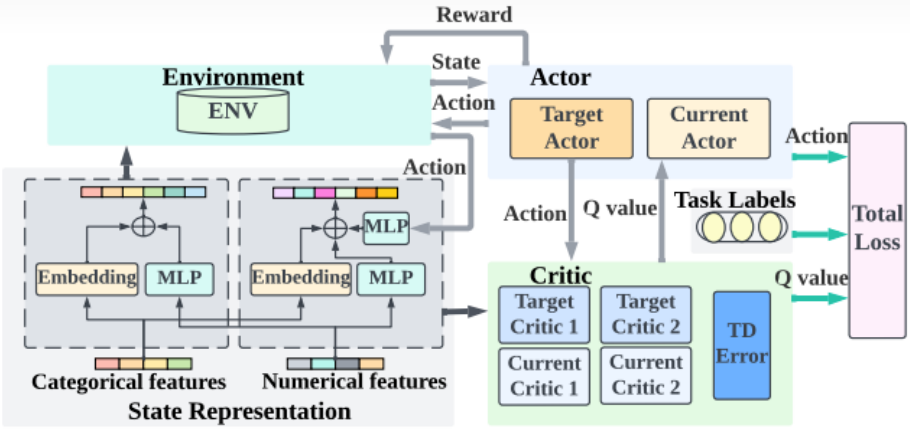


快手2023-RMTL框架：推荐系统多任务学习之旅，强化学习点亮新思路

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

论文原文《Multi-Task Recommendations with Reinforcement Learning》

Introduction

近年来，研究人员提出了许多推荐技术，包括协同过滤，矩阵分解，以及深度学习的推荐（例如YouTube）。RS的主要目标是优化特定的推荐对象，例如点击率（CTR）和用户转化率（UCR）。然而，用户通常对单个项目具有不同的交互行为。例如，短视频推荐服务中，用户表现出各种各样的行为，例如点击、点赞和停留时间；这些潜在问题促使研究和行业界开发用于推荐系统的多任务学习（MTL）技术。

现有的MTL作品通常将多个任务损失函数线性组合，用常数权重来优化目标函数，这可能会导致预测性能有限。 另一方面，现有的MTL模型通常将输入视为特征嵌入和用户-物品交互，而忽略了按顺序组织的数据（即逐个会话输入）。例如，短视频用户的点击和转化行为通常发生在特定的会话期间，因此他们的输入也是时间相关的。然而，这会降低MTL模型的性能。 现有的MTL模型主要关注网络结构设计以提高泛化能力，而研究逐个会话模式来增强多任务预测权重的方法尚未得到足够的重视。

为了解决上述两个问题，我们提出了一个强化学习增强的多任务推荐框架**RMTL**，该框架能够将用户-物品交互的顺序属性纳入MTL推荐中，并自动更新整体损失函数的逐任务权重。因此，我们的RMTL框架可以将会话式推荐系统（RS）数据转换为MDP方式，并训练一个演员-评论家框架来为多任务损失函数生成动态权重。为了实现多任务输出，我们采用双塔式MTL主干模型作为演员网络，每个任务由两个不同的评论家网络进行优化。与现有的具有物品级输入和恒定损失函数权重设计的MTL模型不同，我们的RMTL模型从会话式MDP输入中提取序列模式，并自动更新每个数据批次的损失函数权重。

THE PROPOSED Framework

RMTL框架通过动态调整损失函数权重的会话式多任务预测，有效解决了现有工作的瓶颈。RMTL框架可以更加灵活地处理多任务学习问题，并且能够根据不同的任务动态地调整损失函数的权重，从而更好地平衡了不同任务的学习效果。在会话式多任务预测方面，RMTL框架也可以更加有效地利用历史信息，并且可以更加准确地预测未来的结果。相比现有的方法，RMTL框架可以更好地处理多任务学习问题，并且具有更好的泛化性能。

Preliminary and Notations

累积损失来优化目标。定义包含K-任务（在CTR/CTCVR预测设置中，K=2）预测数据集D，其中包含N个用户-物品交互记录，每个记录包含K个二进制标签y1,...,yK。将会话定义为会议记录，每个会话包含T个不同的离散时间戳：tm,t,...,tT，其中t表示会话m中的第t个时间戳。相应的标签表示为(y1,...,yK)。由θ1,...,θK参数表示的所有会话的会话级别损失函数定义为：

$$\arg\min_{\{\theta_1^s,\ldots,\theta_K^s\}} \mathcal{L}^s(\theta_1^s,\cdots,\theta_K^s) = \sum_{k=1}^{K}(\sum_{m=1}^{M}\sum_{t=1}^{T_m}\omega_{k,\tau_{m,t}}^s L_{k,\tau_{m,t}}^s(\theta_k^s))$$
$$s.t. \quad L_{k,\tau_{m,t}}^s(\theta_k^s) = -[y_{k,\tau_{m,t}}^s log(z_{k,\tau_{m,t}}^s(\theta_k^s)) + (1-y_{k,\tau_{m,t}}^s)log(1-z_{k,\tau_{m,t}}^s(\theta_k^s))]$$

ω_{k,τ_m}^s 由强化学习框架估计，用几个动态批评网络调整输出。$z_{n,k}^s(\theta_k^s)$表示任务k在n-th数据点上的会议预测值。

Framework Overview

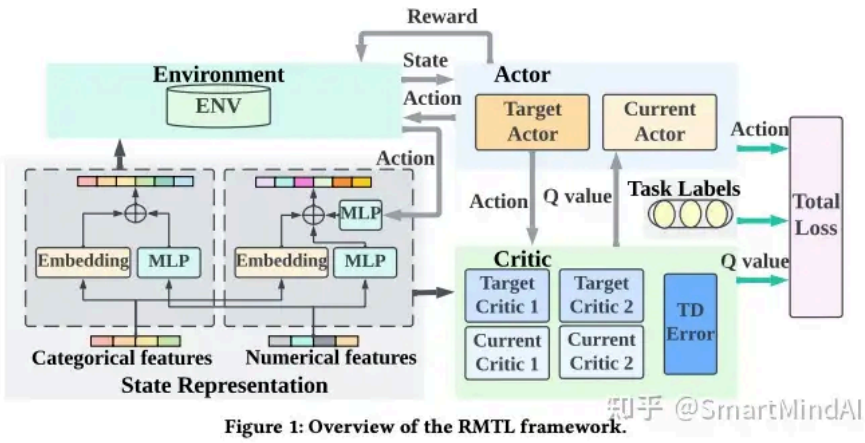


Figure 1: Overview of the RMTL framework.

图中的RMTL框架，包括状态表示网络、执行者网络和评论家网络。RMTL的学习过程概括为两个步骤：

- **前向步骤** 给定用户-物品组合特征，状态表示网络基于时间戳t的输入特征生成状态st。然后，我们从提取状态信息的演员网络中获取动作(a1t,a2t)。动作值(a1t,a2t)和用户-物品组合特征被进一步由MLP层和嵌入层处理，作为评论家网络的输入，从每个任务k的评论家网络Q(st,at;φk)计算Q值。最后，多任务的整体损失函数L(θ1,θ2)可以根据每个任务的BCE损失和由Q(st,at;φk)控制的重叠权重ωkt进行估计。
- **后退步骤** 我们首先根据TD误差σ和Q值的梯度来更新评论家网络的参数φk。然后，我们针对整体损失函数优化演员网络参数θk

Markov Decision Process (MDP) Design

通过将多任务预测数据转换为马尔可夫决策过程⁺（MDP）格式$(\mathcal{S},\mathcal{A},\mathcal{P},\mathcal{R},\gamma)$，使其适用于强化学习框架。

- 状态空间⁺$\mathcal{S}$ 是由状态 $s_t \in \mathbf{R}^d$ 组成，其中包含用户-物品组合特征，而 d 是组合特征维度；
- 动作空间 在我们的CTR/CTCVR预测设置中，动作空间$\mathcal{A}$是连续动作对$(a_{1,t},a_{2,t}) \in [0,1]^2$的集合，包含两个特定预测任务的动作。其中，$\mathcal{A}$中的每个元素表示CTR和CTCVR的预测值。
- 奖励 $r_{k,t} = y_{k,t}log(a_{k,t}) + (1-y_{k,t})log(1-a_{k,t}), \quad k=1,2$
- Transition 的转移概率 $\mathcal{P}_{ss'}$ 是基于用户交互序列从状态 s 到状态 s' 的转移概率，其中 χ 是指示函数⁺，当下一个状态（项目）与序列相对应时，将概率设置为 1。
- 贴现率γ被设为0.95。

RMTL模型学习最优策略$\pi(s_t;\theta^*)$，将物品-用户特征$s_t \in \mathcal{S}$转换为连续动作值$(a_{1,t},a_{2,t}) \in \mathcal{A}$，以最大化加权总奖励：

$$s.t. \quad R_{k,\tau_{m,t}}(\theta_k) = r(a_{k,\tau_{m,t}}, y_{k,\tau_{m,t}}) \quad k = 1, 2$$

由于由方程定义的奖励函数是负BCE，并且动作值是由由参数θ_k参数化的策略生成的，因此上述优化问题等价于最小化会话损失函数。

Session MDP Construction

我们构建了用于RL训练的会话MDP，这可以改善MTL模型的性能。经典的MTL方法通常在将用户行为的顺序引入建模方面面临挑战，其中用户行为的timestamp高度相关。基于MDP序列的强化学习可以解决这个问题。但是，MDP的对齐顺序可能会对RL的性能产生很大的影响，因为下一个动作的决策取决于时间差(TD)。本文构建了会话MDP，它由"sessionid"组织。对于每个会话τ_m，通过存储在原始数据集中的时间戳来分隔转换记录。这种构造生成了由顺序用户行为组织的会话MDP序列，并具有整体损失权重更新的优点。对于CTR/CTCVR预测任务，即$K = 2$，假设我们有一个基于会话的语料库$D_s = \{S_{\tau_m} \mathbf{1}_{\tau_m}, U_{\tau_m} \mathbf{1}_{\tau_m}, \mathbf{I}_{\tau_m}, (\mathbf{y}_{1,\tau_m}, \mathbf{y}_{2,\tau_m})\}_{m=1}^{M}$ 具体来说，它首先将每个会话的数据根据时间戳进行分离，然后随机选择一个会话并使用函数 f 和 F 生成状态信息。该会话中记录的状态信息将按顺序输入到代理策略 $\pi(s; \theta_k)$ 中，生成估计的动作对 $(a_{1,t}, a_{2,t})$。然后，根据公式进一步计算奖励值，并将时间戳 t 的转移 $(s_t, a_{1,t}, a_{2,t}, s_{t+1}, r_{1,t}, r_{2,t})$ 存储到经验回放缓冲区 $\mathcal{B}$ 中，以提供更稳定的训练环境。这样就完成了用于 RL 训练的会话级别点击率/点击率和转化率预测的设置。

State Representation Network.

Input: 会话数据集组织如下 $\{S_{\tau_1} : (\mathbf{x}_{\tau_1}, \mathbf{y}_{1,\tau_1}, \mathbf{y}_{2,\tau_1}), \cdots, S_{\tau_M} : (\mathbf{x}_{\tau_M}, \mathbf{y}_{1,\tau_M}, \mathbf{y}_{2,\tau_M})\}$

Reorganize: 用状态标签对表示每个会话，如下所示 $(f(x_{\tau_{m,t}}), y_{1,\tau_{m,t}}, y_{2,\tau_{m,t}})$, where f 是一个映射函数，返回任何给定用户-项目的特征向量 id\

Reset: 随机采样一个会话 τ_m 并使用函数F将 F 映射到状态表示.

Input: 使用智能体策略$\pi(s; \theta_k)$中的动作值 $a_{1,t}, a_{2,t}$ 作为CTR/CTCVR的预测值

Step: 根据以下公式计算奖励 $r_{1,t}, r_{2,t}$ 、通过动作和标签，返回下一个状态s_{t+1}. 存储转换 $(s_t, a_{1,t}, a_{2,t}, s_{t+1}, r_{1,t}, r_{2,t})$ 写入回放缓冲区 完成会话τ_m

RMTL Framework

RMTL使用自适应动态权重来优化整体损失函数。通过设计恒定速率更新策略来更新权重的损失。$\omega'_{k,t} = \omega'_{k,0} \gamma'^t$ 其中$\omega'_{k,t}$代表在时间戳为t的任务k的权重，γ'是常数衰减率，然而，这个权重仅受时间t控制，与任务类型无关。

为了解决这个问题，我们使用评论家网络来评估代理策略产生的每个任务的行动a_k的估计总奖励V_k。然后，每个任务的损失函数权重的值通过波兰变量λ进行线性变换计算得出。这种新颖的整体损失设计可以更好地优化多任务学习模型参数，提高效率和预测性能。

基于强化学习的CTR/CTCVR预测训练过程由两个主要成分组成：主体网络和特定任务评论家网络。

主体网络 是我们工作中用于预测CTR/CTCVR的主要结构，可以将其视为由参数θ_k参数化的代理策略$\pi(s_t; \theta_k)$。基于该主体网络，代理可以为每个状态选择一个行动。

评论家网络（也称为可微动作值网络）表示为$Q(s_t, a_{k,t}; \phi_k)$ 是由梯度和TD误差δ更新的分区分行动值网络，可以通过学习环境和奖励之间的关系来观察当前状态的潜在奖励。

此外，它生成目标函数的不适定调整BCE损失的权重，并沿着更好策略决策的方向更新主体网络参数θ_k。

Actor Network.

(i) 在动作空间较大的环境中，行动者评论家网络的表现较差，

(ii) 估计的代理策略是由其自身预期动作价值训练得出的，并非真实数据。

通过应用一种特定的 MTL 模型作为具有确定动作输出的代理策略，解决了上述两个问题。此外，我们使用真实任务标签更新了神经网络中的网络参数。 图展示了行动者网络的结构，其与特定 MTL 模型的结构类似。

我们在此部分中采用 ESMM 模型来指定它。 在原始行动者网络设计中移除了共享底层层并将其设置为生成回放缓冲区 $\mathcal{B}$ 的状态表示网络。从回放缓冲区 $\mathcal{B}$ 中获取一批 MDP 序列的批量处理，我们将状态信息 s_t 输入到行动者网络中，该网络由两个平行的神经网络组成，由参数 θ_1 和 θ_2 表示的两个塔层表示。 $\pi(s_t;\theta_k)=a_{k,t}, k=1,2$ 在我们的设置中，每个塔的输出都是确定性的作用值，代表特定任务的预测。

具体来说，一个塔输出CTR预测值，另一个输出CTCVR预测值。在批量处理MDP序列的训练过程结束后，我们根据BCE损失的加权总和计算整体损失函数：

$$L = \alpha L_{ctr} + (1-\alpha) L_{ctcvr}$$

其中，L_{ctr}和L_{ctcvr}分别表示CTR和CTCVR预测的损失函数，α是权重参数。整体损失函数的计算方法：

$$\mathcal{L}(\theta_1,\theta_2) = \sum_{(s_t,\cdots,s_{t+1},\cdots)\in b} \sum_{k=1}^{2} \omega_{k,t} BCE(\pi(s_t;\theta_k), y_{k,t}))$$

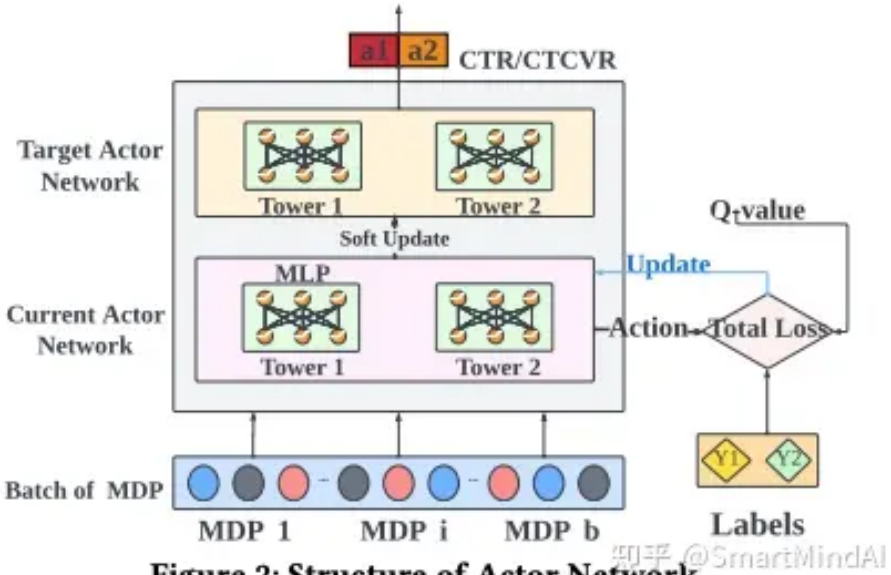


Figure 2: Structure of Actor Network.

Critic Network.

传统的评论家网络估计由演员网络产生的行动价值Q，然后根据价值Q更新演员网络参数。当我们选择MTL模型作为演员网络时，问题在于如何设计一个适合多参数更新的评论家网络结构。

本文提出了一种具有两个并行MLP网络的多元评论家结构，共享一个底层层。这种设计能够更新演员网络中MTL模型参数，并针对特定任务润色损失函数权重。评论家网络的结构如图所示Critic。评论家网络的第一部分是一个共享的底层层，它同时转换用户-物品特征和动作信息。类似于子节sss中状态表示网络srn，我们应用一个嵌入层和一个MLP结构进行特征提取。 然后，我们将用户-物品特征和动作信息组合作为两个可微分行动价值网络（由参数ϕ_k表示）的输入，它们根据每个任务的状态-动作信息输出估计的Q值。对于给定的当前状态s_t和动作$a_{k,t}$，Q值的计算如下：

$$\begin{aligned} Q'(s_t,a_{k,t}) &= \mathbb{E}[r_{k,t}+\gamma V(s_{t+1})|s_t,a_{k,t}] \\ &= r_{k,t}+\gamma \sum_{s^{t+1}\in\mathcal{S}} p_{s_t,a_t,s_{t+1}} \cdot \max Q'(s_{t+1},a_{k,t+1}) \end{aligned}$$

我们通过沿Q值的反方向调整目标损失函数的权重，来改善行为者网络的优化过程。这个调整涉及到带有惩罚变量λ的线性变换。 $\omega_{k,t} = 1 - \lambda * Q(s_t, a_{k,t}; \phi_k)$ 在给定的上下文中，$a_{k,t}$是在时间戳t处执行任务k的动作，而$Q(s_t, a_{k,t}; \phi_k)$表示状态s_t和动作$a_{k,t}$的动作值。

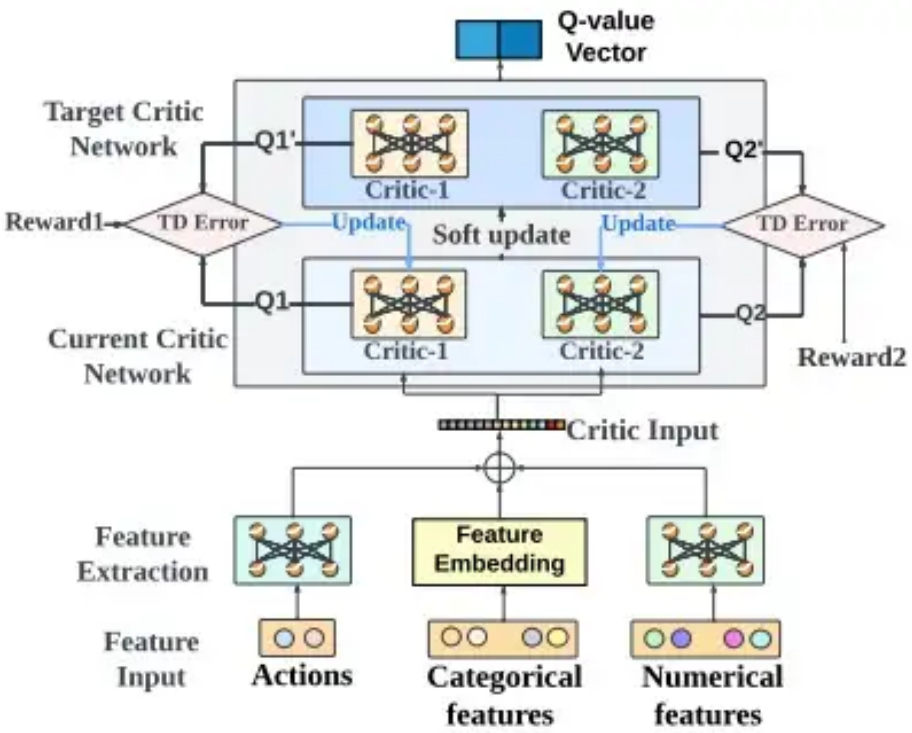


Figure 3: Structure of Critic Network.

Optimization

演员-评论家网络的一般框架通常面临一个挑战，即评论家网络的收敛无法得到保证。

为了解决这个问题，我们借鉴确定性策略梯度算法的思想，将目标网络引入学习框架。目标网络与提出的演员和评论家网络（即估计网络）具有完全相同结构，我们将其表示为 $\pi(s_t; \widetilde{\theta_k})$和 $Q(s_t, a_{k,t}; \widetilde{\phi}_k)$， 其中具有延迟参数$\tilde{\theta}_k$和$\widetilde{\phi}_k$。在本文中，我们将估计演员网络参数化为$\theta_k$和估计评论家网络参数化为$\phi_k$的主要演员-评论家网络。传统的演员-评论家网络通常会遇到参数发散的问题，并且评论家网络的收敛不能得到保证。

目标网络可以通过从记录参数神经网络生成恒定目标值来解决这些问题，从而平滑了训练过程。我们详细描述了估计演员-评论家网络模型参数的优化过程，以及目标网络的软更新。

在训练过程中，我们采用了一种基于目标Q值的目标函数，该目标函数考虑了当前状态和未来Q值的预期变化。通过梯度下降算法[+]，我们可以更新估计演员和评论家网络的参数，以最小化目标函数。此外，我们还引入了目标网络来平滑训练过程，通过使用恒定目标值来避免参数发散和评论家网络收敛的问题。

在估计演员-评论家网络模型的优化过程中，我们采用了以下步骤：首先，我们计算每个时间步的Q值和策略梯度；然后，我们使用这些梯度来更新估计演员和评论家网络的参数。在每个时间步，我们使用目标Q值来计算目标函数，并使用梯度下降算法来更新参数。我们还引入了目标网络来解决评论家网络收敛的问题。在每个时间步，我们使用恒定目标值来更新目标网络参数，以避免参数发散和评论家网络收敛的问题。 通过这些优化步骤，我们可以有效地训练演员-评论家网络模型，并解决传统演员-评论家网络遇到的问题。

$$TD_{k,t} = r_{k,t} + \gamma Q(s_{t+1}, a_{k,t+1}; \widetilde{\phi}_k)$$
$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{\pi}\left[r_{t+1} + \gamma V^{\pi}(s_{t+1}) | s_t = s, a_t = a\right]$$

$V^{\pi}(s)=\mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\gamma^{t}r_{t}|s_{0}=s\right]$

因此，策略 π 的 Q-value 可表示为：

$Q^{\pi}(s,a)=\mathbb{E}_{\pi}\left[r_{t}+\gamma V^{\pi}(s_{t+1})|s_{t}=s,a_{t}=a\right]Q_{k,t}=Q(s_{t},a_{k,t};\phi_{k})$

在回放缓冲区中重演b批次的转换后，我们计算平均TD误差δ：

$\delta=\frac{1}{b}\sum_{i=1}^{b}(y_{i}-F(x_{i}))$

其中，y_i表示第i个样本的真实标签，$F(x_i)$表示模型在第i个样本上的预测。

$$\delta=\frac{1}{2|b|}\sum_{(s_{t},\cdots,s_{t+1},\cdots)\in b}\sum_{k=1}^{2}(TD_{k,t}-Q_{k,t})$$

$$=\frac{1}{2|b|}\sum_{(s_{t},\cdots,s_{t+1},\cdots)\in b}\sum_{k=1}^{2}[r_{k,t}+\gamma Q(s_{t+1},a_{k,t+1};\widetilde{\phi}_{k})-Q(s_{t},a_{k,t};\phi_{k})]$$

然后我们用以下梯度下降法✦更新每个任务的当前评论网络：

$\phi_{k}^{t+1}=\phi_{k}^{t}-\alpha^{\phi}\mathbf{I}\delta\nabla_{\phi_{k}}Q(s_{t},a_{k,t};\phi_{k})$ 其中 $\nabla_{\phi_{k}}Q(s_{t},a_{k,t};\phi_{k})$

是第 k 个目标 Q 值的梯度。这样就完成了对估计评论家网络的优化。

估计行为者网络更新。 在时间差分误差 δ 收敛到阈值 ϵ 之前，我们通过每个批次从 $\mathcal{B}$ 发生的前向过程的每个塔层损失函数的梯度反向传播来更新当前的行为者网络，该网络由参数 θ_k 参数化：

$\theta_{k}^{t+1}=\theta_{k}^{t}+\alpha^{\theta}\mathbf{I}\nabla_{\theta_{k}}\mathcal{J}(\theta_{k})$

其中损失来自Tower层，其定义为平均Q值的负值，即

$\mathcal{J}(\theta_{k})=-\frac{1}{b}\sum_{(s_{t},\cdots,s_{t+1},\cdots)\in b}Q'(s_{t},\pi(s_{t};\theta_{k}))。$

我们进一步根据CTR/CTCVR预测的整体损失函数的梯度来更新当前的Actor网络。

$\theta_{k}^{t+1}=\theta_{k}^{t}-\alpha^{\theta}\mathbf{I}\nabla_{\theta_{k}}\mathcal{L}(\theta_{1},\theta_{2})$

$\mathcal{L}(\theta_{1},\theta_{2})=\sum_{(s_{t},\cdots,s_{t+1},\cdots)\in b}\sum_{k=1}^{2}\omega_{k,t}BCE(\pi(s_{t};\theta_{k}),y_{k,t})$

目标网络的软更新。目标执行网络和两个特定的目标评论网络在当前的评论网络达到收敛条件之前，朝着当前网络参数的方向进行更新：

$\tilde{\theta}_{k}=\beta\tilde{\theta}_{k}+(1-\beta)\theta_{k}$

$\tilde{\phi}_{k}=\beta\tilde{\phi}_{k}+(1-\beta)\phi_{k}$

Experimental Setup

Dataset.

我们发现仅有开源数据集RetailRocket（包含了点击和支付的顺序标签。为了对RMTL进行全面的比较，我们还使用了"Kuairand-1K"数据集进行进一步分析。我们将每个数据集按照时间戳t以6:2:2的比例分割成训练、验证和测试集✦，用于模型的学习和验证。

Baselines.

我们选择了两个基线模型：一个是具有默认损失的MTL模型，另一个是基于RL的模型。单任务学习方法被广泛用于比较MTL模型的性能。共享底部是MTL模型的基本结构。ESMM是针对每个任

将DDPG应用于PLE模型作为基于RL的基线。所有基线模型的损失函数都是通过取不变的加权平均值得到的。

Evaluation Metrics.

- AUC 和 Logloss 是评估 CTR/CTCVR 预测任务最直接的方法。根据 Guo 等人(2017)的研究，AUC 或 Logloss 在推荐任务中的变化具有统计意义上的显著性。
- s-Logloss: 该度量定义为所有会话的平均Logloss。

Implementation Details

我们使用公共库实现了基线模型（没有ESMM的7个标准MTL模型），并为每个模型提供了参考性能。所有模型具有相同的网络结构，即：输入嵌入层维度为128，底部使用$128 \times 512 \times 256$ MLP的多层感知机（MLP），使用$256 \times 128 \times 64 \times 1$ MLP的塔式层。具体来说，对于使用专家层的模型（与底部结构相同），我们将专家数量固定为8。隐藏层的激活函数*为ReLU，输出层的激活函数为Sigmoid。我们还设置了0.2的dropout率。我们使用Adam优化器（$lr = 1e-3$）学习所有模型。我们还实现了ESMM，共享上述模型的超参数。

由于我们的RMTL模型中的演员网络是具有多任务输出的双塔式设计，因此与大多数现有的基于MTL的推荐模型兼容。在本文中，我们将RMTL结构应用于三个代表性的MTL模型：ESMM、MMOE和PLE。

对于RMTL结构，我们设置了与指定MTL模型相同结构的演员网络。评论家网络也共享相同的底层，并通过一个塔式层输出。此外，我们添加了一个1×128动作层以与输入特征保持一致，并将输出激活更改为负ReLU，因为我们的奖励只有负值。为了提高数据效率，我们从MTL模型中初始化演员参数并保持冻结状态，直到评论家收敛。然后我们将评论家的值乘以总损失并重新训练MTL模型。演员网络的默认学习率α^{θ}和评论家网络的默认学习率α^{ϕ}设置为0.001。我们设置默认的软更新率$\beta = 0.2$，惩罚变量$\lambda = 0.7$。

Overall Performance and Comparison

Table 1: Performance on CTR/CTCVR tasks for different methods.

Dataset	Task	Metric	Methods								
			Single Task	Shared Bottom	ESMM	MMoE	PLE	D-PLE	RMTL-ESMM	RMTL-MMoE	RMTL-PLE
RetailRocket	CTR	AUC ↑	0.7273	0.7287	0.7282	0.7309	0.7308	0.7308	0.7338*	**0.7350***	0.7339*
		Logloss ↓	0.2065	0.2048	0.2031	0.2021	0.2056	0.2058	0.2024	**0.1995**	0.2013
		s-Logloss ↓	0.0846	0.0839	0.0852	0.0853	0.0827	0.0830	0.0836	0.0848	**0.0824**
RetailRocket	CTCVR	AUC ↑	0.7250	0.7304	0.7316	0.7347	0.7387	0.7386	0.7341	0.7396	**0.7419***
		Logloss ↓	0.0489	0.0493	0.0486	0.0496	0.0486	0.0490	0.0485	0.0490	**0.0480**
		s-Logloss ↓	0.0150	0.0149	0.0150	0.0145	0.0147	0.0149	0.0150	**0.0143**	0.0146
Kuairand	CTR	AUC ↑	0.7003	0.7018	0.7009	0.7014	0.7026	0.7025	0.7031	0.7029	**0.7053***
		Logloss ↓	0.6127	0.6114	0.6128	0.6119	0.6111	0.6123	0.6111	0.6105	**0.6092***
		s-Logloss ↓	0.6263	0.6250	0.6261	0.6255	0.6252	0.6257	0.6252	0.6243	**0.6242**
Kuairand	CTCVR	AUC ↑	0.7342	0.7310	0.7350	0.7324	0.7339	0.7339	**0.7377***	0.7345	0.7367*
		Logloss ↓	0.5233	0.5249	0.5220	0.5237	0.5235	0.5237	**0.5200***	0.5225	0.5221
		s-Logloss ↓	0.5449	0.5468	0.5436	0.5450	0.5454	0.5451	**0.5433**	0.5446	0.5447

↑: higher is better; ↓: lower is better. Underline: the best baseline model. **Bold**: the best performance among all models.
"*": the statistically significant improvements (i.e., two-sided t-test with $p < 0.05$) over the best baseline.

我们比较了五种基线MTL模型与RMTL模型在两个不同数据集上用于CTR/CTCVR预测任务的性能。不同方法在CTR/CTCVR任务上的整体表现如表1所示。可以看出：

1. 在两个数据集上，RMTL模型都显著优于所有基线MTL模型。

2. 特别是在数据集2上，RMTL模型的性能明显优于其他模型。

3. 这表明RMTL模型在处理跨任务相关性方面具有优势，从而在CTR/CTCVR预测任务中实现了更优秀的性能。

- 在两个数据集的预测任务中，单任务模型表现最差，因为每个任务的特征表示网络和损失函数优化是分开的，无法检测到多个任务之间的内在关系。因此，共享底层方法（共享特征提取参数）在两个数据集上都优于单任务方法。
- 在多数情况下，PLE模型在所有MTL基线模型中展现出最佳性能。PLE模型在MMoE模型的基础上，对共享底部与特定任务塔之间的参数交互进行了控制，并明确了专家层之间的参数共享，从

- 我们提出的RMTL模型在预测任务中表现优于非RL版本基准模型，如RMTL-ESMM与ESMM相比。在RetialRocket数据集上，RMTL模型在AUC上获得约0.003-0.005的收益。RL增强方法能够处理逐会话推荐数据，通过自适应调整损失函数权重，实现CTR / CTCVR预测任务的显著改进。
- 与基准模型相比，RMTL模型在s-Logloss上的改进小于0.001，比Logloss的改进要小。这可能是由于基于会话的度量与在线A/B测试具有相似的性能，较难改进。需要注意的是，由于两个数据集的平均会话长度不同，Logloss和s-Logloss之间的比率是不同的。而特定数据集的平均会话Logloss受到其平均会话长度的影响。

综上所述，RMTL模型在针对不同真实数据集的CTR和CTCVR预测任务上，均优于现有的MTL模型。此外，它可以应用于大多数MTL模型中，验证了其良好的兼容性。这表明RMTL模型具有广泛的应用前景，可以为推荐系统和广告系统中的点击率预测任务提供有效的解决方案。

编辑于 2023-11-25 12:07 · IP 属地北京

强化学习 (Reinforcement Learning) 推荐系统 多目标优化

文章被以下专栏收录

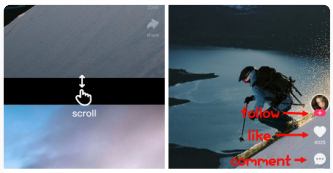
推荐+广告+搜索
搜广推文章合集

推荐阅读

第七篇：Django框架

1. Django ORM查询中select_related和prefetch_related的区别？？ def select_related(self, *fields) 性能相关：表之间进行join连表操作，一次性获取关联的数据。 总结： 1. select_related...

Egon林... 发表于必会常识



快手-2023：多目标优化不是Pareto最优，两阶段强化学...

SmartMindAI

Aquila2-34B推出Int4量化版本，低资源实现最强开源模型...

北京智源人工智能研究院

快手电商&广告2024年春ranking模型中做的端到

丁炜杰 发表于丁丁