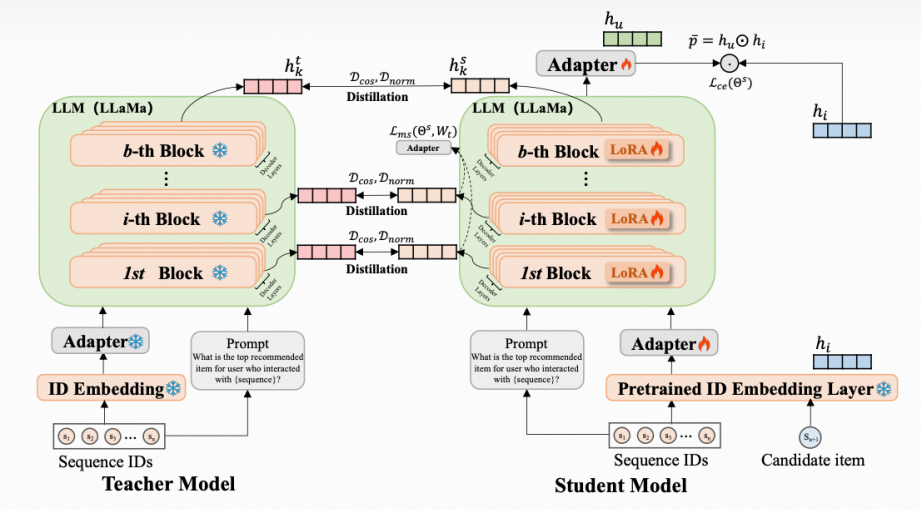
知乎



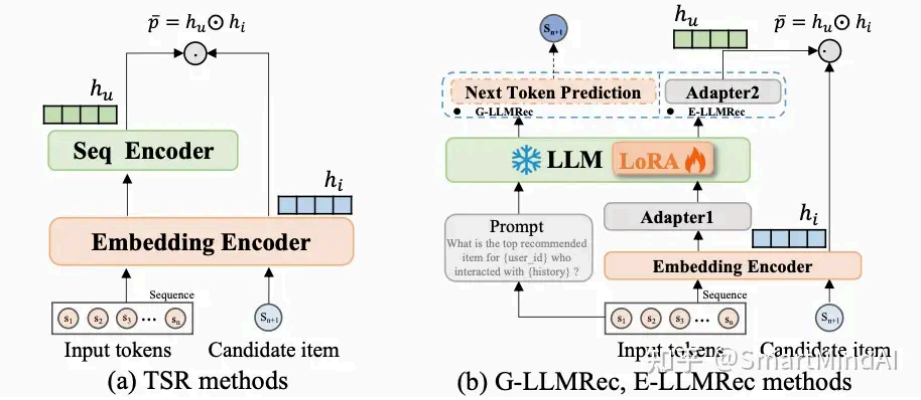
2024蚂蚁：SLMRec将LLM蒸馏，让大模型解决序列推荐的工业级挑战

SmartMindAI
专注搜索、广告、推荐、大模型和人工智能最新技术，欢迎关注我
已关注

20 人赞同了该文章

Introduction

学习时间相关兴趣信息是序列推荐模型的核心。传统的序列推荐（TSR）方法注重构建复杂的序列编码器，从LSTM和GRU架构发展到自注意力层和Transformer模型。然而，尽管这些方法在TSR领域取得了最先进的性能，但受限于模型通常只有不到0.1亿个参数，其性能提升已接近瓶颈。 最近，随着大型语言模型（LLMs）通过增加训练数据量或模型规模的进步，它们在多个方面取得了显著进展。基于先前研究的规模定律，LLMs展现出更强的表达能力，从而在基准测试中达到卓越表现。当前基于LLMs的推荐架构的发展引起了一种趋势的担忧。 当前的LLM基推荐系统可以分为两类：



1) 生成型（G-LLMRec）通过预测序列中的下一个token。 2) 嵌入型（E-LLMRec）以用户隐状态为用户表示，用适配器计算偏爱。 LLMs的应用极大地推动了序列推荐任务的发展，相对于TSR模型，在特定基准上的性能提升近20%。这激发了我们研究，探索如何利用LLMs更有效地服务于推荐任务。我们专注于探究在序列推荐（SR）中，是否能通过合理利用LLMs的规模，而非单纯追求大模型。NLP领域中LLM的冗余性启示我们，尽管LLMs在NLP任务上表现出色，但在SR任务中过大的模型并非必需。我们关注的是找到合适大小的LLM，既能保证性能，又能减少资源消耗。现有的LLM驱动的SR模型如P5、CoLLM等虽然性能提升显著，但参数量膨胀问题严重，增加了70倍，这在面对海量日志和实时更新的环境时显得不切实际。因此，我们的研究旨在优化LLM在SR中的应用，通过减少不必要的模型大小，如利用LLMs的规模定律，以实现既能提升性能又节省资源的目标。

我们的主要工作包括： 1. 深入探讨LLMs规模需求：我们研究了大型语言模型在SR任务中的必要性，挑战传统观念，质疑过大的模型是否真的必要。

知乎

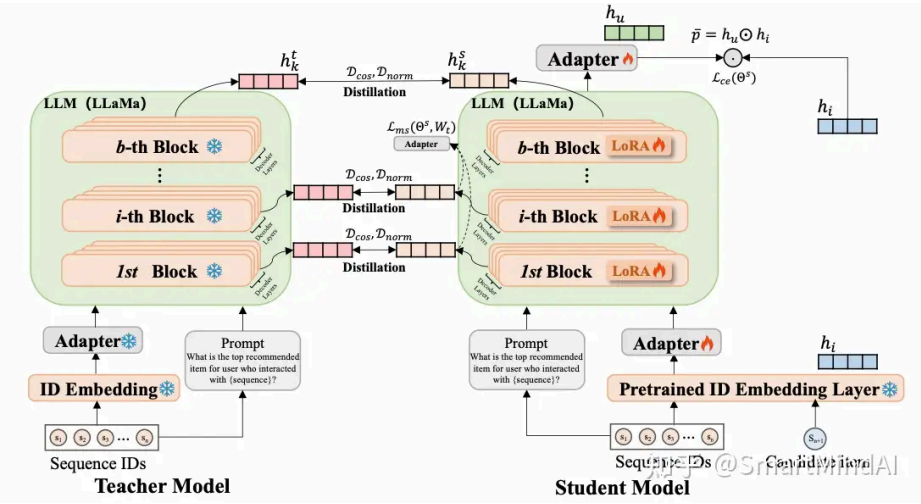
2. 提出SLMRec：我们创新地提出SLMRec，这是一种小型语言模型，通过知识蒸馏技术，减少了对LLM参数的依赖，性能优越且资源利用率高。
3. 性能与资源优化：我们通过实验验证，SLMRec仅需LLM参数的13%，就能达到与大模型相当的性能，同时大幅提升了训练和推理速度。
4. 解决实际问题：针对实时推荐的挑战，我们的工作提供了一种方法论，为如何在保持性能的同时降低对硬件资源的要求提供了解决方案。

Motivation

我们致力于研究如何通过缩小LLMs，如LLaMa-7B的规模，来探究其在推荐中的实际效果，关注的是观察这种减小参数量对性能的影响。 我们采用E4SRec*作为基础，通过去掉softmax并用嵌入点积生成推荐。首先，从SASRec预训练的嵌入获取项目嵌入，与分词后的提示嵌入拼接。之后，通过注意力块处理获取LLMs最后一层的用户表示。沿袭TSR，通过嵌入内积计算用户-项目配对得分。为减少计算负担，我们使用LoRA进行参数更新。为确保公正对比，我们选取999个无交互项目和1个真实交互项目作为评估样本，来自亚马逊18版本数据集。

评估策略集中在探究LLM参数量与性能关联。我们对E4SRec*进行减量训练，保留32层解码器不参与训练和推理，直接利用最后10层的输出作为用户表示。这样既不使用新标签，也非直接通过LLM推理，而是训练一个轻量级的E4Rec。我们构建了不同层次（$\{1, 2, 4, 8, 16, 24, 32\}$）的模型系列，分别称为E4Rec$_l^*$，以观察不同深度对推荐性能的影响。结果将以图表展示，以揭示模型复杂度对推荐效能的影响。

Preliminaries



本研究简化了E-LLMRec方法，针对序列推荐任务定制模型架构。我们采用BERT4Rec、SASRec和GRU4Rec等TSR模型的ID嵌入层，它们在特定数据集上预训练。以用户行为序列S为输入，通过截断或填充保持一致长度，生成动作序列掩码。将序列转为

$$\mathbf{S} \in \mathbb{R}^{T \times d_0}$$

通过ID嵌入和线性变换+，将其从低维提升到LLM的内在维度d_1。接着，LLM将自然语言输入转化为文本嵌入和注意力掩码，两者结合进入解码器。最后，从解码器得到的时序输出$\mathbf{h}_M$压缩回d_0，作为用户表示，通过点积+预测用户-项目交互。优化过程通过交叉熵损失+进行。

$$p_i = \frac{e^{p_i}}{\sum_{j \in I} e^{p_j}}; \quad \mathcal{L}_{ce} = -\sum_{u \in U, i \in I} y_{ui} log(p_i)$$

该研究中，他们通过最小化知识蒸馏损失$\mathcal{L}_{KD}$来优化模型。该损失度量了学生模型$f_s(\Theta^s)$与教师模型$f_t(\Theta^t)$预测的差距。目标是让学生模型f_s通过模仿大模型f_t的行为，尽管它有较少参数，来提高性能。

$$\min_{\Theta^s} [\mathcal{L}_{ce}(\Theta^s) + \mathcal{D}_{kd}(\Theta^t, \Theta^s)]$$

策略采用离线蒸馏，首先让教师模型Θ^t在大量数据上充分训练，保持其结构不变。接着，学生模型Θ^s针对每个样本，依据$\mathcal{L}_{KD}$这个目标，调整自身以减小与教师模型的差距，从而模仿其优秀性能，同时优化交叉熵损失$\mathcal{L}_{ce}$以完成标准的监督学习。

SLMRec

我们不使用基于logits的知识蒸馏，而是直接进行特征蒸馏。我们选择LLaMa模型，教师模型为深度的，学生模型为较浅的，它们具有相同的隐藏维度。为确保特征方向的一致性，我们设计了一个余弦相似性*损失$\mathcal{L}_{fsim}$，通过比较教师和学生模型每间隔i层的特征向量来测量相似性。损失项$\mathcal{L}_{fsim}$的目标是让学生模型的特征尽可能与教师模型匹配，促进特征的相似性和知识传递。

$$\mathcal{D}_{cos}(\Theta^t, \Theta^s) = \frac{1}{b}\sum_{k\in b}\frac{\mathbf{h}_k^t \cdot \mathbf{h}_k^s}{\|\mathbf{h}_k^t\|_2 \cdot \|\mathbf{h}_k^s\|_2}.$$

我们还加入了特征范数正则化，通过计算教师和学生模型每组间隔i层的特征差的平方和来量化L2距离。数学表达为$\mathcal{L}_{fnorm}$，目标是使学生模型的特征尽可能接近教师模型，以保持相似性和防止偏差。这个正则化项有助于稳定学习过程。

$$\mathcal{D}_{norm}(\Theta^t, \Theta^s) = \frac{1}{b}\sum_{k\in b}\|\mathbf{h}_k^t - \mathbf{h}_k^s\|_2^2.$$

我们引入了多源指导，通过学习额外的适应器W_t来细化学生模型的学习。预测p^{mp}通过公式

$$f(\mathbf{h}^s, W_t, \mathbf{x})$$

计算，其中$\mathbf{h}^s$是学生的基础特征W_t用于降维$\mathbf{x}$代表推荐相关的输入信息。这种方法旨在让学生模型从多个来源吸收推荐领域的专业知识，增强其理解和生成推荐的能力，但需在增加复杂性和防止过拟合*之间找到平衡。

$$\mathcal{L}_{ms}(\Theta^s, W_t) = \frac{1}{b}\sum_{k\in b}\mathcal{L}_{ce}(y, p^{mp}).$$

总损失函数$\mathcal{L}_{total}$由三部分组成：知识蒸馏损失$\mathcal{L}_{KD}$，特征相似性损失$\mathcal{L}_{fsim}$，以及多源指导的预测损失$\mathcal{L}_{mp}$。λ_{fsim}是调节特征相似性损失权重的参数。$\mathcal{L}_{fsim}$用来确保学生特征与教师一致，而$\mathcal{L}_{mp}$关注于从推荐相关输入获取的额外知识。通过联合优化这三种损失，学生模型能既学习教师知识，又能保持自身特征质量和对推荐的理解，从而提升其泛化能力和推荐表现。

$$\min_{\Theta^s, W_t}[\mathcal{L}_{ce}(\Theta^s) + \lambda_1\mathcal{D}_{cos}(\Theta^t, \Theta^s) + \lambda_2\mathcal{D}_{norm}(\Theta^t, \Theta^s) + \lambda_3\mathcal{L}_{ms}(\Theta^s, W_t)]$$

其中，我们引入了三个超参数*λ_1 λ_2, 和λ_3，分别代表知识蒸馏损失、特征相似性损失$\mathcal{L}_{fsim}$和多源指导预测损失$\mathcal{L}_{mp}$的重要性。它们允许我们在知识学习、特征保持和理解推荐知识之间进行动态平衡。通过调整这些权重，我们可以优化模型，确保在学习教师知识的同时，保证特征质量和对推荐信息的处理，从而增进学生模型的泛化能力和推荐表现。

Experiment Setup

在实验中，我们利用亚马逊包含18个行业且规模庞大的数据集，特别是服装、电影、音乐和体育类别（链接提供）。数据集中，我们仅考虑评分高于3作为有效反馈，代表用户对商品的积极互动，并通过时间戳追踪动作的先后顺序。为保证数据质量，我们剔除了交互少于5次的用户和物品。每个用户的交互历史均分为三个时间段进行分析。

知乎

Dataset	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	Density
Cloth	1,219,678	376,858	11,285,464	0.002%
Movie	297,529	60,175	3,410,019	0.019%
Music	112,395	73,713	1,443,755	0.017%
Sport	332,447	12,314	146,639	0.008%

$|\mathcal{U}|, |\mathcal{V}|, |\mathcal{E}|$ denote the number of user, item and ratings, respectively.

知乎 @SmartMindAI

Performance Comparisons

我们对比了三种基线：首先，单领域序列推荐模型，如BERT4Rec, GRU4Rec和SASRec；其次，我们使用G-LLMRec的改进版本，即基于Open-P5_LLaMa库的LLaMa，以确保评估最优表现；最后，选择E-LLMRec中的E4SRec作为另一个参考。这些基线的详细信息在附录中有说明。由于G-LLMRec和E-LLMRec内部差异不大，我们主要关注的是如何高效利用语言模型，因此选择了生成型（Open-P5）和嵌入型（E4SRec）这两种代表性的方法作为对照。

Methods	Cloth Dataset						Movie Dataset						Average Improv.
	HR			NDCG		MRR	HR			NDCG		MRR	
	@1	@5	@10	@5	@10		@1	@5	@10	@5	@10		
GRU4Rec [18]	13.79	15.46	16.83	14.64	15.08	15.15	10.56	19.47	25.21	15.11	16.96	15.46	-20.26
BERT4Rec [46]	13.60	14.66	15.55	14.14	14.43	14.59	9.68	14.91	17.98	12.40	13.38	12.74	-29.84
SASRec [29]	13.08	16.94	20.26	15.01	16.08	15.76	5.57	16.80	26.85	11.17	14.42	12.08	-25.19
Open-P5$_{\text{LLaMa}}$ [54]	14.13	17.68	19.74	17.02	16.40	-	12.66	21.98	27.24	17.13	19.81	-	-9.96
E4SRec* [35]	16.71	19.45	21.86	18.09	18.86	18.77	14.74	23.79	29.09	19.45	21.16	19.74	0.00
E4SRec$_8$* [35]	15.30	18.54	21.29	16.91	17.79	17.60	13.32	22.49	28.57	17.99	19.94	18.46	-5.90
E4SRec$_4$* [35]	14.58	18.05	20.92	16.32	17.25	17.01	11.80	21.54	28.02	16.73	18.82	17.20	-10.24
SLMRec$_{8\leftarrow32}$	16.56	19.05	21.33	17.79	18.53	18.48	15.18	23.93	29.30	19.69	21.41	20.06	-0.17
SLMRec$_{4\leftarrow32}$	14.86	18.03	20.70	16.45	17.30	17.12	13.70	22.73	28.44	18.37	20.21	18.74	-6.56
SLMRec$_{4\leftarrow8}$	16.10	18.85	21.33	17.48	18.28	18.17	14.83	23.08	28.02	19.08	20.67	19.45	-2.55
$+\mathcal{D}_{norm}$: SLMRec$_{4\leftarrow8}$	16.28	19.12	21.75	17.69	18.53	18.40	14.86	23.89	30.22	19.36	21.39	19.84	-0.37
$+\mathcal{L}_{ms}$: SLMRec$_{4\leftarrow8}$	16.85	19.05	20.93	17.96	18.57	18.59	15.05	23.48	28.60	19.40	21.05	19.76	-0.85
$+\mathcal{D}_{norm}+\mathcal{L}_{ms}$: SLMRec$_{4\leftarrow8}$	16.69	19.47	21.90	18.07	18.85	18.74	15.29	24.25	30.19	19.90	21.82	20.36	+1.49

知乎 @SmartMindAI

对于RQ1的定量结果，我们发现基于LLM的推荐方法在提取序列兴趣模式方面明显优于传统TSR方法。具体来说，我们的模型$OursDS48$通过知识蒸馏技术✦，在预测层上表现优于E4SRec$_8$，提升约8%的性能。此外，即使在不改变结构的情况下引入知识蒸馏$OursDS48$仍能在E4SRec$_{32}$的基础上略胜一筹，证明小型语言模型在适当策略下能与大模型抗衡。

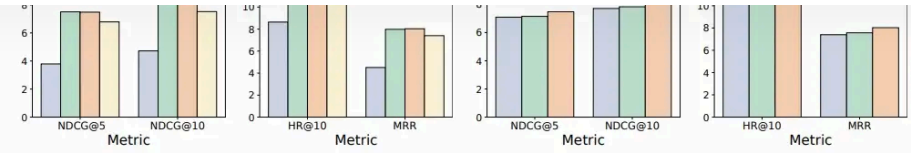
Method	Tr time(h)	Inf time(h)	Tr params (B)	Inf params (B)
Open-P5$_{\text{LLaMa}}$	0.92	4942	0.023	7.237
E4SRec*	3.95	0.415	0.023	6.631
SLMRec$_{4\leftarrow8}$	0.60	0.052	0.003	0.944

关于RQ2，模型效率对比显示，尽管Open-P5作为生成型LLMRec具有合理训练时间，但推理速度慢（4942小时，1000个候选项），不适用于大规模商品排序。相比之下，我们的SLMRec在参数量上仅为E4SRec的13%，在训练上快6.6倍，推理上快8.0倍，显示出显著的时间优势。

在RQ3的实验中，我们证实了将SLMRec与不同的知识正则化项（如$\mathcal{D}_{cos}$、$\mathcal{D}_{norm}$和$\mathcal{L}_{ms}$结合使用时，其性能有所增强。$\mathcal{D}_{cos}$和$\mathcal{D}_{norm}$通过保持中间表示与教师模型的一致性，增强了SLMRec的表示抽取能力。而$\mathcal{L}_{ms}$则通过在早期层次传递推荐系统相关领域知识，进一步优化了模型的学习。这些正则化策略的有效整合强化了SLMRec的表现。

Model Study

知乎



(a) Online KD (b) Online KD (c) Block Number (d) Block Number

知乎 @SmartMindAI

Figure 4: Model study of online KD and block number b in the music dataset.

对于RQ4，我们探讨了在线知识迁移的有效性。我们发现，即使在先对教师模型进行下游推荐任务的离线训练后，再用它指导训练SLMRec，也能获得良好的性能，验证了在线学习的可行性。至于RQ5，我们研究了块数b的影响，发现当$b = 4$时，模型表现最佳，而$b = 1$或2时，每个块的特征模仿对教师的依赖降低，导致性能下滑。

原文《SLMRec: Empowering Small Language Models for Sequential Recommendation》

发布于 2024-06-11 10:54 · IP 属地北京

LLM 序列推荐 知识蒸馏

理性发言，友善互动

还没有评论，发表第一个评论吧

推荐阅读

LLM-从大模型蒸馏到小模型的性能提升之路

南朝四百八... 发表于是NLP还...

大模型RAG入门及实践

奇舞团

大模型LLM之混合专家模型MoE（上-基础篇）

前言大模型的发展已经到了一个瓶颈期，包括被业内所诟病的罔顾事实而产生的“幻觉”问题、深层次的逻辑理解能力、数学推理能力等，想要解决这些问题就不得不继续增加模型的复杂度。 随着不...

爱吃牛油果的璐璐

【分享】大模型对齐工作去、现在和未来-2024年

刷到了Allen Institute for AI Nathan Lambert在Stanford上做讲座的slides，个人觉得LLM（尤其是开源LLM）对齐探究的发展脉络较好的梳理，趣的可以看一下： https://d

卡里奇