多任务多目标模型

邸凡祎/2022.08.31

目录

- 多任务学习概念
- 改进方向 - 模型结构的设计
- 改进方向 - loss优化
- 我们怎么做

多任务学习概念

如果有n个任务(传统的深度学习方法旨在使用一种特定模型以解决一项任务)，**而这n个任务或它们的一个子集彼此相关但不完全相同,则称为多任务学习(MTL)** 通过使用所有n个任务中包含的知识，将有助于改善特定模型的学习。

单任务学习:一次只学习一个任务(task)，大部分的机器学习任务都属于单任务学习，各个任务之间的模型空间(Trained Model)是相互独立的。
多任务学习:把多个相关(related)的任务放在一起学习，同时学习多个任务，多个任务之间的模型空间(Trained Model)是共享的。

多任务学习方法分为:**hard parameter sharing**和**soft parameter sharing**

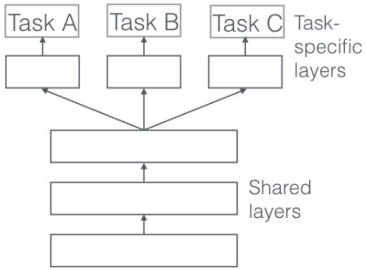


Figure 1: Hard parameter sharing for multi-task learning in deep neural networks

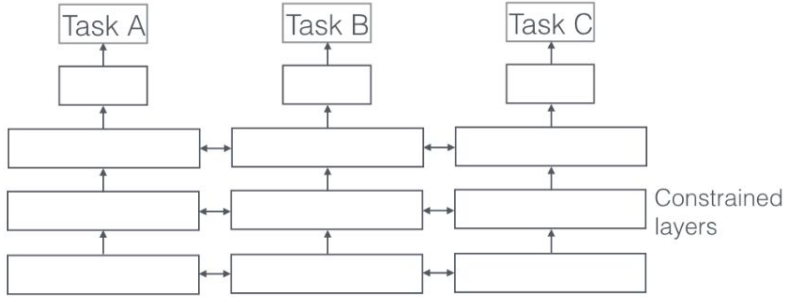


Figure 2: Soft parameter sharing for multi-task learning in deep neural networks

多任务学习概念

为什么需要多任务学习？

全局偏差/Global bias
不同目标表达不同的偏好程度

- 电商场景中，购买行为表达的偏好高于点击浏览和收藏
- 新闻场景中，浏览时长超过20s这个行为表达的偏好高于点击
- 短视频场景，完播行为表达的偏好高于点击

物品偏差/Item bias
单个目标衡量不全面

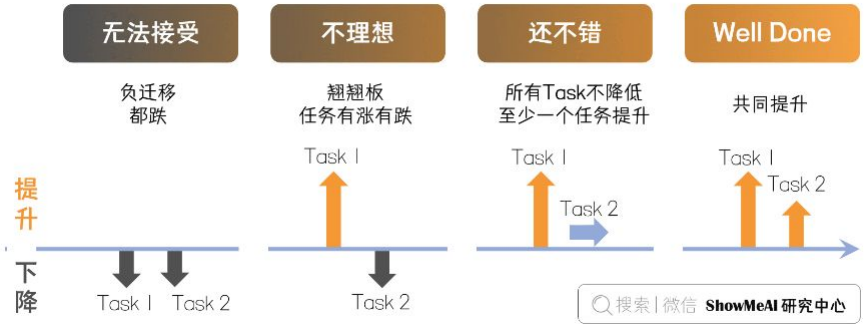
- 信息流产品中，标题党增加点击率，但降低满意度
- 短视频场景中，悬念设计提升完播率，但需要观看下一个引发用户更多操作的不满
- 自媒体资讯产品，鼓励转发率，可能会提升『转发保平安』等恶性操作

用户偏差/User bias
□户表达满意度的□式不同

- 信息流产品中，用户有 深度阅读、点赞、收藏 等不同表达满意的方式
- 短视频场景中，用户有 点赞、收藏、转发 等不同表达满意的方式

难点

- 部分目标数据稀疏，模型准确率低
- 在线服务计算量□
- 多个目标间重要性难以量化
- 分数融合的超参难以学习



多任务学习概念

为什么多任务学习是有效的？

(1)多个相关任务放在一起学习，有相关的部分，但也有不相关的部分。当学习一个任务(Main task)时，与该任务不相关的部分，在学习过程中相当于是噪声，因此，引入噪声可以提高学习的泛化(generalization)效果。

(2)单任务学习时，梯度的反向传播倾向于陷入局部极小值。多任务学习中不同任务的局部极小值处于不同的位置，通过相互作用，可以帮助隐含层逃离局部极小值。

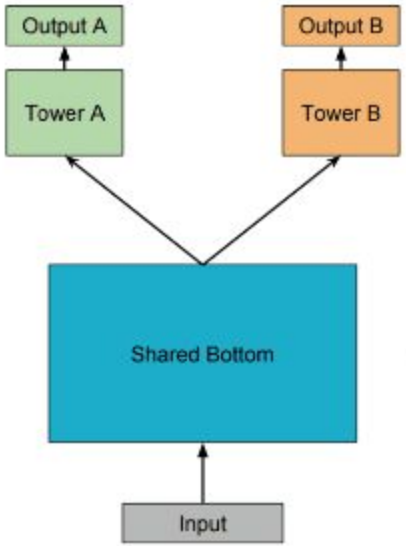
(3)添加的任务可以改变权值更新的动态特性，可能使网络更适合多任务学习。比如，多任务并行学习，提升了浅层共享层(shared representation)的学习速率，可能，较大的学习速率提升了学习效果。

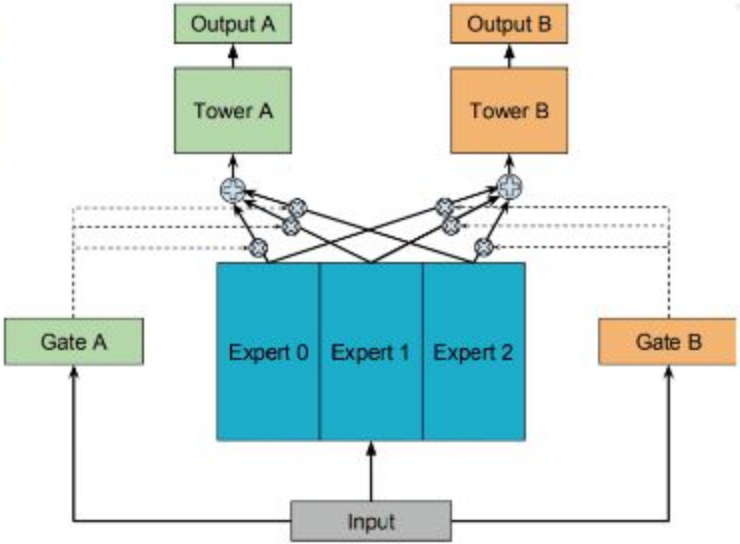
(4)多个任务在浅层共享表示，可能削弱了网络的能力，降低网络过拟合，提升了泛化效果。

(5)多任务学习假设不同任务数据分布之间存在一定的相似性，在此基础上通过共同训练和优化建立任务之间的联系. 这种训练模式充分促进任务之间的信息交换并达到了相互学习的目的，尤其是在各自任务样本容量有限的条件下，各个任务可以从其它任务获得一定的启发，借助于学习过程中的信息迁移能间接利用其它任务的数据，从而缓解了对大量标注数据的依赖，也达到了提升各自任务学习性能的目的。

改进方向-模型结构设计

多头模型结构：share-bottom->mmoe->ple





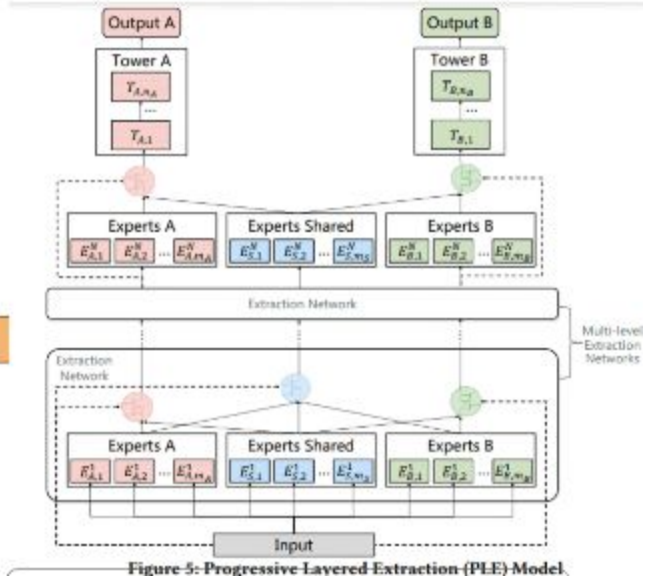
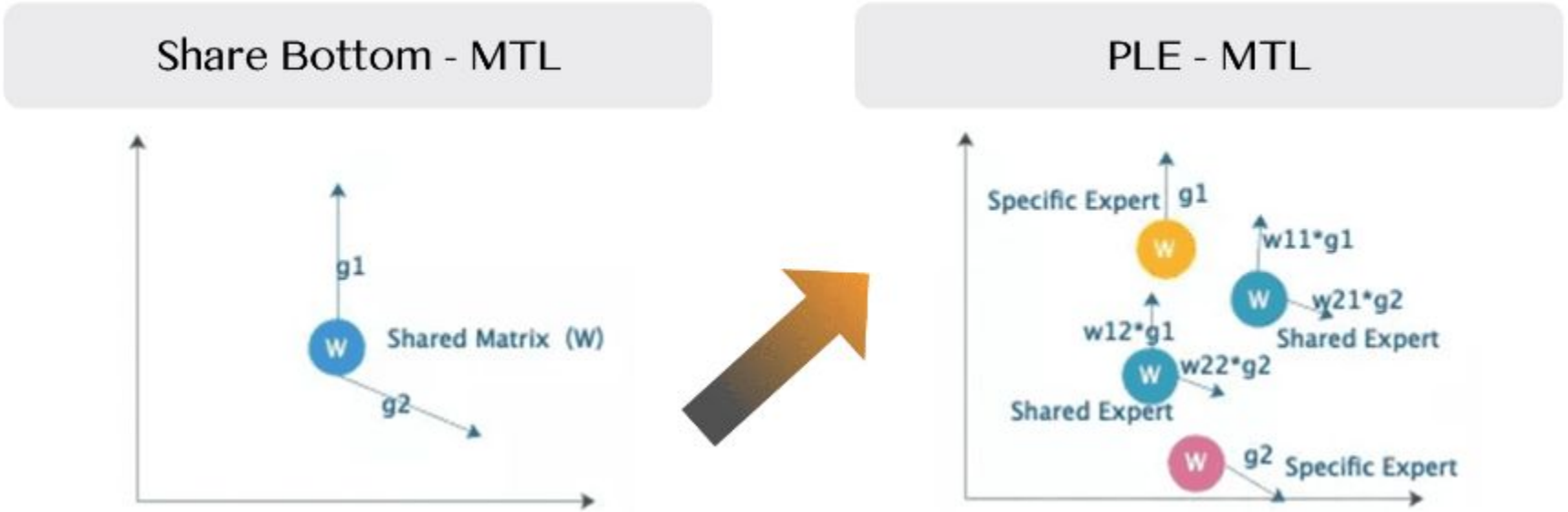


Figure 5: Progressive Layered Extraction (PLE) Model

- Shared Bottom MMoE：MMoE将shared bottom分解成多个Expert，然后**通过门控网络自动控制不同任务对这些Expert的梯度贡献。**
- MMoE PLE：PLE在MMoE的基础上又为每个任务增加了自有的Expert，仅由本任务对其梯度更新。更好的降低了相关性不强的任务之间的信息共享带来的问题。

改进方向-模型结构设计

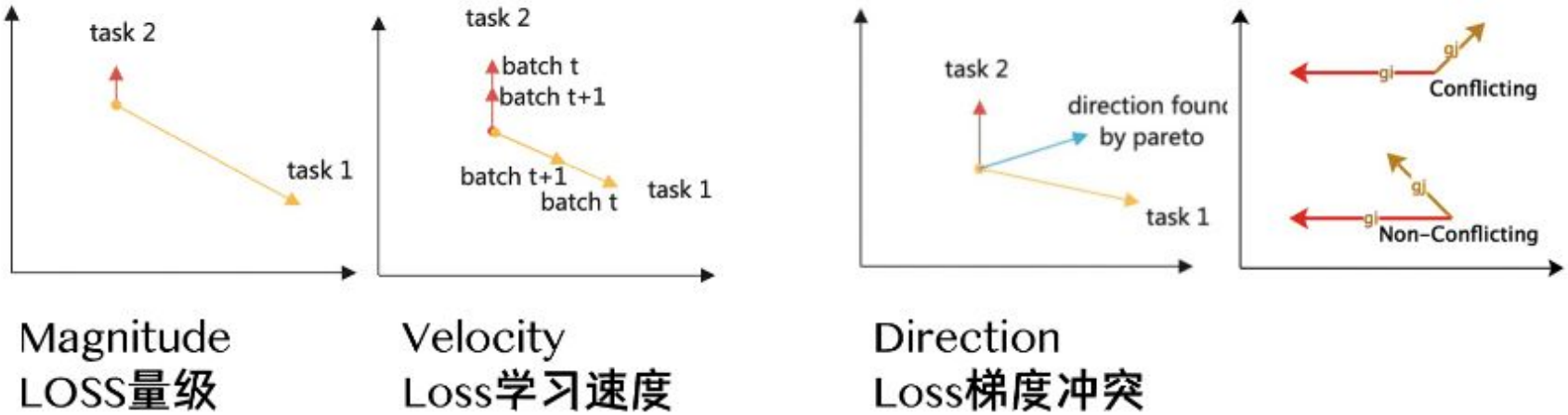


概率转移模式：ESMM->ESM2

专家底模式+概率转移模式：AITM *https://zhuanlan.zhihu.com/p/539987606*

改进方向-loss优化

- Magnitude（量级）：Loss 值有大有小，取值大的 Loss 可能会主导
- Velocity（学习速度）：不同任务因为样本的稀疏性、学习的难度不一致，在训练和优化过程中，存在 Loss 学习速度不一致的情况
- Direction（梯度冲突）：不同任务的 Loss 对共享参数进行更新，梯度存在不同的大小和方向，相同参数被多个梯度同时更新的时候，可能会出现冲突，导致相互消耗抵消，进而出现跷跷板、甚至负迁移现象



Magnitude
LOSS量级

Velocity
Loss学习速度

Direction
Loss梯度冲突

改进方向-loss优化

分类	方法名称【时间】	简介
Magnitude	Uncertainty Weight（2018）	自动学习任务的uncertainty，给uncertainty大的任务小权重、uncertainty小的任务大权重。
	GradNorm （2018）	综合任务梯度的二范数和loss下降速度，设计关于权重的损失函数Gradient Loss，并通过梯度下降更新权重。
Velocity	DWA （2019）	用loss下降速度来衡量任务的学习速度，直接得到任务的权重。
Direction	MGDA （Pareto）（2018）	提出一种基于Frank-Wolfe的优化方法，能适应大规模问题。并且为优化目标提供了一个上限，通过优化上限可以通过单次反向传播来更新梯度，而无需单独更新特定任务的梯度，减小了MGDA的计算开销。
	PE-LTR（ Pareto）（2019）	在pareto的基础上为每个任务引入优先级约束，并且提供了一个两步解法来求解新的优化目标。
	PCGrad（2020）	当两个梯度冲突时，直接把一个任务的梯度投影到另一个任务的法向量上以减轻梯度干扰。
	GradVac（2021）	利用任务相关性设置梯度的相似性目标，并且自适应地对齐任务梯度以实现这些目标。

改进方向-loss优化-量级

Uncertainty Weight

$$L_{\text{total}} = \sum_i w_i L_i$$

$$\mathcal{L}(W, \sigma_1, \sigma_2) \approx \frac{1}{2\sigma_1^2}\mathcal{L}_1(W) + \frac{1}{2\sigma_2^2}\mathcal{L}_2(W) + \log \sigma_1 + \log \sigma_2$$

总 Loss 设计成这样的形式，模型优化过程中会倾向于惩罚高 Loss 而低 的情况（如果一个任务的 Loss 高，同时 又σ_1的话，这一项就会很大，优化算法就会倾向于优化它）。

这样优化的结果就是往往 Loss 小（『相对简单』）的任务会有一个更大的权重。

缺点：loss之间可能差几十倍，小label很容易过拟合

改进方向-loss优化-量级&速度

GradNorm

Gradient normalization方法的主要思想是：

- 希望不同的任务的 Loss 量级是接近的
- 希望不同的任务以相似的速度学习

	物理意义	数学表达
单任务梯度大小	wi*Li对参数W的梯度L2范数	$G_i^W(t)$
多任务平均梯度大小	多任务的梯度平均值	$\bar{G}^W(t)$
任务学习速度	当前step与初始loss的比值来衡量	$\hat{L}_i(t) = L_i(t)/L_i(0)$
任务相对学习速度	多任务之间的学习速度相对值	$r_i(t) = \hat{L}_i(t)/E_{\text{task}}\left[\hat{L}_i(t)\right]$

gradient loss

$$L_{\text{grad}}(t; w_i(t)) = \sum_i \left| G_W^{(i)}(t) - \overline{G}_W(t) \times [r_i(t)]^{\alpha} \right|_1$$

改进方向-loss优化-速度

DWA

- 定义了一个指标来衡量任务学习的快慢，然后来指导调节任务的权重
- 用这一轮loss除以上一轮loss，这样可以得到这个任务loss的下降情况用来衡量任务的学习速度，然后直接进行归一化得到任务的权重。
- 当一个任务loss比其他任务下降的慢时，这个任务的权重就会增加，下降的快时权重就会减小。

$$w_k(t-1) = \frac{L_k(t-1)}{L_k(t-2)}$$

$$\lambda_k(t) = \frac{Kexp(w_k(t-1)/T)}{\sum_i exp(w_i(t-1)/T)}$$

改进方向-loss优化-方向

PE-LTR：帕累托优化

假设：拍累托有效是指处于一种任何一个目标都不可能在不伤害其他目标的前提下得到进一步的改进的状态。

$$\min_{\substack{\theta^{sh}\\ \theta^1,\dots,\theta^T}} L(\theta^{sh},\theta^1,\dots,\theta^T) = \min_{\substack{\theta^{sh}\\ \theta^1,\dots,\theta^T}} (\hat{\mathcal{L}}^1(\theta^{sh},\theta^1),\dots,\hat{\mathcal{L}}^T(\theta^{sh},\theta^T))^T$$

针对共享参数 θ^{sh} 和任务specific参数 θ^T 的KKT条件（是达到最优状态的必要条件 necessary for optimality）：

1. 存在$w_1,\dots,w_T \geq 0$，使得$\sum_{t=1}^T w_t = 1$且$\sum_{t=1}^T w_t \nabla_{\theta^{sh}} \hat{\mathcal{L}}^t(\theta^{sh},\theta^t) = 0$

2. 对所有任务的speicfic参数$\nabla_{\theta^t} \hat{\mathcal{L}}(\theta^{sh},\theta^t) = 0$

$$\begin{aligned} &min. || \sum_{i=1}^T w_i \nabla_\theta \mathcal{L}_i(\theta) ||_2^2 \\ &s.t.\ \sum_{i=1}^T w_i = 1, w_i \geq c_i, \forall i \in \{1,..,T\} \end{aligned}$$

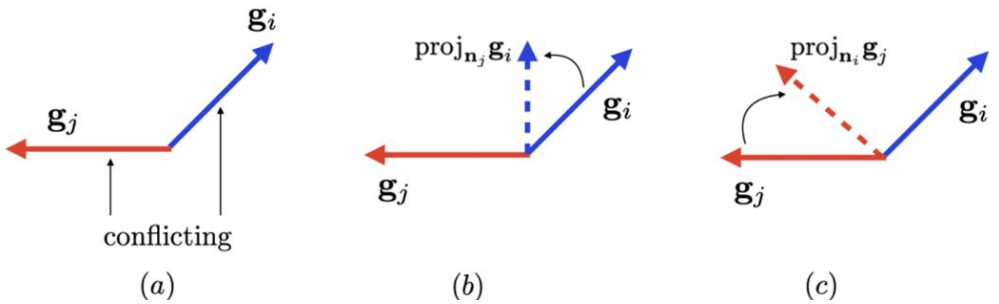
1. 初始化。定义多目标各自的loss（可以理解为每个目标常规的损失函数）；设置标量初始值，即多目标组合权重初始值w；以及权重边界c。
2. 使用权重向量，将多目标loss线性组合为单个loss。并开始batch loop
3. 根据组合后的单个loss，梯度回传，更新模型参数。（常规的模型更新）
4. 根据 **PECsolver** 更新组合权重w。（PECsolver，结合KKT条件求解复杂二次规划问题）。
5. 更新多目标组合共识，得到新的单个Loss
6. 依次训练，直到结束。（最终求解的是模型参数和融合权重）

《A Pareto-Efficient Algorithm for Multiple Objective Optimization in E-Commerce Recommendation》

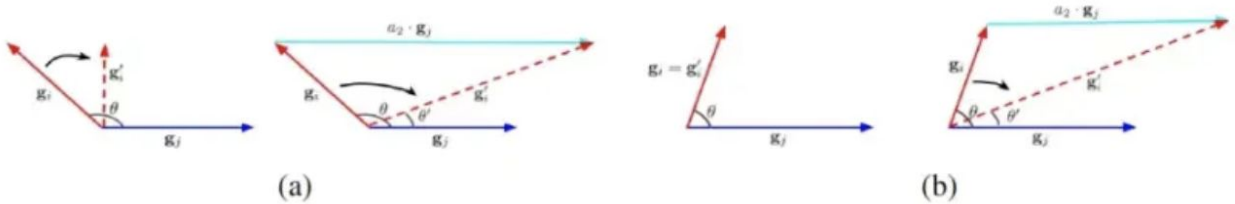
改进方向-loss优化-方向

PCGrad

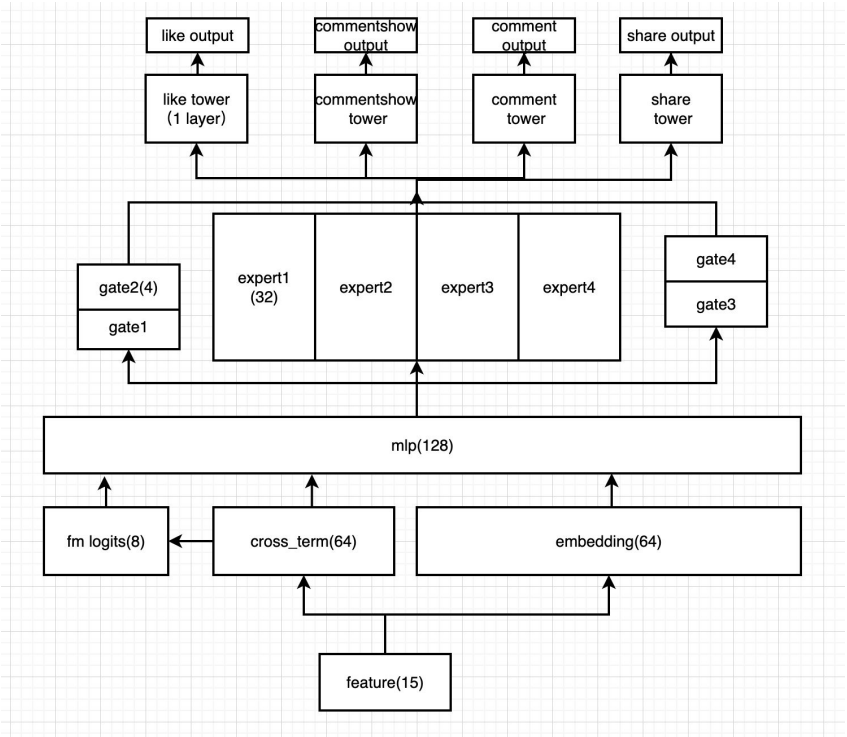
- 先检测不同任务的梯度是否冲突，冲突的标准就是是否有negative similarity；
- 如果有冲突，就把冲突的分量clip 掉（即，把其中一个任务的梯度投影到另一个任务梯度的正交方向上）。



GradVac



我们怎么做



建模目标: like、commentshow、share和comment

特征: 与原始特征一致，替 换成新版本特征，具体 slot为 ["1570", "1568", "2123", "2130", "2128", "2125", "2127", "1591", "1632", "1593", "1614", "2039", "2045", "2047", "1624"]

模型结构: mmoe结构

我们怎么做

模型在除like目标以外的其他目 标上，auc和uauc都有明显提升。其中 对于更加稀疏的目 标（comment，share）的uauc提升明显。

口径：使用2022-06-16最新的ckpt，预测2022-06-17至19，3天的数据。使用 预测结果离线评估。

	predict_uauc_old	predict_uauc_new	uauc_diff	predict_auc_old	predict_auc_new	auc_diff
like	59.84%	58.08%	-2.94%	85.88%	86.00%	0.14%
cc	62.40%	68.36%	9.55%	67.32%	78.97%	17.31%
comment	57.88%	70.42%	21.67%	86.00%	87.16%	1.35%
share	67.11%	72.58%	8.15%	75.98%	81.51%	7.28%

互动(单位‰)	平均(Action Per VV)	人均(Action Per User)	用户渗透率(User Penetration Rate)
like	-2.17% (148.6 -> 145.4, p=0.0%)	+4.55% (5727.1 -> 5987.8)	+4.59% (1487.0 -> 1555.2, p=0.0%)
share	+0.42% (5.8 -> 5.8, p=70.2%)	+6.69% (223.3 -> 238.3)	+8.27% (130.0 -> 140.8, p=0.0%)
follow	+16.37% (10.2 -> 11.8, p=0.0%)	+24.03% (391.6 -> 485.7)	+18.32% (180.4 -> 213.4, p=0.0%)
head	+14.07% (9.7 -> 11.1, p=0.0%)	+21.87% (375.4 -> 457.5)	+18.83% (277.1 -> 329.3, p=0.0%)
comment	+1.34% (8.0 -> 8.1, p=15.5%)	+8.61% (309.6 -> 336.3)	+7.43% (185.2 -> 199.0, p=0.0%)
comment_click	+6.63% (347.2 -> 370.3, p=0.0%)	+13.97% (13411.3 -> 15284.5)	+5.63% (2603.6 -> 2750.0, p=0.0%)

目前的问题&TODO

1、大量样本集中在高热和高作弊用户上（like、comment、dislike）

- 现象：超高行为用户和诱导互动视频 dislike作弊 like作弊
  - 10vv以上，点赞率50%用户(3800)，占点赞用户3.20%，like_cnt占比30%，vv占比0.50%，watchtime占比0.65%
  - 10vv以上，dislike_cnt>30且dislike率>0.5的user，约占每天dislike-user的(17/2444) 0.6%，dislike_cnt(1082/5514) 19.6%
  - 按照历史comment率倒排，comment_rate>0.002，vv占1.6%，like样本占2.5%，comment样本占23.1%
- TODO：去除作弊用户数据训练

2、金币/非金币用户差异巨大（实验/分析数据）

- 现象：在互动目标上，金币的uv渗透和pv_rate 都是非金币用户的2倍以上 分人群互动看板 新用户互动分析
  - 分人群调整参数：互动分人群调整实验
  - 分人群拆分模型：非金币用户的uauc和后验auc均有较大提升 只替换非金币模型
- TODO
  - 对金币用户降权reweight
  - PPNET分人群网络

3、模型的calibration 模型的calibration

- **现象：小label的calibration较差，对于新资源的calibration较差**
  - 去掉正样本加权对于模型auc/uauc影响不大
- TODO：
  -

day_diff	st_calibration	comment_calibraton	head_calibration	finish_calibration	like_calibration	follow_calibration	cc_calibration	share_calibration
7	1.00289596	1.249776199	1.014838477	1.005654965	0.7264687557	1.087643495	0.8360163016	0.821358779
15	1.003882864	1.344299275	1.035972599	1.012656572	0.8195692935	1.077766892	0.9513793358	0.9108594305
30	0.9962147572	1.42401849	1.096881942	1.013657428	0.791961621	1.038641192	0.9620849195	0.9101578112
60	0.9959811645	1.542045151	1.015533198	1.013000591	0.7941623505	1.009609776	0.9579647068	0.9448376008
%null%	1.004485378	1.650658302	0.9025337216	1.005928203	0.8419803472	1.02078393	1.110575888	1.069001558

4、特征利用不充分

- **现象：**目前的base模型使用的特征不足
  - AUTOINT+特征扩镇（143个，主要增加了互动的counter和序列特征）
- TODO
  - 交叉counter特征的开发和使用
    - 人群*video*counter：金币/非金币，核心/非核心，新用户/低活/老用户
    - l1/l2 tag*user*counter

参考文献

[1] Caruana, R. (1997). Multitask Learning. Machine Learning, 28(1), 41 - 75. doi: 10.1023/A:1007379606734

[2] Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning.Presented at the Proceedings of the 25th international conference ….

[3] Lounici, K., Pontil, M., Tsybakov, A. B., & van de Geer, S. (2009, March 8). Taking Advantage of Sparsity in Multi-Task Learning. arXiv.org.

[4] Zhang, Y., & Yeung, D.-Y. (2012, March 15). A Convex Formulation for Learning Task Relationships in Multi-Task Learning. arXiv.org.

[5] Zhou, J., Chen, J., & Ye, J. (2012) Multi-Task Learning , Theory, Algorithms, and Applications, SDM

THANKS !