文章编号：1003-0077（2017）00-0000-00

**基于矩阵乘积算符表示的序列化推荐模型**

刘沛羽11 姚博文22 高泽峰3\*1,2 赵鑫4\*1

（1.中国人民大学 高瓴人工智能学院，北京市 100872；2.中国人民大学 物理系，北京市 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。预训练和微调的方法在基于序列化数据的商品表示学习中被广泛采用，然而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，本文构建了一种基于矩阵乘积算符表示的神经网络结构，并实现了两种灵活的微调策略。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。经过广泛的实验验证，该方法在现有开源数据集上均实现了显著的性能提升，充分展示了在实现通用的物品表示问题上的有效性。

**关键词：微调；过拟合；欠拟合**

**中图分类号：**TP391  **文献标识码：**A

#### Matrix Product Operator based Sequential Recommendation Model

Peiyu Liu1 , Bowen Yao2 , Ze-Feng Gao\*1,2 and Wayne Xin Zhao\*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, we introduce a novel neural network architecture grounded in the framework of matrix product operator (MPO). Additionally, the study presents two versatile fine-tuning strategies. 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: fine-tuning; overfitting; underfitting**

**0 前言**

序列化推荐任务是推荐系统领域一个备受关注的议题，它旨在通过用户与物品之间的交互序列来建模用户的偏好。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（RNN）的方法[1]以及基于Transformer模型的方法[2]。尽管这些方法在很多情况下都表现出色，然而在面对新的推荐场景，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。

为了解决这个问题，一种典型的方法是采用预训练与微调的策略。这个策略在自然语言处理领域被广泛应用，模型首先在大规模的语料库上进行自监督学习预训练，然后通过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[3]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。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[4]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。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。但是这一问题在基于预训练与微调范式的推荐模型的研究中被忽略了。因此，一个直观的想法是，通过灵活地控制微调过程中的参数量，可以通过少量参数的微调来实现对模型的正则化，以应对过拟合问题，或者通过更多参数的微调来辅助模型的优化，从而应对欠拟合问题，这也是本文的研究出发点。

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，并设计了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，创造性地构建了一种新的神经网络权重表示结构。这种表示方式将权重矩阵分解为多个局部张量的乘积形式，其具备两个关键特点：首先，大部分局部张量的参数量小于原始矩阵，而所有局部张量的参数量总和略大于原始矩阵的参数量；其次，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，本文提出了两种灵活的微调策略，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，这样可以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，本文的方法证明可以有效地缓解模型的过拟合和欠拟合问题。

总之，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，并提出了灵活的微调策略，以应对微调过程中可能存在的过拟合和欠拟合问题。实验结果表明，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了新的思路和方法。

1. **相关工作**
	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，这些数据是反映用户偏好的关键信息。基于这种序列化数据的推荐系统在实际应用中显得尤为关键和典型。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，方法[5]首次使用了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. **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，方法[8]重新学习通用的用户表示以应对面向用户的下游任务，而方法[3]从物品序列中学习通用的物品表示。然而，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。方法[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，方法[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。然而，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. **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Ze-Feng Gao等人[10]引入，作为神经网络中线性层更高效表示的方法。它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MoE结构结合，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]。然而，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。

1. **预备知识**

为了方便起见，这里首先定义本章需要用到的数学符号。这里本文一般将标量表示为小写字母（如，），向量表示为粗体小写字母（如，），矩阵表示为大写粗体字母（如，），高阶的张量表示为加粗的欧拉体字母（如，）。矩阵乘积算符方法（Matrix Product Operator，MPO）是一种源自量子物理领域解决多体问题的张量分解方法，在深度神经网络中被常被用于模型压缩[9–11,14–17]以及微调[9,14]。接下来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。

* 1. **矩阵乘积算符的分解和重建过程**

MPO分解可以表示为这样一个过程：输入矩阵，通过MPO分解可以得到个张量乘积的形式：

其中是一个四阶张量，下标中的字符与矩阵形状的关系为,。其中表示分解的张量的个数，越大则会引入更多的参数量。关于的选择在方法[14]中都有过讨论，没有特别大的影响，一般选择，我们在第5.1.4章分析了这个参数的影响。符号表示张量之间的连接键，它的计算方法为：

* 1. **矩阵乘积算符方法的特点**

矩阵乘积算符方法主要有两个特点。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。特别的，如果是奇数，则主要集中在中间的一个张量，如果是偶数，则集中在中间的两个张量，这两个张量参数量相等。其中第个张量的参数量以及总参数量可以通过下列方法计算：

另外，矩阵乘积算符方法得到的所有张量中更新任意一个张量，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，可以通过优化分解后的部分张量来实现对权重整体的学习，又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，使得模型的轻量化微调成为了可能。

1. **方法**

本章旨在介绍基于矩阵乘积算符方法的商品表示学习。主要分为两个关键部分：基于矩阵乘积算符技术的神经网络层设计（见第4.1节），旨在构建可高效微调的模型结构；以及多种灵活的微调策略（见第4.2节），用于支持轻量化微调和过参数化微调的策略。随后，我们将详细阐述这两部分的具体内容。

* 1. **基于矩阵乘积算符技术的神经网络层设计**

在本节中，我们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，名为LinearMPO。通常，神经网络的线性层结构由权重矩阵和偏置项组成，在输入的情况下，其工作原理可以表示为以下方程：

然而，传统的线性层结构中，矩阵包含了整个网络层的关键信息，这使得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，我们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，得到个不同尺寸的张量结构，如在第3.1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以下方程：

其中代表矩阵经过MPO分解得到的个张量，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，我们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。这种设计使得我们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。

* + 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，请注意，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三个关键阶段：参数初始化、前向传播和反向传播。首先，在参数初始化阶段，我们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。然而，本文的创新之处在于，我们随后将初始化后的进行MPO分解，得到一组张量，作为张量集合的初始状态。在前向传播阶段，我们首先将张量集合合并为矩阵，并将其与输入进行计算。值得注意的是，合并过程遵循如前文第3.1所述的方式，不会引入额外的误差。在反向传播阶段，我们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。

* + 1. **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，我们可以采取一种优化方法，即将所有LinearMPO层中的张量重新重构为矩阵，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，因为它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，从而提升了模型的实际应用性能。通过这样的策略，我们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，这样为本文后续实现灵活的微调策略打好了基础。本文将在第4.1节继续介绍微调策略。

|  |
| --- |
| 算法1 MPO 线性层的训练过程 |
| 输入: :第*l*层的全连接层矩阵;*η*:学习率;*CE*:损失函数;*L*:模型层数。**初始化过程**1: for do2: Xavier 初始化 W(l)3: 4: end for**训练过程**5: while 未收敛 do (前向传播)6: 7: *=CE(x,y;*W*(l))* (反向传播)8:  *t*9: 10:11: end while **推理过程**(初始化全连接层代替LinearMPO)12:  |

* 1. **基于多种微调策略的物品表示学习**

这一部分将介绍基于LinearMPO的商品表示学习的多种微调策略。典型的预训练-微调方法通常可以支持在与预训练领域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。然而，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，我们迫切需要一种方法，既能够支持轻量化微调，又能够支持过参数化微调。接下来，本章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。



图 1 轻量化微调和过参数化微调

* + 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，现有方法要么通过引入额外的可训练参数模块[18]来修改模型结构，不适用于复杂网络，要么通过辅助指标计算来选择需要微调的参数[4]，增加了额外的计算成本。受到[14]的启发，本文将权重进行MPO分解为。这些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一个或两个中间位置的张量上。这些中间位置的张量在为奇数时为一个张量，在为偶数时为两个张量。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）。

* + 1. **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，这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额外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，当输入权重被表示为，通过在分解后的张量中插入个额外的形状为的张量，模型总参数量扩大倍，其中：

另外，过参数化微调完成后，张量可以通过合并重建为权重矩阵的形式，这样以来，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。

通过以上两种微调策略，我们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。

1. **实验**
	1. **实验设置**
		1. **实验数据**

为了评估该文方法的有效性，该文参考论文[3]从亚马逊评论数据集（Amazon Review Dataset）[19]中选择了5个数据集进行测试，包括“Pantry”、 “Scientific”、“Instruments”、“Arts”、 “Crafts” 和“Office Products”作为该文的评测数据集。在下面的表格中列出来了数据集的详细统计信息。

表格 1数据集评测信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据集 | 用户数 | 产品数 | 交互数 | Avg. n | Avg. c |
| Scientific | 8842 | 4385 | 52427 | 7.04 | 182.87 |
| Pantry | 13101 | 4898 | 126962 | 9.69 | 83.17 |
| Instruments | 24962 | 9964 | 208926 | 8.37 | 165.18 |
| Arts | 45486 | 21019 | 395150 | 8.69 | 155.57 |
| Office | 87436 | 25986 | 684837 | 7.84 | 193.22 |

* + 1. **对比模型**

本文考虑两种典型的序列化推荐场景，分别是将物品序号也作为特征进行学习，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的矩阵乘积算符形式，并且微调除以外的所有张量。
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的矩阵乘积算符形式，并且微调所有张量，这个时候可训练参数量略微比原始的全连接层多一些。
3. MPORec（）：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的矩阵乘积算符形式，其中位置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，另外一种是直接基于已有预训练模型的方法。

该文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。

表格 2对比不同基线模型的评测结果

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | Metric | S3rec | BERT4rec | CCDR | UniSRec | MPORec | MPORecLight |
| Scientific | hit@10 | 0.0525 | 0.0488 | 0.0695 | 0.1095 | 0.1103 | **0.1116** |
| hit@50 | 0.1418 | 0.1185 | 0.1647 | 0.2119 | 0.2056 | **0.2222** |
| ndcg@10 | 0.0275 | 0.0243 | 0.034 | 0.0598 | 0.0596 | **0.0599** |
| ndcg@50 | 0.0468 | 0.0393 | 0.0546 | 0.0835 | 0.0835 | **0.0837** |
| Pantry | hit@10 | 0.0444 | 0.0308 | 0.048 | 0.0627 | **0.0664** | 0.605 |
| hit@50 | 0.1315 | 0.103 | 0.1262 | 0.1711 | **0.179** | 0.1701 |
| ndcg@10 | 0.0214 | 0.0152 | 0.0203 | 0.0308 | **0.0324** | 0.0305 |
| ndcg@50 | 0.04 | 0.0305 | 0.0385 | 0.0542 | **0.0568** | 0.0541 |
| Instruments | hit@10 | 0.1056 | 0.0813 | 0.0848 | 0.1124 | **0.1164** | 0.1078 |
| hit@50 | 0.1927 | 0.1454 | 0.1753 | 0.2086 | **0.2200** | 0.1968 |
| ndcg@10 | 0.0713 | 0.062 | 0.0451 | 0.0658 | **0.0676** | 0.0629 |
| ndcg@50 | 0.0901 | 0.0756 | 0.0647 | 0.0867 | **0.0901** | 0.0823 |
| Arts | hit@10 | 0.1103 | 0.0722 | 0.0671 | 0.1018 | **0.1019** | 0.0934 |
| hit@50 | 0.1888 | 0.1367 | 0.1478 | 0.1993 | **0.1998** | 0.1861 |
| ndcg@10 | 0.0601 | 0.0479 | 0.0348 | 0.0573 | **0.0575** | 0.0519 |
| ndcg@50 | 0.0793 | 0.0619 | 0.0523 | 0.0784 | **0.0789** | 0.072 |
| Office | hit@10 | 0.103 | 0.0825 | 0.0549 | 0.0947 | **0.0958** | 0.0828 |
| hit@50 | 0.1613 | 0.1227 | 0.1095 | 0.1647 | **0.1684** | 0.1442 |
| ndcg@10 | 0.0653 | 0.0634 | 0.029 | 0.0560 | **0.0561** | 0.0496 |
| ndcg@50 | 0.078 | 0.0721 | 0.0409 | 0.0713 | **0.0714** | 0.0629 |

这里主要考虑UniSRec[3]模型，这是一种基于混合专家结构（Mixture-of-experts，MoE）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。模型，是一种基于最大化互信息（MIM）的自监督学习框架，结合了四个预训练目标，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]提出了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。

* + 1. **实验结果**

**对比不同的基线模型**。本节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，结果展示在表格 2中。首先对比本文方法和基线模型，我们发现在测试的五个数据集上，本文的方法基本都达到了最佳的的效果。而且我们还发现，不同数据集上本文方法的提升幅度也不同，值得注意的是在Pantry数据集上，本文方法的效果特别显著，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，这个结果的原因可能是在这两个数据集中与预训练阶段的数据差异导致的，这个问题我们将会在未来的研究中继续跟进。最后，相比较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取得了明显的优势，这个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。

**对比不同的微调策略。**本节对比了不同微调策略的结果。在表格 3中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。这个结果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。在表格 3中本文和已经进行轻量化微调的“UniSRec\_F”进行对比。

我们发现，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。这个结果说明，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，但由于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。同时这个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，可以灵活适配各种方法。

表格 3 对比不同微调策略的结果

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | Metric | UniSRec\_F | MPORec | MPORecLight | MPORec+ex2 | MPORec+ex4 | MPORec+ex6 | Improv. |
| Scientific | hit@10 | 0.1188 | **0.1252** | 0.1121 | 0.1243 | 0.1227 | 0.122 | 5.39% |
| hit@50 | 0.2394 | **0.24** | 0.2212 | 0.236 | 0.2376 | 0.2379 | 0.25% |
| ndcg@10 | 0.0641 | **0.0654** | 0.0609 | 0.0653 | 0.0650 | 0.0652 | 2.03% |
| ndcg@50 | 0.0903 | **0.0902** | 0.0848 | 0.0897 | 0.0900 | 0.0904 | 0.11% |
| Pantry | hit@10 | 0.0636 | 0.0673 | 0.0619 | 0.0666 | 0.0679 | **0.0692** | 8.81% |
| hit@50 | 0.1658 | 0.1801 | 0.1698 | 0.1794 | 0.1786 | **0.1809** | 9.11% |
| ndcg@10 | 0.0306 | 0.032 | 0.0297 | 0.0317 | 0.0324 | **0.0327** | 6.86% |
| ndcg@50 | 0.0527 | 0.0564 | 0.0531 | 0.0561 | 0.0562 | **0.0569** | 7.97% |
| Instruments | hit@10 | 0.1189 | **0.1211** | 0.1092 | 0.1161 | 0.1188 | 0.12 | 1.85% |
| hit@50 | 0.2255 | **0.2256** | 0.2038 | 0.2201 | 0.2242 | 0.226 | 0.22% |
| ndcg@10 | 0.068 | **0.069** | 0.0641 | 0.0673 | 0.068 | 0.0688 | 1.47% |
| ndcg@50 | 0.0912 | **0.0917** | 0.0846 | 0.0898 | 0.0909 | 0.0918 | 0.66% |
| Arts | hit@10 | 0.1066 | **0.1083** | 0.0922 | 0.1074 | 0.1050 | 0.1038 | 1.59% |
| hit@50 | 0.2049 | **0.2122** | 0.1833 | 0.2097 | 0.2064 | 0.2043 | 3.56% |
| ndcg@10 | 0.0586 | **0.0594** | 0.0502 | 0.0592 | 0.0576 | 0.0571 | 1.37% |
| ndcg@50 | 0.0799 | **0.0821** | 0.0701 | 0.0815 | 0.0797 | 0.079 | 2.75% |
| Office | hit@10 | 0.1013 | **0.1029** | 0.0880 | 0.1009 | 0.1010 | 0.1007 | 1.58% |
| hit@50 | 0.1702 | **0.1710** | 0.1506 | 0.1691 | 0.1688 | 0.1682 | 0.47% |
| ndcg@10 | 0.0619 | **0.0632** | 0.0540 | 0.0625 | 0.0621 | 0.6170 | 2.10% |
| ndcg@50 | 0.0769 | **0.0781** | 0.0676 | 0.0773 | 0.0769 | 0.0765 | 1.56% |

* + 1. **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。我们通过在MPO分解的过程中，设置不同的分解长度分别是，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，我们在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，结果展示在表格 4中。从表格中看出，MPORec放在针对不同的分解长度差异并不大，同时我们注意到，对比不同的分解长度，只需要取可以获得最佳的微调效果。

表格 4矩阵乘积算符参数影响分析

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 长度n | 3 | 5 | 7 | 9 |
| MPORec | 0.1243 | 0.1252 | 0.1202 | 0.1182 |

**超参数敏感性分析**。这里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，本文的方法相比较原始的全连接层，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，这里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一般被认为是最敏感的参数。这里选择学习率的范围为，结果展示在表格 5中。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，并且最佳的学习率1e-3对应的结果相比其他结果的平均差异基本维持在3%以内。

表格 5超参数影响分析

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学习率 | 1e-4 | 2e-4 | 4e-4 | 6e-4 | 8e-4 |
| MPORec | 0.1197 | 0.1198 | 0.1202 | 0.1228 | 0.1230 |

**微调参数比较。**本章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成绩算符的权重表示可以灵活的支持多种微调策略，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，结果展示在表格 6中。其中表示UniSRec的实现变种，即固定编码器结构不更新。我们可以看出本文实现的方法均可以有效的调整模型的参数量，由于这种调节并不需要修改模型结构，为不同下游任务的轻量化微调或者过参数化微调提供了支持。从训练时间来看，轻量化微调的MPORecLight训练参数更少，速度更快，更适合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高的场景下更加适合。

表格 6训练效率对比分析

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 | 1498 |
| MPORecLight | 6.5 | 0.3 | 7.76 | 703 |
| MPORec | 6.5 | 2.1 | 7.78 | 980 |
| +Expand2 | 6.6 | 2.2 | 7.81 | 1448 |
| +Expand4 | 6.7 | 2.3 | 7.84 | 1567 |
| +Expand6 | 6.8 | 2.4 | 7.88 | 3509 |

1. **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。本文基于MPO分解构建张量表示的神经网络，并实现了两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的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，这个方法可以有效的应对模型在微调中出现的欠拟合问题。经过扎实的实验验证，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分解的微调方法应用于不同的推荐系统场景中，并进一步开发针对不同的场景适配的轻量化微调策略。

**致谢**

这项工作部分得到了中国国家自然科学基金的支持，项目编号为62206299和62222215，以及北京市杰出青年科学家计划的支持，项目编号为BJJWZYJH012019100020098，以及CCF-Zhipu人工智能大型模型基金的支持。高泽峰和赵鑫是通讯作者。

 **参考文献**

[1] Li J, Ren P, Chen Z, 等. Neural attentive session-based recommendation: arXiv:1711.04725[Z/OL]. arXiv, 2017(2017–11–13)[2023–08–09]. http://arxiv.org/abs/1711.04725. DOI:10.48550/arXiv.1711.04725.

[2] Hou Y, Hu B, Zhang Z, 等. CORE: simple and effective session-based recommendation within consistent representation space[C]. Amigó E, Castells P, Gonzalo J, 等, 编//SIGIR ’22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022. .

[3] Hou Y, Mu S, Zhao W X, 等. Towards universal sequence representation learning for recommender systems: arXiv:2206.05941[Z/OL]. arXiv, 2022(2022–06–13)[2023–04–13]. http://arxiv.org/abs/2206.05941. DOI:10.48550/arXiv.2206.05941.

[4] Runxin Xu, Luo F, Zhang Z, et al. Raise a child in large language model: towards effective and generalizable fine-tuning[C/OL]//Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021: 9514–9528[2022–06–16]. https://aclanthology.org/2021.emnlp-main.749. DOI:10.18653/v1/2021.emnlp-main.749.

[5] Hidasi B, Karatzoglou A, Baltrunas L, 等. Session-based recommendations with recurrent neural networks[C]//4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings. .

[6] Zhou K, Yu H, Zhao W X, 等. Filter-enhanced mlp is all you need for sequential recommendation[C]. Laforest F, Troncy R, Simperl E, 等, 编//WWW ’22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022. .

[7] Chang J, Gao C, Zheng Y, 等. Sequential recommendation with graph neural networks: arXiv:2106.14226[Z/OL]. arXiv, 2023(2023–07–26)[2023–08–09]. http://arxiv.org/abs/2106.14226. DOI:10.48550/arXiv.2106.14226.

[8] Yuan F, He X, Karatzoglou A, 等. 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, Virtual Event, China, July 25-30, 2020. .

[9] Gao Z-F, Zhou K, Liu P, 等. 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 (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023. .

[10] Gao Z-F, Cheng S, He R-Q, 等. Compressing deep neural networks by matrix product operators[J]. Phys. Rev. Res., 2020, 2(2): 023300. DOI:10.1103/PhysRevResearch.2.023300.

[11] Gao Z-F, Sun X, Gao L, 等. Compressing lstm networks by matrix product operators[J]. arXiv preprint arXiv:2012.11943, 2020.

[12] Novikov A, Podoprikhin D, Osokin A, 等. Tensorizing neural networks[J]. arXiv preprint arXiv:1509.06569, 2015.

[13] Garipov T, Podoprikhin D, Novikov A, 等. Ultimate tensorization: compressing convolutional and fc layers alike[J]. arXiv preprint arXiv:1611.03214, 2016.

[14] Liu P, Gao Z-F, Zhao W X, 等. 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 (Volume 1: Long Papers). .

[15] Gao Z-F, Liu P, Zhao W X, 等. Parameter-efficient mixture-of-experts architecture for pre-trained language models[C/OL]//Proceedings of the 29th International Conference on Computational Linguistics. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, 2022: 3263–3273[2023–02–02]. https://aclanthology.org/2022.coling-1.288.

[16] Liu P, Gao Z-F, Chen Y, 等. Scaling pre-trained language models to deeper via parameter-efficient architecture: arXiv:2303.16753[Z/OL]. arXiv, 2023(2023–04–10)[2023–05–06]. http://arxiv.org/abs/2303.16753. DOI:10.48550/arXiv.2303.16753.

[17] Sun X, Gao Z-F, Lu Z-Y, 等. 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. DOI:10.1109/TASLP.2020.3030495.

[18] Edward J. Hu, Shen Y, Wallis P, et al. LoRA: low-rank adaptation of large language models: arXiv:2106.09685[Z/OL]. arXiv, 2021(2021–10–16)[2022–06–16]. http://arxiv.org/abs/2106.09685.

[19] Ni J, Li J, McAuley J J. Justifying recommendations using distantly-labeled reviews and fine-grained aspects[C]. Inui K, Jiang J, Ng V, 等, 编//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019. .

[20] Zhou K, Wang H, Zhao W X, 等. S^3-rec: 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]. http://arxiv.org/abs/2008.07873. DOI:10.1145/3340531.3411954.

[21] Sun F, Liu J, Wu J, 等. BERT4Rec: sequential recommendation with bidirectional encoder representations from transformer[C/OL]. Zhu W, Tao D, Cheng X, 等, 编//Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019. ACM, 2019: 1441–1450. https://doi.org/10.1145/3357384.3357895. DOI:10.1145/3357384.3357895.

[22] Keyu Wen, Tan Z, Cheng Q, 等. Contrastive cross-modal knowledge sharing pre-training for vision-language representation learning and retrieval: arXiv:2207.00733[Z/OL]. arXiv, 2022(2022–07–08)[2022–07–12]. http://arxiv.org/abs/2207.00733.

刘沛羽（1992年—），博士研究生，主要研究领域为自然语言处理和模型压缩。

E-mail：liupeiyustu@ruc.edu.cn



姚博文（1999年—），硕士研究生，主要研究领域为预训练模型的高效微调。

E-mail：bolganis@ruc.edu.cn



高泽峰（1995年—），通信作者，博士，博士后，主要研究领域为张量网络，预训练语言模型。

E-mail：zfgao@ruc.edu.cn



赵鑫（1985年—），通信作者，博士，教授，主要研究领域为自然语言处理以及推荐系统。

E-mail：batmanfly@qq.com