多目标建模

MLE 算法指北

2025 年 5 月 16 日

1 引言

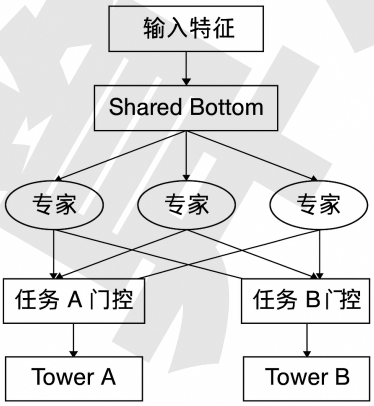
多目标建模（Multi-task Learning, MTL）旨在通过共享模型结构与知识，提升多个相关任务的学习效果。然而，由于任务之间差异性、梯度冲突等问题，如何高效组织网络结构并优化多目标的训练过程，是当前研究与工业实践的关键难点。

2 MMOE 结构

2.1 结构简介

MMOE（Multi-gate Mixture-of-Experts）是一种典型的多任务建模结构，其核心思想是多个共享的专家网络（Experts）+ 每个任务对应的门控网络（Gate）：

- 所有任务共享若干专家；
- 每个任务有自己的 Gate，通过 softmax 输出不同专家的加权组合；
- 最终输出传入任务专属的 Tower 网络，完成预测。



2.2 数学表达

给定输入特征 x：

$$
\begin{aligned}
h_i &= f_i(x), \quad i = 1, \ldots, E \\
g_t &= \text{softmax}(W_t x + b_t) \\
z_t &= \sum_{i=1}^{E} g_{t,i} \cdot h_i \\
y_t &= \text{Tower}_t(z_t)
\end{aligned}
$$

2.3 优点

- 支持任务间差异建模，避免负迁移；
- 门控机制动态选择专家，适应不同任务；
- 表达能力强，适合复杂任务组合。

2.4 多目标建模中的梯度冲突与解决方案

梯度冲突（Gradient Conflict）是多任务学习（MTL）中一个核心难点，指的是不同任务在共享参数上的优化方向存在显著差异甚至对抗，导致模型训练不稳定，甚至主任务性能下降。

2.4.1 定义与现象

在多任务学习中，若多个任务共享部分网络参数 θ，每个任务 t 对参数的梯度为 $\nabla_\theta L_t$。如果存在两个任务 i 和 j 满足：

$$\nabla_\theta L_i^\top \nabla_\theta L_j < 0,$$

则称这两个任务的梯度存在冲突。

典型表现包括：

- 某些任务 loss 持续下降缓慢或震荡；
- 主任务性能被非关键任务影响；
- 梯度范数不均、方向分散、训练不稳定；
- 多任务间存在负迁移。

2.4.2 解决梯度冲突的方法

1. **PCGrad（Projected Conflicting Gradient）**
   - **核心思想**：在多个任务之间检测是否存在冲突梯度，并将冲突部分进行正交投影去除。
   - **数学表达**：
   $$\text{If } \nabla_i^\top \nabla_j < 0, \quad \nabla_i \leftarrow \nabla_i - \frac{\nabla_i^\top \nabla_j}{\|\nabla_j\|^2} \cdot \nabla_j$$
   - **优点**：直接解决梯度方向冲突，不改变结构，易于实现；
   - **限制**：训练成本增加，对任务数量多时需两两比较。
2. **GradNorm**
   - **核心思想**：调节每个任务 loss 的权重 w_t，使得不同任务的训练速度保持一致。
   - **目标函数**：
   $$\mathcal{L}_{\text{gradnorm}} = \sum_t \left\| \|w_t \nabla_\theta L_t\| - r_t^\alpha \cdot \bar{G} \right\|_1$$
   其中 $r_t = \frac{L_t(t)}{L_t(0)}$ 表示任务相对进度，$\bar{G}$ 为梯度模长平均值。
   - **优点**：任务训练速率均衡，防止任务主导；
   - **缺点**：不能解决方向冲突，仅适用于训练速度失衡问题。
3. **GradVec / MGDA / CAGrad 等向量合成方法**
   - **核心思想**：将多个任务的梯度合并为一个方向，使整体更新方向最小冲突。
   - **优化目标**：
   $$\min_\alpha \left\| \sum_t \alpha_t \nabla L_t \right\|^2, \quad \text{s.t. } \alpha \in \mathcal{C}$$
   - **优点**：理论上最优合成方向；
   - **缺点**：计算成本高，依赖优化器求解子问题。
4. **DropGrad / GradDrop**
   - **核心思想**：随机地“丢弃”部分任务的梯度或梯度分量，以减少冲突区域；
   - **适合场景**：大任务集、噪声梯度多的任务场景。
5. **结构性解耦方法：PLE（Progressive Layered Extraction）**
   - **核心思想**：通过结构上显式区分任务特有与共享专家，物理隔离梯度来源；
   - **优势**：无需额外优化器或梯度修改，天然解决冲突；
   - **适用场景**：任务间相关性弱、负迁移显著场景。

方法	改梯度方向	调权重	改结构	适用场景
PCGrad	是	否	否	中小任务集，方向冲突突出
GradNorm	否	是	否	训练速度不平衡
GradVec/MGDA	是	是	否	高精度合成方向
DropGrad	是	否	否	大任务集，快速训练
PLE	否	否	是	结构级任务冲突

表 1: 梯度冲突解决方法对比表

2.4.3 方法对比总结

总结:
梯度冲突是多任务学习中的关键优化难题。不同方法从梯度方向修正、权重调整、结构解耦等角度出发，提供了互补性方案。实际使用中，应根据任务相关性、性能目标与工程约束选择最适合的策略。

2.5 MMOE 极化问题及解决策略

极化现象是指：在 MMOE 架构中，多个任务在训练过程中会偏好少数几个专家，导致其他专家长期几乎不被调用，训练不到有效表达能力。这种不均衡现象会导致模型表现下降、专家资源浪费和泛化能力不足。

典型解决方法包括：

- **熵正则 / KL 正则化**：鼓励 Gate 输出更加均匀。
  - 熵正则项（最大化 softmax 熵）：

$$\mathcal{L}_{\text{entropy}} = -\sum_{i=1}^{E} g_i(x) \log g_i(x)$$

  - KL 散度正则项（最小化与均匀分布差异）：

$$\mathcal{L}_{\text{KL}} = \sum_{i=1}^{E} g_i(x) \log \frac{g_i(x)}{1/E}$$

  - 优点：简单易实现；缺点：可能压制合理偏好。
- **负载均衡损失（Load Balance Loss）**：鼓励所有专家的平均使用频率趋近于一致。
  - 每个专家的使用频率：

$$p_i = \frac{1}{N} \sum_{j=1}^{N} g_{t,i}^{(j)}$$

  - 损失函数：

$$\mathcal{L}_{\text{balance}} = \sum_{i=1}^{E} \left(p_i - \frac{1}{E} \right)^2$$

  - 优点：直接调控专家使用率；缺点：需统计 Gate 输出均值，略有计算成本。
- **加噪探索机制（Gumbel Noise / Dropout）**：在训练初期加入扰动，鼓励 Gate 尝试更多专家。
  - Gumbel-softmax：

$$\tilde{z}_i = \frac{\exp((z_i + g_i)/\tau)}{\sum_j \exp((z_j + g_j)/\tau)}$$

  - 其中 $g_i \sim \text{Gumbel}(0, 1)$，$\tau$ 是温度；
  - 优点：促进 early-stage 探索；缺点：参数敏感，稳定性依赖退火策略。
- **Top-k Masking / 异构专家设计**：
  - Top-k Masking：每次仅选择前 k 个专家参与 softmax；
  - 异构专家：设计结构差异大的专家（如层数、宽度不同）增强多样性；
  - 优点：强制均衡、表达能力更强；缺点：实现复杂度更高。

对比总结如下：

方法	是否修改结构	是否修改损失	可控性	提升专家利用率
熵/KL 正则	否	是	中等	✓
负载均衡损失	否	是	强	✓✓
Gumbel / Dropout	否	否	中等	✓
Top-k / 异构专家	是	否	强	✓✓✓

表 2: MMOE 极化解决策略对比

2.6 MMOE 专家数量选择策略

专家数量是 MMOE 架构中影响模型容量与多任务性能的关键超参数。合理选择专家数量可以提高共享效率、缓解极化、增强模型表达能力。

设计原则如下：

- **基本下限**：专家数应满足 $E \geq T + k$，其中 T 为任务数，k 为冗余量（建议 $k \in [1, 5]$）；
- **任务差异性大** → 需要更多专家以支持任务特化；
- **数据量或计算资源有限** → 控制专家数（如 4 ~ 6）防止过拟合；
- **推理效率敏感场景** → 建议不超过 8 个专家。

任务数量与推荐专家数量对应表：

任务数量	推荐专家数	应用场景示例
2	4 ~ 6	点击率 + 转化率
3 ~ 4	6 ~ 8	推荐 + 停留 + 负反馈
5	8 ~ 16	多目标广告系统、大规模内容平台

表 3: 任务数与专家数推荐关系

动态调整建议：

- **Gate 热图**：观察专家使用频率是否集中于少数；
- **梯度范数**：监控每个专家是否获得有效梯度更新；
- **对比实验**：评估主任务指标（AUC、logloss）随专家数变化趋势。

最终建议：专家数应基于任务相关性、资源情况与训练稳定性进行综合权衡，而不是越多越好。

3 PLE 模型结构

PLE（Progressive Layered Extraction）是一种用于多任务建模的深层专家结构。其设计核心是：**逐层提取共享与特化表示**，明确区分任务公共表达与任务私有特征。

3.1 结构组成

每一层包含以下三类组件：

- **共享专家（Shared Experts）**：供所有任务访问；
- **任务专属专家（Task-specific Experts）**：仅被对应任务选择；
- **任务门控（Gate）**：每个任务独立拥有 gate，从共享 + 自己专家中加权选择；

上述结构可堆叠多层，构成深层表达。

3.2 数学表达

以任务 A 的第 l 层为例：

$$g_A^l = \text{softmax}(W_A^l x + b_A^l)$$

$$z_A^l = \sum_{i=1}^{E_s+E_a} g_{A,i}^l \cdot h_i^l$$

输出 z_A^l 作为输入送入下一层，或直接接入任务塔网络。

3.3 优势对比

与 MMOE 相比，PLE 拥有更清晰的结构分离、更强的表达能力和更灵活的专家建模方式，尤其适合任务异质性较强的场景。

特性	MMOE	PLE
专家区分	无共享/私有区分	显式共享 + 专属
是否多层	否（一般单层）	可堆叠多层
任务门控来源	所有专家	自己专属 + 共享专家
表达能力	中等	强
适合任务类型	相对相关任务	强相关 + 弱相关任务混合

表 4: MMOE 与 PLE 架构对比

4 多目标优化策略

多任务学习中的主要挑战之一是任务之间的梯度冲突或不均衡更新。下列优化策略从不同角度解决该问题：

4.1 1. Magnitude Balancing

目标：使不同任务的梯度模长一致。

4.2 2. Velocity Balancing

目标：保持各任务 loss 的下降速度一致。

4.3 3. GradNorm

核心：通过比较每个任务当前 loss 与初始 loss 的比例调整其权重：

$$L_{\text{gradnorm}} = \sum_t \left\| \|w_t \nabla_\theta L_t\| - r_t^\alpha \cdot \bar{G} \right\|_1$$

4.4 4. PCGrad

核心思想：将任务间冲突的梯度投影到非冲突方向：

$$g_i \leftarrow g_i - \frac{g_i^\top g_j}{\|g_j\|^2} g_j, \quad \text{if } g_i^\top g_j < 0$$

4.5 5. GradVec

目标：从所有任务梯度中求解整体最优合成方向：

$$\min_\alpha \left\| \sum_t \alpha_t \nabla L_t \right\|^2, \quad \text{s.t. } \alpha \in \mathcal{C}$$

4.6 策略适用建议

- 任务相关性强：使用 GradNorm、Velocity；
- 任务冲突严重：使用 PCGrad、GradVec；
- 高效部署场景：推荐 GradNorm + Top-k 限制；

5 总结

多目标建模的核心在于同时建模共享与差异信息结构（如 MMOE、PLE），并采用动态优化策略（如 GradNorm、PCGrad）平衡任务冲突与训练速度。设计一个高性能多目标系统需在建模结构与优化策略之间共同发力。