

赞同 39



分享

2024快手：GradCraft-动态梯度平衡技术，打造卓越多任务优化，成就高效多目标模型



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

39 人赞同了该文章

收起

Introduction

推荐系统^{*}在个性化信息筛选中扮演着核心角色，极大地塑造了个人的在线体验。系统的效能往往取决于全面建模用户兴趣的能力，这通常需要同时优化反映用户满意度不同层面的多种用户反馈。例如，一个短视频推荐系统需要同时优化观看视频的时间和喜欢视频的可能性。因此，多任务学习在推荐系统中的应用越来越普遍。

多任务学习旨在同时优化多个目标。当前在推荐领域的做法主要直接应用机器学习的多任务优化技术⁺。这些方法通常从两个视角出发，以实现任务间的适当平衡，避免产生负面转移效应。第一类工作为加权损失，根据特定标准如不确定性调整梯度大小，以有效平衡不同任务的关注度。然而，这些方法在处理任务冲突时存在局限性，表现出不稳定的性能，尤其是在面对显著任务异质时。第二类工作则专注于调整梯度方向，减少任务间的负面余弦相似性⁺。然而，它们的梯度调整通常只在任务对之间进行，缺乏全局非冲突的保证，并且忽略梯度大小对方向调整的影响。这些局限性在涉及多个任务的推荐场景中会显著影响多任务优化的有效性。

鉴于现有方法的优缺点，我们将多任务优化的核心提炼为在梯度上同时实现适当的大小平衡和全局方向平衡。首先，确保在异质推荐任务中梯度大小的一致性至关重要。缺乏这种平衡可能导致某些任务压倒其他任务，从而导致推荐性能不佳。同时，同时完全解决众多推荐任务之间的方向冲突，确保梯度的全局非冲突，也是必要的。未能做到这一点可能导致某些任务之间的残余冲突，阻碍知识的转移，最终损害多任务优化的效能。

在本文中，我们引入了"GradCraft"，一种动态梯度平衡的多任务优化方法⁺。为了同时实现大小和方向的平衡，我们设计了一个训练过程，包括梯度范数⁺对齐和方向投影。首先，我们根据最大范数动态对齐所有任务的梯度范数，建立适当的大小平衡。随后，利用这个平衡的结果，我们同时应用投影来消除方向冲突，考虑所有冲突的任务，从而确保全局方向平衡。在这个训练过程中，实现方向平衡的关键在于首先实现大小平衡，避免由大小不平衡带来的干扰。在幅度平衡中，我们不追求不同任务间的绝对梯度范数对齐；相反，我们旨在防止范数差异变得过于显著（例如跨越多个数量级），从而在保留任务特异性的同时，避免某些任务的主导地位。在方向平衡中，我们实现了投影后的简单正交性⁺。我们的重点在于要求一定的正向相似度的水平，以促进任务间的知识正向转移，从而增强冲突解决。这些设计原则使我们能够实现更好的幅度平衡和冲突解决。

Preliminary

Multi-task Recommendation

多任务推荐旨在同时优化多个推荐目标。令 \mathcal{D} 表示历史数据集。 \mathcal{D} 中的每个样本以 (\mathbf{x}, \mathbf{y}) 的形式呈现，其中 \mathbf{x} 是用户-内容对的特征， $\mathbf{y} = [y_1, \dots, y_T]$ 表示用户行为的 T 个不同任务标签，如有效

$$l_i = L(f_{\theta}(\mathbf{x})_i, y_i; \mathcal{D}) \quad (i = 1, \dots, T)$$

其中 L 表示推荐损失函数, 如二元交叉熵⁺损失 (BCE) 和均方误差损失 (MSE) 。

多任务优化 为了优化多个目标, 现有方法遵循一个统一的范式: 首先, 对不同任务的梯度进行操作, 然后使用专门方法将它们组合成一个单一梯度; 随后, 根据组合结果更新模型参数。每个任务的梯度可以通过反向传播⁺获得。第 i 个任务的梯度为:

$$g_i = \nabla_{\theta} l_i \in \mathbb{R}^d$$

其中 d 表示模型参数的维度。

Gradient Balance

推荐任务经常在多个方面展现出显著的异质性, 例如数据稀疏性。这种异质性可能导致任务之间的梯度幅度差异, 并可能导致更新方向的不一致性, 从而产生潜在的负迁移效应⁺。为了减轻这些影响, 实现幅度和方向的平衡是关键。

Magnitude Balance

任务梯度 g_i 的大小评估通常基于其范数, 表示为 $\|g_i\|$ 。幅度平衡旨在保持不同任务梯度幅度的一致性, 以避免任务 i 和 j 之间出现显著的幅度差异。这种差异通过 $\|g_i\|$ 与 $\|g_j\|$ 之间的比较来体现。在AI领域中, 任务梯度和幅度平衡的概念用于确保模型训练过程中的公平性和效率, 通过调整任务的权重或难度, 使得不同任务的训练难度接近, 从而提高整体模型的性能和学习效率。

$$\|g_i\| \gg \|g_j\| \quad or \quad \|g_i\| \ll \|g_j\|$$

在幅度平衡缺失的情况下, 特定任务可能在优化过程中占主导, 因此使得推荐性能不理想。

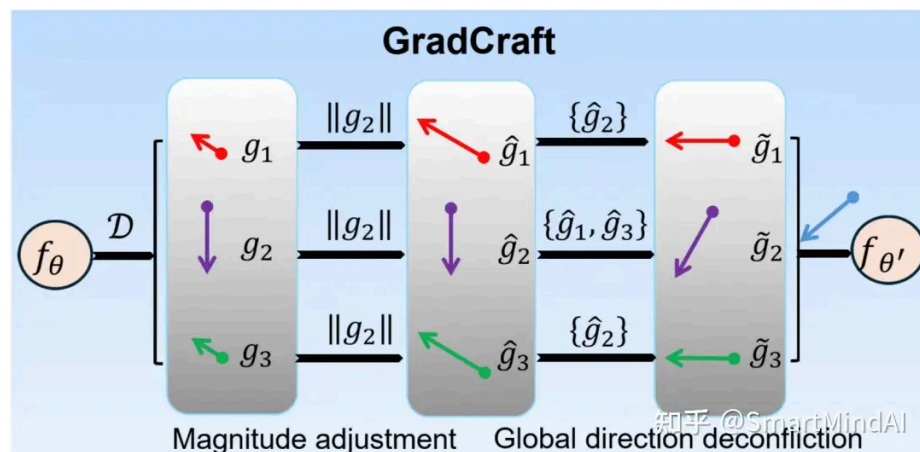
Direction Balance

方向平衡旨在避免不同任务之间的冲突, 这些冲突由两个任务梯度之间的负余弦相似度⁺定义。具体来说, 如果两个任务梯度 g_i 和 g_j 之间的内积⁺满足以下条件, 则它们被认为是处于冲突状态:
 $g_i \cdot g_j < 0$ 当 $g_i \cdot g_j < 0$ 时, 冲突发生: $\langle g_i, g_j \rangle < 0$

实现方向平衡需要消除这些负面相似性。在缺乏方向平衡的情况下, 不同推荐任务之间的知识转移可能会受到影响, 最终导致多任务优化效果的下降。

Methodology

Overview



最大范数动态地对所有任务的梯度幅度进行调整, 从而在幅度上建立适当的平衡。其次, 利用这个平衡的结果, 我们同时使用投影操作来消除梯度冲突, 确保在整体的方向上实现全局的平衡。最后, 我们将梯度合并, 并更新推荐模型。由于我们的方法在梯度层面操作, 我们将其命名为"GradCraft"。

Magnitude Adjustment

为了减轻由任务间梯度幅度差异引发的干扰, 我们主要关注的是调整梯度, 以实现幅度的平衡, 达到合适的程度。我们不追求梯度范数的绝对统一, 而是旨在避免梯度范数间的过度差异, 比如差异跨越多个数量级。这有助于防止某些任务的主导地位, 同时保持任务的特性。为了达到这个目的, 我们为每个任务调整梯度范数, 将其原始范数与任务间的最大范数相结合。形式上, 调整方式如下: 1. 确定每个任务的原始梯度范数。2. 找出所有任务中梯度范数的最大值。3. 将每个任务的原始梯度范数与最大梯度范数进行比较, 根据需要进行调整, 以确保幅度的平衡达到合适的程度。

$$\hat{g}_i = \tau \frac{\max_j \|g_j\|}{\|g_i\|} g_i + (1 - \tau) g_i$$

其中 g_i 表示任务 i 的原始梯度, $\max_j \|g_j\|$ 是所有任务中梯度范数的最大值, \hat{g}_i 表示调整后的任务梯度。超参数 $\tau \in [0, 1]$ 根据验证性能进行调整。通过这种方式, 我们确保任务之间的最大和最小梯度范数的差异不超过 $\frac{1}{\tau}$ 倍。

Global Direction Deconfliction

调整梯度幅度后, 我们的目标是实现全局梯度平衡。对于每个任务, 我们使用投影来确保其梯度不会与其他任务的梯度冲突。然后, 我们线性组合所有任务的无冲突梯度来更新最终模型。梯度投影 对于给定的任务梯度 \hat{g}_i , 我们将与其冲突的梯度表示为

$$G_i = [\hat{g}_{i_1}, \dots, \hat{g}_{i_n}] \in \mathbb{R}^{n \times d}$$

其中 \hat{g}_{i_j} 表示第 j 个冲突梯度。我们定义一个投影目标, 以实现无冲突梯度与所有冲突梯度之间的非负相似性:

$$G_i \tilde{g}_i^T = z,$$

$$z = [\epsilon \|\hat{g}_i\| \|\hat{g}_{i_1}\|, \dots, \epsilon \|\hat{g}_i\| \|\hat{g}_{i_n}\|],$$

其中 \tilde{g}_i 代表去冲突的任务梯度, $\epsilon \geq 0$ 作为调整相似度的因子, 值越高表示期望的正相似度越高。我们关注的是, 通过 ϵ 的调整, 强调任务间的知识正向转移, 而非仅追求梯度正交 ($\epsilon = 0$)。

理论上, 期望的梯度 \tilde{g}_i 可以通过原始梯度与所有冲突梯度在线性空间上的投影求和得到, 这可以表示为:

$$\tilde{g}_i = \text{原始梯度} + \text{冲突梯度的线性空间投影}$$

$$\tilde{g}_i = \hat{g}_i + \sum_{k=1}^n w_k \hat{g}_{i_k} = \hat{g}_i + \mathbf{w}^T G_i$$

其中 $\mathbf{w} \in \mathbb{R}^{n \times 1}$ 是一个需要被确定的权重向量。权重向量 \mathbf{w} 的具体形式需要通过这些等式来确定:

$$G_i G_i^T \mathbf{w} = -G_i \hat{g}_i^T + z$$

鉴于模型参数的维度远大于任务的数量, 即 $d \gg n$, 合理地假设矩阵 G_i 为满秩矩阵。因此, 可以确保 $G_i G_i^T$ 的正定性, 使得权重向量 \mathbf{w} 能够通过闭式解法得到:

$$\mathbf{w} = (G_i G_i^T)^{-1} (-G_i \hat{g}_i^T + z)$$

在获取所有任务的无冲突梯度后, 我们线性组合它们, 并通过聚合梯度更新模型, 公式表示为:

$$\theta' = \theta - \eta \frac{1}{T} \sum_{i=1}^T \tilde{g}_i$$

其中 η 表示学习率。

我们的梯度投影方法可以视作是对正常投影方法的拓展。在特定条件下, 我们的方法可能接近于正常投影方法的近似等价。具体而言, 当每个给定的任务梯度 \mathbf{g}_i 仅与之冲突一个梯度 \mathbf{g}_{i_1} , 并且将超参数 ϵ 设置为零时, 方程中的梯度计算为:

$$\tilde{\mathbf{g}}_i = \hat{\mathbf{g}}_i - \frac{\langle \hat{\mathbf{g}}_i, \hat{\mathbf{g}}_{i_1} \rangle}{\|\hat{\mathbf{g}}_{i_1}\|} \hat{\mathbf{g}}_{i_1}$$

相比之下, 我们的方法同时处理所有冲突任务, 而每个任务都需要一定的正相似度, 从而实现全局和彻底的冲突解决。值得注意的是, 我们的方法并未显著增加额外的计算复杂性⁺。考虑到:

$$\mathbf{G}_i \mathbf{G}_i^T \in \mathbb{R}^{n \times n}$$

其中 n 是 \mathbf{g}_i 冲突任务梯度的数量, 我们可以在式子中高效地计算其逆, 并获得解冲突的梯度。

Experiment

Experimental Setting

Datasets

在开放世界数据集以及我们的产品数据集, 如微信和快手, 上进行了大量的实验。

在短视频推荐中, 有两类任务: 与观看行为相关的任务和与互动行为相关的任务。因此, 我们将用户使用时间和参与度设定为优化目标, 通过观看标签和参与标签进行评估。具体来说, 我们选择EffectiveView (EV)、LongView (LV) 和CompleteView (CV) 作为观看标签。EV表示一个例子的观看时间是否超过了数据集整体观看时间的50%, LV表示是否超过了75%。CV则表示一个例子的观看时间是否超过了视频的时长。对于参与标签, 我们直接使用Like、Follow和Forward。所有上述标签均为二进制, 并适合使用BCE损失函数。

Performance Comparison (RQ1)

我们首先评估了比较方法在多目标优化领域的整体性能, 根据表格1中的总结, 我们得出了以下几点观察:

Wechat												
Method	Single	EW	UC	DWA	MGDA	PCGrad	PCGrad+	GradVac	CAGrad	IMTL	DBMTL	GradCraft
AUC	EV	0.7641	0.7641	0.7633	0.7646	0.7569	0.7651	0.7644	0.7648	0.7629	0.7636	0.7653
	LV	0.8484	0.8484	0.8479	<u>0.8490</u>	0.8429	0.8491	0.8486	0.8489	0.8489	0.8478	<u>0.8490</u>
	CV	0.7610	0.7604	0.7596	0.7620	0.7515	0.7614	0.7611	0.7613	0.7614	0.7589	<u>0.7616</u>
	Like	0.8661	0.8664	<u>0.8671</u>	0.8656	0.8604	0.8675	0.8668	0.8665	0.8662	0.8669	0.8650
	Fol	0.8829	0.8810	0.8763	0.8809	0.8803	0.8825	0.8827	0.8791	0.8801	0.8827	0.8888
	For	0.8940	0.9012	0.9006	0.8983	0.8937	0.8968	0.9000	0.8991	0.9003	0.9008	0.9001
	AV-A	0.8361	0.8369	0.8358	0.8367	0.8309	0.8371	<u>0.8373</u>	0.8366	0.8369	0.8367	0.8385
	RI-A	0.000%	0.091%	-0.038%	0.078%	-0.639%	0.118%	0.135%	0.065%	0.099%	0.056%	0.278%
GAUC	EV	0.6207	0.6209	0.6194	0.6189	0.6055	0.6226	0.6195	0.6218	0.6200	0.6201	0.6178
	LV	0.7731	0.7745	0.7740	0.7739	0.7684	<u>0.7754</u>	0.7736	0.7755	0.7743	0.7742	0.7732
	CV	0.6499	0.6503	0.6489	0.6499	0.6345	<u>0.6515</u>	0.6493	0.6509	0.6491	0.6488	0.6518
	Like	0.6324	0.6382	<u>0.6405</u>	0.6368	0.6328	0.6422	0.6380	0.6384	0.6390	0.6393	0.6385
	Fol	0.6847	0.6820	<u>0.6962</u>	0.6915	0.6874	0.6899	0.6870	0.6721	0.6930	0.6894	0.7003
	For	0.7012	0.7129	0.7154	0.7141	0.7021	<u>0.7164</u>	0.7140	0.7152	0.7135	0.7144	0.7124
	AV-G	0.6770	0.6798	0.6824	0.6809	0.6718	<u>0.6830</u>	0.6802	0.6790	0.6815	0.6810	0.6842
	RI-G	0.000%	0.413%	0.791%	0.559%	-0.809%	<u>0.887%</u>	0.472%	0.288%	0.653%	0.589%	1.056%
Kuaishou												
Method	Single	EW	UC	DWA	MGDA	PCGrad	PCGrad+	GradVac	CAGrad	IMTL	DBMTL	GradCraft
AUC	EV	0.7569	0.7581	0.7582	0.7575	0.7400	0.7558	0.7564	0.7556	0.7560	0.7579	0.7568
	LV	0.8263	0.8269	0.8275	0.8266	0.8143	0.8263	0.8265	0.8264	0.8266	0.8273	0.8265
	CV	0.8550	0.8559	0.8561	0.8555	0.8421	0.8551	0.8548	0.8551	0.8548	<u>0.8560</u>	0.8547
	Like	0.9347	0.9287	0.9310	0.9303	0.9297	0.9325	<u>0.9345</u>	0.9329	0.9340	0.9307	0.9347
	Fol	0.8322	0.8463	0.8503	0.8469	0.8430	0.8444	<u>0.8586</u>	0.8437	0.8581	0.8503	0.8592
	For	0.8156	0.8180	0.8163	0.8133	0.8118	0.8241	<u>0.8302</u>	0.8239	0.8288	0.8171	0.8309
	AV-A	0.8368	0.8390	0.8399	0.8384	0.8302	0.8397	<u>0.8435</u>	0.8396	0.8431	0.8399	0.8438
	RI-A	0.000%	0.280%	0.383%	0.197%	-0.817%	0.355%	0.811%	0.342%	0.758%	0.379%	0.844%
GAUC	EV	0.6724	0.6746	0.6749	0.6738	0.6546	0.6721	0.6715	0.6718	0.6719	0.6742	0.6730
	LV	0.7798	0.7800	0.7810	0.7797	0.7689	0.7797	<u>0.7802</u>	0.7794	0.7800	0.7801	0.7799
	CV	0.8317	0.8317	0.8326	0.8313	0.8223	0.8316	0.8315	0.8317	0.8316	<u>0.8321</u>	0.8314
	Like	0.6556	0.6617	0.6621	0.6616	0.6417	0.6661	0.6605	<u>0.6647</u>	0.6574	0.6624	0.6621
	Fol	0.5987	0.6443	0.6529	<u>0.6603</u>	0.6176	0.6349	0.6525	0.6297	0.6490	0.6629	0.6375
	For	0.5714	0.6318	0.6287	0.6299	0.6108	0.6393	0.6422	0.6405	<u>0.6370</u>	0.6253	0.6450
	AV-G	0.6849	0.7040	0.7054	0.7061	0.6860	0.7039	<u>0.7064</u>	0.7030	<u>0.7045</u>	<u>0.7060</u>	0.7071
	RI-G	0.000%	3.248%	3.451%	3.601%	0.457%	3.243%	<u>3.671%</u>	3.087%	3.351%	3.594%	3.791%

- GradCraft展现了卓越的性能, 相较于基线, 在两个数据集上的表现均优于之, 特别是在AV-A、AV-G、RI-A和RI-G的指标上。这表明其在适配度和全局方向平衡方面的能力出众, 证明了其在多任务优化的有效性。

- 现了全面而彻底的方向解冲突。
- 然而，损失重权重方法，如EW、UC和DWA，表现不佳。它们依赖于总体损失值，而没有进行粒度级的梯度分析，这限制了它们在提升多任务优化的效果方面的有效性。这凸显了考虑更精细的梯度大小和方向对于提高性能的至关重要性。
 - 专注于单一的幅度平衡或方向平衡的方法，在多任务推荐中可能难以实现最佳推荐性能，甚至可能导致性能退化（MGDA）。这强调了在多任务推荐中全面解决幅度与方向平衡的紧迫性，以避免性能的恶化。

Online Experiment

我们生产平台上进行了一项在线A/B实验，利用了超过1500万用户的流量。我们评估了三个关键指标：用户观看视频的平均时间（WT），有效的视频观看记录数量（VV），以及视频分享的次数（Share）。我们的发现，如表所示，我们的方法取得了显著成果，相比快手实现的多任务学习基准，我们的方法实现了卓越的成果。

	WT	VV	Share
Base	-	-	-
GradCraft	+0.505%	+0.950%	+1.746%

编辑于 2024-09-04 11:12 · IP 属地北京

快手 多目标优化 工业级推荐系统



理性发言，友善互动



发布



还没有评论，发表第一个评论吧

推荐阅读

coder-only foundation model for time-series forecastin

Abhimanyu Das¹ Weihao Kong¹ Rajat Sen¹ Yichen Zhou¹

Google时序预测基础模型

智能运维之时间序列预测中的经典时序模型

技术黑板报第七期

查看详情



基于多任务深度学习的时流量预测