

RecSys'24 | 阿里MLoRA:将LoRA用于多域CTR预估

原创 州懂学习笔记 州懂学习笔记 2024年10月19日 00:27 广东



州懂学习笔记
分享大模型推荐系统相关知识和学习笔记
53篇原创内容

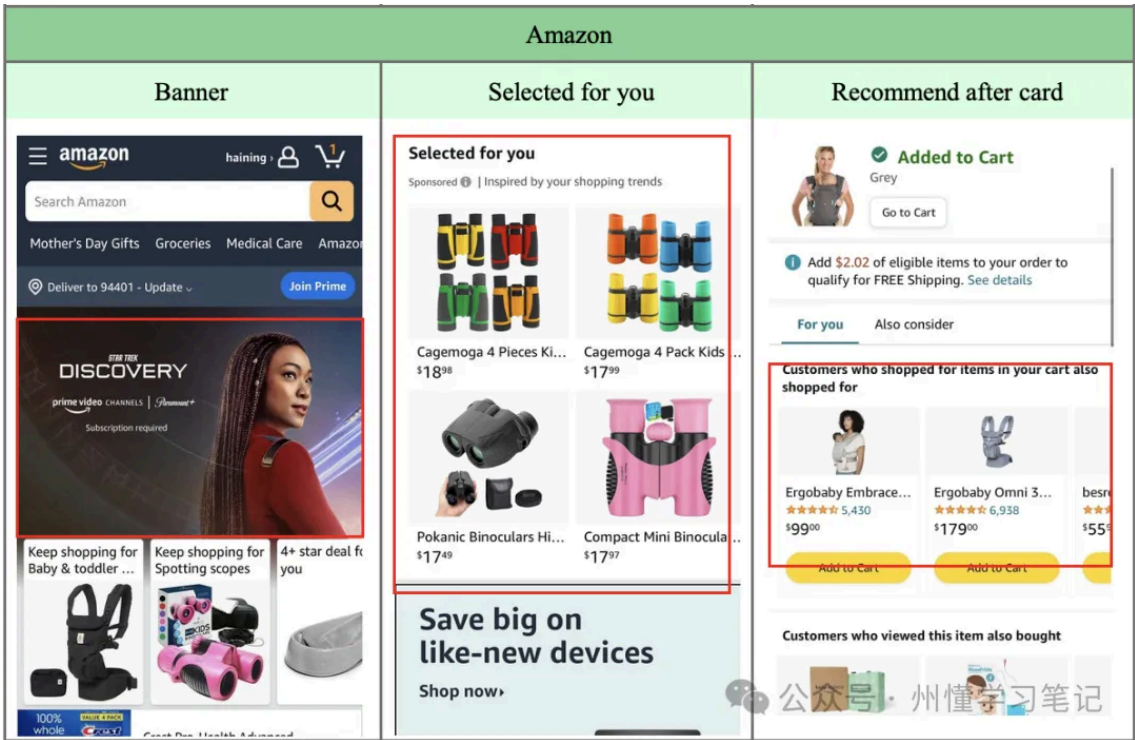
公众号

RecSys'24 | 阿里MLoRA:将LoRA用于多域CTR预估

标题: MLoRA: Multi-Domain Low-Rank Adaptive Network for Click-Through Rate Prediction
地址: <https://dl.acm.org/doi/pdf/10.1145/3640457.3688134>
公司: 阿里
会议: RecSys'24

1. 前言

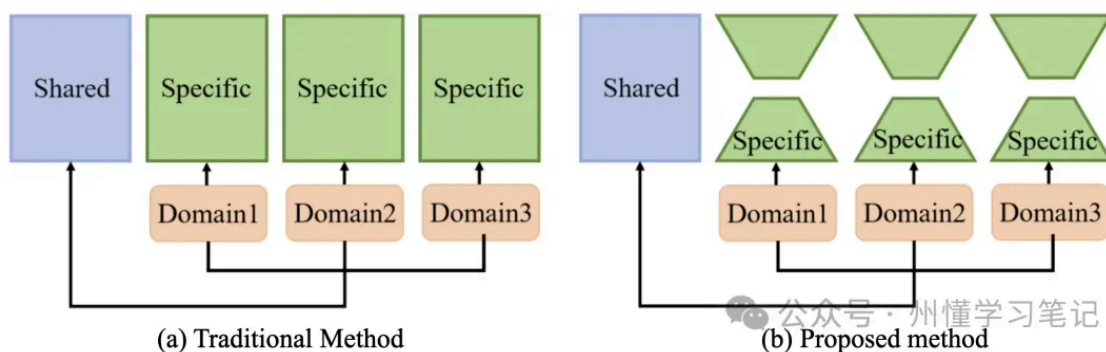
主流的App基本上都会有很多推荐场景以满足用户各方面的需求, 比如下图中亚马逊也会有Banner、Selected for you、Recommend after card等推荐场景, 这里的推荐场景也常被称域。



如果每个场景都独立建模, 会有几个弊端:

- **数据稀疏问题:** 不同场景的数据量级是不同的, 有些小场景的数据稀疏, 单场景的模型容易欠学习,
- **场景关联问题:** 各场景独立建模忽视了各场景间的内在联系, 天花板受限
- **训练&维护成本高:** 每个场景都要单独训练&维护, 新场景的接入成本过高

因此, 业内一般都会采用多域/跨域建模来解决跨场景推荐问题, 比如业内用的最多的MMoE、PLE、STAR方法, 它们会将参数划分成所有场景共享模块以及各场景内部私有模块。这些方法一定程度上解决了问题, 但会使得模型参数量显著增加, 并且对于一些数据量不足的场景, 还是会在训练不充分的问题。

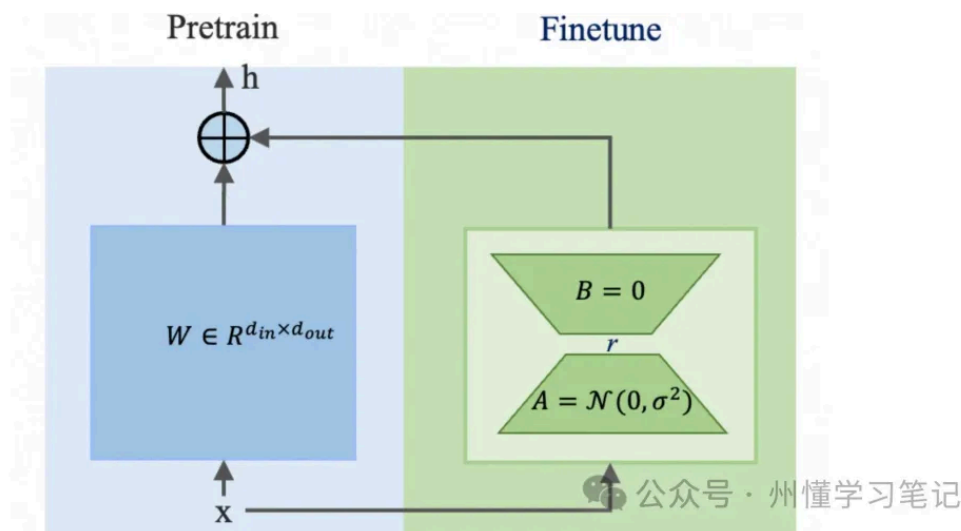


为此, 作者借鉴LLM高效参数微调的思想, 将LoRA引入进来, 为每个域都增加了个专门的LoRA模块。关于LoRA, 如有想更进一步了解的, 也可以查看公众号的历史文章《大模型微调技术: LoRA及24年顶会若干改进方法》。

2. 方法

2.1 LoRA方法简介

LoRA(Low Rank Adaption)是LLM高效参数微调的最经典方法, 它的思想很直接, 如下图所示:



左侧是预训练模型的一个参数权重 $\mathbf{W} \in \mathbb{R}^{d_{in} \times d_{out}}$, 在微调过程中是被冻结的, 不会进行梯度更新。右边是额外加的低秩权重矩阵 $\Delta \mathbf{W} = \mathbf{B}\mathbf{A}$, 这部分在Finetune的过程中, 是会更新的:

$$\mathbf{h} = \mathbf{W}'\mathbf{x} = \mathbf{W}\mathbf{x} + \mathbf{B}\mathbf{A}\mathbf{x}$$

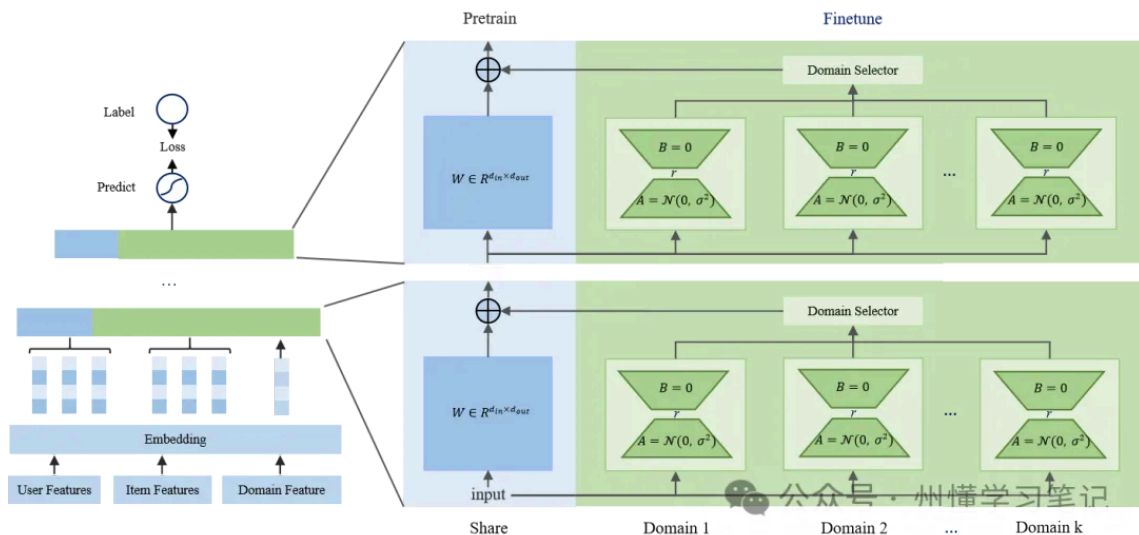
这里 \mathbf{x} 为输入, \mathbf{h} 为输出, $\mathbf{A} \in \mathbb{R}^{d_{in} \times r}$, $\mathbf{B} \in \mathbb{R}^{r \times d_{out}}$, 由线性代数的基础知识可知, 两者权重矩阵 $\mathbf{B}\mathbf{A}$ 的秩是 $\leq r$, 并且 $r \ll \min(d_{in}, d_{out})$ 。

在初始化时, 会随机高斯分布初始化 \mathbf{A} , 用 $\mathbf{0}$ 矩阵初始化 \mathbf{B} , 保证训练开始时此旁路矩阵依然是 $\mathbf{0}$ 矩阵, 且不影响微调过程中正常的梯度更新。多提一下, 如果两个矩阵都用 $\mathbf{0}$ 矩阵初始化, 那这些参数的梯度更新是有问题的。

2.2 MLoRA用于多域CTR预估

2.2.1 整体框架

多域的MLoRA的CTR模型框架如下图所示, 也是比较直观的。



2.2.2 模型&训练细节

在训练CTR模型时, 有两部分需要去学习拟合的: 泛化的公共信息以及个性化的独特信息, 这样CTR预估可以表述成:

$$\hat{y} = F(\mathbf{x}) = F_0(\mathbf{x}) + \Delta F(\mathbf{x})$$

考虑到单个域的稀疏性, 使用低秩矩阵作为参数的模型 $L(x)$ 用于学习每个域的个性化信息, 这样, 第 t 个域的CTR预估可以表示成:

$$\hat{y}_t = F_0(\mathbf{x}) + L_t(\mathbf{x})$$

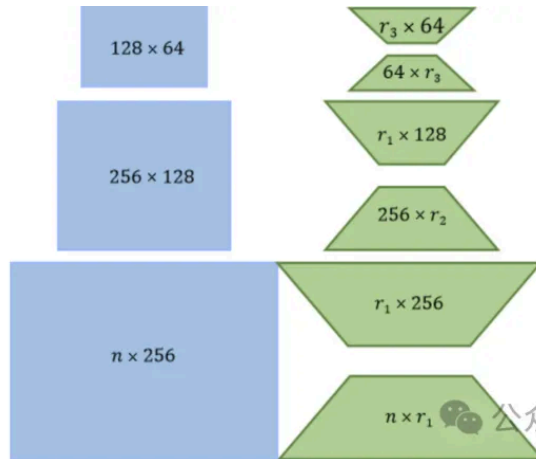
MLoRA将模型的每一层都拆分为公共部分和个性化部分, 而不是将模型作为一个整体来处理。单层可以表示为:

$$\mathbf{h}_t = \mathbf{W}\mathbf{x} + \Delta \mathbf{W}_t = \mathbf{W}\mathbf{x} + \mathbf{B}_t \mathbf{A}_t \mathbf{x}$$

与NLP模型不同, CTR模型通常在层之间的网络宽度是不同的, 因此在不同层之间的秩 r 在设

计时是不同的。本文设计了一个温度系数 α 来计算 r

与NLP模型不同，CTR模型通常在层与层之间的网络宽度是不一样的，如下图所示，因此，不同层使用同样的秩 r 肯定是不合理的。



为此，作用在MLoRA采用了固定比例因子，称为温度系数 α ，这样，各个层的秩 r 可以通过下式计算：

$$r = \max\left(\frac{d_{out}}{\alpha}, 1\right)$$

MLoRA采用两阶段训练策略。在预训练阶段，骨干网络使用大规模预训练数据进行训练，以学习可以泛化到各个领域的信息。在微调阶段，添加了MLoRA网络，同时冻结骨干网络。微调阶段只关注 A 和 B 更新来学习每个域的个性化信息。

可能有读者会有疑问，骨干网络完全冻结住不会影响性能吗，这里笔者的理解是，CTR模型的参数分成Sparse参数和Dense参数，MLoRA方法针对的就是对Dense 参数的处理逻辑，Sparse 参数还是会例行会更新的，而Dense参数一般又都是比较稳定的，在例行更新时，更新 A 和 B 就可以让模型的Dense参数也能稳定更新，性能理论上不会折损太多。

3. 实验

3.1 整体效果

在各种主干网络下，MLoRA的效果提升

Dataset	Approach	MLP	STAR	WDL	NFM	AutoInt	PNN	DCN	FiBiNET	DeepFM	xDeepFM	Avg
Taobao-10	Base	72.92	75.51	73.12	76.88	75.56	76.42	72.07	76.51	75.01	75.26	74.93
	Base+MLoRA	74.53	76.17	73.51	77.08	75.83	76.74	72.14	76.77	75.66	75.64	75.41
	Δ	+1.61	+0.66	+0.39	+0.20	+0.27	+0.32	+0.07	+0.26	+0.65	+0.38	+0.48
Amazon-6	Base	75.07	77.26	73.25	65.07	74.10	74.03	75.61	74.93	73.55	74.64	73.75
	Base+MLoRA	77.48	77.28	74.25	67.41	74.46	75.18	75.62	75.27	74.08	74.83	74.58
	Δ	+2.41	+0.02	+1.00	+2.34	+0.36	+1.15	+0.01	+0.34	+0.53	+0.19	+0.83
Movielens-gen	Base	80.15	80.18	80.17	80.57	80.25	80.26	80.28	80.47	80.24	80.26	80.28
	Base+MLoRA	80.39	80.36	80.41	80.59	80.41	80.39	80.42	80.74	80.46	80.44	80.46
	Δ	+0.24	+0.18	+0.24	+0.02	+0.16	+0.13	+0.14	+0.27	+0.22	+0.18	+0.18
-	Avg(Δ)	+1.42	+0.29	+0.54	+0.85	+0.26	+0.53	+0.07	+0.29	+0.47	+0.25	+0.50

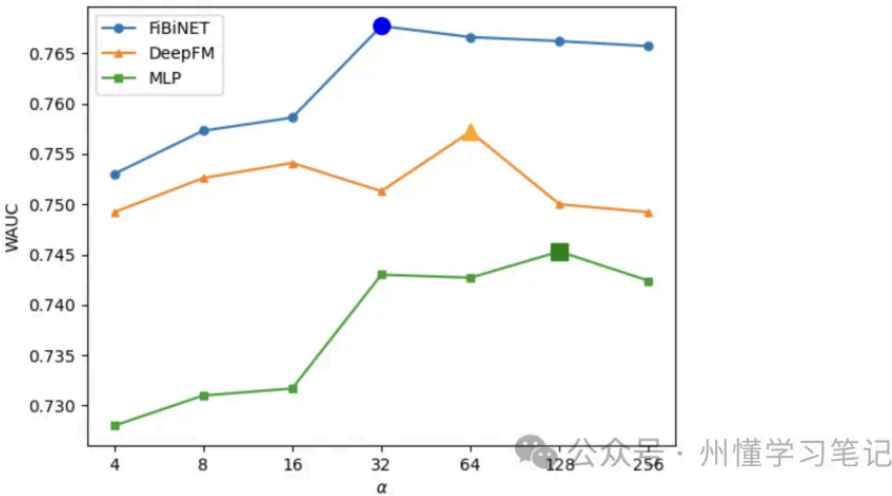
3.2 分场景效果

在各个子场景下, MLoRA都能有稳定提升

Approach	0	1	2	3	4	5	6	7	8	9	WAUC
MLP	69.09	58.23	68.17	77.29	79.70	74.21	56.56	75.47	64.43	70.21	72.92
MLP+MLoRA	70.40	62.24	69.03	79.11	80.87	74.47	60.12	76.61	65.90	72.46	74.53
Δ	+1.31	+4.02	+0.86	+1.82	+1.17	+0.26	+3.55	+1.14	+1.47	+2.25	+1.61
FiBiNET	71.90	63.13	70.43	79.28	81.19	78.11	68.04	82.54	65.42	76.08	76.51
FiBiNET+MLoRA	72.56	63.70	70.65	79.35	81.20	78.58	68.58	83.25	65.55	76.08	76.77
Δ	+0.67	+0.57	+0.22	+0.07	+0.01	+0.47	+0.54	+0.71	+0.13	+0.00	+0.26
DeepFM	72.44	64.27	66.60	76.46	81.36	76.56	66.85	83.53	67.34	72.69	75.01
DeepFM+MLoRA	72.81	64.41	67.21	77.80	81.65	77.41	67.11	83.65	67.58	72.79	75.66
Δ	+0.37	+0.14	+0.61	+1.34	+0.29	+0.85	+0.26	+0.12	+0.24	+0.10	+0.65
Avg(Δ)	+0.78	+1.57	+0.56	+1.08	+0.49	+0.53	+1.45	+0.66	+0.61	+0.78	+0.84

3.3 消融实验

不同温度系数 α 的影响



在不同数量级(10/20/30)的场景下, MLoRA的效果

Dataset	Approach	MLP	STAR	WDL	NFM	AutoInt	PNN	DCN	FiBiNET	DeepFM	xDeepFM	Avg
Taobao-10	Base	72.92	75.51	73.12	76.88	75.56	76.42	72.07	76.51	75.01	75.26	74.93
	Base+MLoRA	74.53	76.17	73.51	77.08	75.83	76.74	72.14	76.77	75.66	75.64	75.41
	Δ	+1.61	+0.66	+0.39	+0.20	+0.27	+0.32	+0.07	+0.26	+0.65	+0.38	+0.48
Taobao-20	Base	77.05	77.89	77.33	78.55	77.50	79.13	79.32	79.27	79.52	79.13	78.27
	Base+MLoRA	77.54	78.66	77.83	78.84	78.09	79.21	79.49	79.29	79.81	79.45	78.62
	Δ	+0.49	+0.77	+0.50	+0.29	+0.59	+0.08	+0.17	+0.02	+0.29	+0.32	+0.35
Taobao-30	Base	76.96	77.03	77.74	78.09	78.34	78.66	75.81	76.38	76.13	79.27	77.44
	Base+MLoRA	77.15	78.29	77.90	78.32	78.36	78.72	76.46	76.59	76.25	79.29	77.73
	Δ	+0.19	+1.26	+0.16	+0.23	+0.02	+0.06	+0.65	+0.21	+0.12	+0.02	+0.29
-	Avg(Δ)	+0.76	+0.90	+0.35	+0.24	+0.29	+0.15	+0.30	+0.16	+0.35	+0.24	+0.37

预训练模型的不同参数量级的影响

MLoRA	1x	2x	3x	4x	5x
WAUC	75.39	75.82	75.99	76.68	76.93

3.4 线上AB实验

CTR相对提升+1.49%, CVR相对提升+3.37%, 付费用户数提升+2.71%。