	0.105 6	0.081 3	0.084 8	0.112 4	**0.116 4**	0.107 8
	hit@50	0.192 7	0.145 4	0.175 3	0.208 6	**0.220 0**	0.196 8
	ndcg@10	0.071 3	0.062 0	0.045 1	0.065 8	**0.067 6**	0.062 9
	ndcg@50	0.090 1	0.075 6	0.064 7	0.086 7	**0.090 1**	0.082 3
Arts	hit@10	0.110 3	0.072 2	0.067 1	0.101 8	**0.101 9**	0.093 4
	hit@50	0.188 8	0.136 7	0.147 8	0.199 3	**0.199 8**	0.186 1
	ndcg@10	0.060 1	0.047 9	0.034 8	0.057 3	**0.057 5**	0.051 9
	ndcg@50	0.079 3	0.061 9	0.052 3	0.078 4	**0.078 9**	0.072 0
Office	hit@10	0.103 0	0.082 5	0.054 9	0.094 7	**0.095 8**	0.082 8
	hit@50	0.161 3	0.122 7	0.109 5	0.164 7	**0.168 4**	0.144 2
	ndcg@10	0.065 3	0.063 4	0.029 0	0.056 0	**0.056 1**	0.049 6
	ndcg@50	0.078 0	0.072 1	0.040 9	0.071 3	**0.071 4**	0.062 9

对比不同的微调策略。表3展示了本文实现的过参数微调结果MPORec和轻量化微调结果MPORecLight,发现在Scientific数据集中MPORecLight取得了最好的结果,而其他数据集中MPORec效果最好,表明下游任务面临不同的微调挑战,而本文的方法由于支持多种灵活的微调策略可以有效应对各种情况。将本文结果和已经进行轻量化微调的UniSRec_F进行对比,发现在Scientific、Instruments、Arts和Office数据集上,MPORec均取得了最好的结果,而在Pantry数据集上,通过进一步扩展6个张量的MPORec+ex6方法取得了最好的结果。上述结果说明,尽管UniSRec_F通过固定编码器结构已经取得了相比UniSRec更好的结果,但因过少的可训练参数反而面临严重的欠拟合的问题,尤其是在Pantry数据集上,这个问题相对来说更加明显一些。而本文方法通过进一步扩充模型参数可以直接缓解这一问题。同时对比结果也说明本文实现的方法有很强的兼容性,由于不需要修改结构,因此可以灵活适配各种方法。

表3 对比不同微调策略的结果

Table 3 Comparison of different fine-tuning strategies

Dataset	Metric	UniSRec_F	MPORec	MPORecLight	MPORec +ex2	MPORec +ex4	MPORec +ex6	Improvement/%
Scientific	hit@10	0.118 8	**0.125 2**	0.112 1	0.124 3	0.122 7	0.122 0	5.39
	hit@50	0.239 4	**0.240 0**	0.221 2	0.236 0	0.237 6	0.237 9	0.25
	ndcg@10	0.064 1	**0.065 4**	0.060 9	0.065 3	0.065 0	0.065 2	2.03
	ndcg@50	0.090 3	**0.090 2**	0.084 8	0.089 7	0.090 0	0.090 4	0.11
Pantry	hit@10	0.063 6	0.067 3	0.061 9	0.066 6	0.067 9	**0.069 2**	8.81
	hit@50	0.165 8	0.180 1	0.169 8	0.179 4	0.178 6	**0.180 9**	9.11
	ndcg@10	0.030 6	0.032 0	0.029 7	0.031 7	0.032 4	**0.032 7**	6.86
	ndcg@50	0.052 7	0.056 4	0.053 1	0.056 1	0.056 2	**0.056 9**	7.97
Instruments	hit@10	0.118 9	**0.121 1**	0.109 2	0.116 1	0.118 8	0.120 0	1.85
	hit@50	0.225 5	**0.225 6**	0.203 8	0.220 1	0.224 2	0.226 0	0.22
	ndcg@10	0.068 0	**0.069 0**	0.064 1	0.067 3	0.068 0	0.068 8	1.47
	ndcg@50	0.091 2	**0.091 7**	0.084 6	0.089 8	0.090 9	0.091 8	0.66
Arts	hit@10	0.106 6	**0.108 3**	0.092 2	0.107 4	0.105 0	0.103 8	1.59
	hit@50	0.204 9	**0.212 2**	0.183 3	0.209 7	0.206 4	0.204 3	3.56
	ndcg@10	0.058 6	**0.059 4**	0.050 2	0.059 2	0.057 6	0.057 1	1.37
	ndcg@50	0.079 9	**0.082 1**	0.070 1	0.081 5	0.079 7	0.079 0	2.75
Office	hit@10	0.101 3	**0.102 9**	0.088 0	0.100 9	0.101 0	0.100 7	1.58
	hit@50	0.170 2	**0.171 0**	0.150 6	0.169 1	0.168 8	0.168 2	0.47
	ndcg@10	0.061 9	**0.063 2**	0.054 0	0.062 5	0.062 1	0.617 0	2.10
	ndcg@50	0.076 9	**0.078 1**	0.0676	0.077 3	0.076 9	0.076 5	1.56

4.4　讨论与分析

矩阵乘积算符参数分析。本节用于评估矩阵乘积算符表示长度对结果的影响。通过在 MPO 分解的过程中,设置分解长度分别是 $n \in \{3,5,7,9\}$,用来表示全连接层的权重矩阵。然后,基于这些不同的表示结构,在 Scientific 数据集中测试基础的 MPORec 方法微调,来对比不同的分解长度对结果的影响,结果如表 4 所示。从表 4 中看出,MPORec 针对不同的分解长度差异并不大,对比不同的分解长度,取 $n=5$ 可以获得最佳的微调效果。

表 4　分解长度参数 n 影响分析

Table 4　Impact of different decomposition length n

长度 n	3	5	7	9
MPORec	0.124 3	0.125 2	0.120 2	0.118 2

超参数敏感性分析。主要用于对比和分析超参数敏感性的影响。本文提出的方法是基于矩阵的乘积算符表示,由于引入了更加复杂的结构和额外的参数量,因此相比较原始的全连接层,本文的方法训练会更加稳定,并且对超参数的敏感性更低。为了验证这一点,选择不同的学习率来测试 MPORec 的微调方法,因为学习率在深度学习方法中通常被认为是最敏感的参数。选择学习率的范围为{1e-4,2e-4,4e-4,6e-4,8e-4},结果见表 5。从结果中看出,虽然不同的学习率带来的结果有差异,但是基本都维持在 0.12 左右。

表 5　学习率影响分析

Table 5　Impact of different learning rate

学习率	1e-4	2e-4	4e-4	6e-4	8e-4
MPORec	0.119 7	0.119 8	0.120 2	0.122 8	0.123 0

微调参数比较。本节对不同微调策略需要训练的参数量、显存占用和微调用时进行了全面的比较。本文提出的基于矩阵乘积算符的权重表示可以灵活地支持多种微调策略,包括使用少量参数更新全部权重的轻量化微调以及提升参数量的过参数化微调。这里统计了不同策略下模型的总参数量和可训练参数量,结果见表 6。其中 UniSRecF 表示 UniSRec 的实现变种,即固定编码器结构不更新。由表 6 可以看出,本文实现的方法均可以有效地调整模型的参数量,由于这种调节并不需要修改模型结构,因此为下游不同任务的轻量化微调或者过参数化微调提供支持。从训练时间来看,一方面,轻量化微调的 MPORecLight 训练参数更少,速度更快,更适合在资源有限的条件下使用;另一方面,过参数化微调方法(+Expand)需要相对较多的资源,在要求下游任务效果更好的场景下更加适合。

表 6　训练效率对比分析

Table 6　Analysis of the training efficiency

模型	总参数量/M	训练参数量/M	显存/GB	训练时间/s
UniSRec	6.3	6.3		
UniSRecF	6.3	1.9	7.74	1 498
MPORecLight	6.5	0.3	7.76	703
MPORec	6.5	2.1	7.78	980
+Expand2	6.6	2.2	7.81	1 448
+Expand4	6.7	2.3	7.84	1567
+Expand6	6.8	2.4	7.88	3 509

5　结论

本文针对推荐系统中应用预训练模型,并将预训练模型通过微调的方法应用于物品表示学习中的微调低效的问题,提出了基于矩阵乘积算符分解的物品嵌入表示学习策略,有效提高了推荐系统中的预训练模型在微调过程中遇到的欠拟合和过拟合问题。基于 MPO 分解构建张量表示的神经网络,并实现了 2 种灵活高效的微调策略:一方面,通过固定中心张量的方法可以有效地缓解微调过程中的过拟合问题;另一方面,通过增加辅助张量的方法可以有效增加可微调参数,从而实现在微调过程中的过参数化策略,这个方法可以有效地应对模型在微调中出现的欠拟合问题。经过详细的实验验证,本文所提出的方法在现有的开源数据集

上均实现了显著的性能提升,充分展示了实现通用的物品表示问题上的有效性。在未来的工作中,可以将基于 MPO 分解的微调方法用更大规模的模型,并进一步开发适配不同场景的轻量化微调策略。

参考文献:

[1] LI Jing, REN Pengjie, CHEN Zhumin, et al. Neural attentive session-based recommendation [EB/OL]. (2017-11-13) [2023-08-09]. http://arxiv.org/abs/1711.04725.

[2] HOU Yupeng, HU Binbin, ZHANG Zhiqiang, et al. CORE: simple and effective session-based recommendation within consistent representation space [C] // SIGIR'22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: Association for Computing Machinery, 2022: 1796-1801.

[3] HOU Y, MU S, ZHAO W X, et al. Towards universal sequence representation learning for recommender systems [EB/OL]. (2022-06-13) [2023-04-13]. http://arxiv.org/abs/2206.05941.

[4] XU Ruixin, LUO Fuli, ZHANG Zhiyuan, et al. Raise a child in large language model: towards effective and generalizable fine-tuning [C] //Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Punta Cana: Association for Computational Linguistics, 2021: 9514-9528.

[5] HIDASI B, KARATZOGLOU A, BALTRUNAS L, et al. Session-based recommendations with recurrent neural networks [C/OL] //4th International Conference on Learning Representations (ICLR 2016). 2016: 1-10. http://arxiv.org/pdf/1511.06939.

[6] ZHOU K H, YU H, ZHAO W X, et al. Filter-enhanced MLP is all you need for sequential recommendation [C] // WWW'22: The ACM Web Conference 2022. Lyon: ACM, 2022: 2388-2399.

[7] CHANG Jianxin, GAO Chen, ZHENG Yu, et al. Sequential recommendation with graph neural networks [EB/OL]. (2023-07-26) [2023-08-09]. http://arxiv.org/abs/2106.14226.

[8] YUAN F, HE X, KARATZOGLOU A, et al. Parameter-efficient transfer from sequential behaviors for user modeling and recommendation [C] //Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2020. New York: ACM, 2020: 1469-1478.

[9] GAO Zefeng, ZHOU Kun, LIU Peiyu, et al. Small pre-trained language models can be fine-tuned as large models via over-parameterization [C] //Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics ACL 2023. Toronto: Association for Computational Linguistics, 2023: 3819-3834.

[10] GAO Zefeng, CHENG Song, HE Rongqiang, et al. Compressing deep neural networks by matrix product operators [J]. Physical Review Research, 2020, 2(2): 023300.

[11] GAO Zefeng, SUN Xingwei, GAO Lan, et al. Compressing LSTM networks by matrix product operators [EB/OL]. (2022-03-31) [2023-05-06]. https://arxiv.org/abs/2012.11943.

[12] NOVIKOV A, PODOPRIKHIN D, OSOKIN A, et al. Tensorizing neural networks [EB/OL]. (2015-09-22) [2023-05-06]. https://arxiv.org/abs/1509.06569.

[13] GARIPOV T, PODOPRIKHIN D, NOVIKOV A, et al. Ultimate tensorization: compressing convolutional and FC layers alike [EB/OL]. (2016-11-10) [2023-05-06]. https://arxiv.org/abs/1611.03214.

[14] LIU P, GAO Z F, ZHAO W X, et al. Enabling lightweight fine-tuning for pre-trained language model compression based on matrix product operators [C] //Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. Stroudsburg: Association for lomputational Linguistics, 2021: 5388-5398.

[15] GAO Z F, LIU P, ZHAO W X, et al. Parameter-efficient mixture-of-experts architecture for pre-trained language models [C] //Proceedings of the 29th International Conference on Computational Linguistics. Gyeongju: International Committee on Computational Linguistics, 2022: 3263-3273.

[16] LIU Peiyu, GAO Zefeng, CHEN Yushuo, et al. Scaling pre-trained language models to deeper via parameter-efficient architecture [EB/OL]. (2023-04-10) [2023-05-06]. http://arxiv.org/abs/2303.16753.

[17] SUN Xingwei, GAO Zefeng, LU Zhengyi, et al. A model compression method with matrix product operators for speech enhancement [J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2020, 28: 2837-2847.

[18] EDWARD J H, SHEN Y, WALLIS P, et al. LoRA: low-rank adaptation of large language models [EB/OL]. (2021-10-16) [2022-06-16]. http://arxiv.org/abs/2106.09685.

(下转第104页)

[23] BRUCH S, ZOGHI M, BENDERSKY M, et al. Revisiting approximate metric optimization in the age of deep neural networks [C] //Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. Paris: ACM, 2019: 124-1244.

[24] PANG Liang, XU Jun, AI Qingyao, et al. Setrank: learning a permutation-invariant ranking model for information retrieval [C] //SIGIR'20: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2020: 499-508.

[25] QIN Tao, LIU Tieyan. Introducing LETOR 4.0 datasets [EB/OL]. (2013-01-09) [2023-10-18]. http://arxiv.org/abs/1306.2597.

[26] DATO D, LUCCHESE C, NARDINI F M, et al. Fast ranking with additive ensembles of oblivious and non-oblivious regression trees [J]. ACM Transactions on Information Systems (TOIS), 2016, 35(2): 1-31.

[27] WANG X H, LI C, GOLBANDI N, et al. The LambdaLoss framework for ranking metric optimization [C] //Proceedings of the 27th ACM International Conference on Information and Knowledge Management. Torino: ACM, 2018: 1313-1322.

(编辑:李艺)

(上接第52页)

[19] NI J, LI J, MCAULEY J J. Justifying recommendations using distantly-labeled reviews and fine-grained aspects [C] //Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. Hong Kong: Association for Computational Linguistics, 2019: 188-197.

[20] ZHOU K, WANG H, ZHAO W X, et al. S3Rec: self-supervised learning for sequential recommendation with mutual information maximization [C/OL] //Proceedings of the 29th ACM International Conference on Information & Knowledge Management. [2023-08-08]. https://doi.org/10.1145/3340531.3411954.

[21] SUN Fei, LIU Jun, WU Jian, et al. BERT4Rec: sequential recommendation with bidirectional encoder representations from transformer [C] // Proceedings of the 28th ACM International Conference on Information and Knowledge Management. Beijing: ACM, 2019: 1441-1450.

[22] WEN Keyu, TAN Zhenshan, CHENG Qingrong, et al. Contrastive cross-modal knowledge sharing pre-training for vision-language representation learning and retrieval [EB/OL]. (2022-07-08) [2023-10-18]. http://arxiv.org/abs/2207.00733.

(编辑:李艺)