帕累托多任务学习

摘要

多任务学习是同时解决多个相关任务的有效方法。然而，通常不可能找到一个单一的解决方案来优化所有任务，因为不同的任务可能相互冲突。近年来，提出了一种新的方法，将多任务学习转化为多目标优化，在不同任务之间寻找一个具有良好权衡的帕累托最优解。本文推广了这一思想，并提出了一种新的帕累托多任务学习算法 (Pareto MTL) 来寻找一组分布良好的帕累托解，这些解可以表示不同任务之间的不同权衡。该算法首先将多任务学习问题表述为多目标优化问题，然后将多目标优化问题分解为一组具有不同权衡偏好的约束子问题。通过并行求解这些子问题，Pareto MTL 可以找到一组具有较好代表性的帕累托最优解，这些最优解在所有任务之间具有不同的权衡。从业者可以很容易地从这些帕累托解决方案中选择他们喜欢的解决方案，或者在不同的情况下使用不同的权衡解决方案。实验结果表明，该算法在许多多任务学习应用中可以生成具有良好代表性的解。

关键词： 多任务学习；多目标优化；帕累托最优

1 引言

多任务学习 (Multi-task learning, MTL) [1] 是机器学习界的一个热门研究课题，旨在同时学习多个相关的任务。通过一起解决多个相关任务，MTL 可以进一步提高每个任务的性能，并减少在许多实际应用程序中执行所有任务的推理时间。过去已经提出了许多 MTL 方法，它们在计算机视觉 [2]、自然语言处理 [3] 和语音识别 [4] 等许多领域都取得了很好的表现。

大多数 MTL 方法都是为了寻找一个单一的解决方案来提高所有任务的整体性能 [5, 6]。然而，在许多应用程序中，可以观察到一些任务可能相互冲突，并且没有一个最优的解决方案可以同时优化所有任务的性能 [7]。在现实应用中，MTL 从业者必须在不同的任务之间做出权衡，例如自动驾驶汽车 [8]、人工智能辅助 [9] 和网络架构搜索 [10, 11]。

如何将不同的任务结合在一起，并在它们之间做出适当的权衡是一个难题。在许多 MTL 应用中，特别是那些使用深度多任务神经网络的应用中，所有任务首先通过线性加权标量化合并为单个代理任务，然后为这些任务分配一组能够反映偏好的固定权重，最后对单个代理任务进行优化。然后，为不同的任务设置合适的权值并不容易，通常需要穷举权值进行搜索。事实上，如果一些任务相互冲突，没有一个解决方案可以同时在所有任务上实现最佳性能。

最近，Sener 和 Koltun [12] 以一种新颖的方式将多任务学习问题表述为多目标优化问题。他们提出了一种有效的算法，在不同的任务中找到一个帕累托 (Pareto) 最优解。然而，MTL 问题在其任务之间可能有许多 (甚至无限数量) 最优权衡，并且通过该方法获得的单个解决方案可能并不总是满足实际需求。

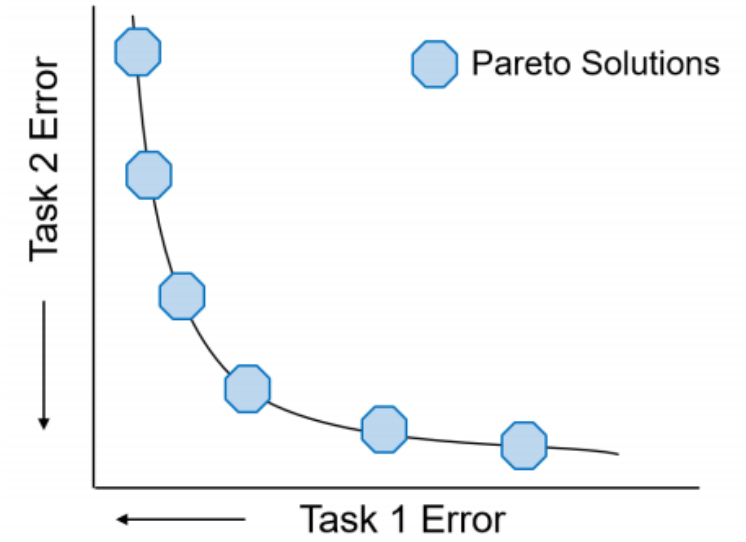


图 1. Pareto MTL 可以找到一组广泛分布的 Pareto 解决方案。

本文推广了多目标优化思想 [12]，并提出了一种新的 Pareto 多任务学习 (Pareto Multi-Task Learning, Pareto MTL) 算法，为给定的 MTL 问题生成一组具有良好代表性的 Pareto 解。如图 1所示，MTL 实践者可以很容易地从具有不同权衡的获得的 Pareto 最优解集中选择他们的首选解，而不是穷尽地为所有任务寻找一组适当的权重。

2 相关工作

多任务学习 (MTL) 算法旨在通过同时学习多个相关任务来提高性能。这些算法通常构造共享参数表示来组合多个任务。它们已被应用于许多机器学习领域。然而，大多数 MTL 算法主要侧重于构建共享表示，而不是在多个任务之间进行权衡 [5,6]。

2.1 传统多任务学习

当 MTL 从业者想要获得一组不同的权衡解决方案时，线性标量化以及网格搜索或权重向量的随机搜索是当前的默认做法。这种方法很简单，但效率极低。最近的一些研究 [7,13] 表明，一种设计良好的权重自适应算法的单次运行可以超过 100 次运行的随机搜索方法。这些自适应权重方法侧重于在优化过程中平衡所有任务，不适合寻找不同的权衡解。

2.2 多目标优化

多目标优化 [14] 旨在寻找具有不同权衡的一组 Pareto 解，而不是单个解。它已被用于许多机器学习应用，如强化学习 [15]、贝叶斯优化 [16–18] 和神经结构搜索 [10, 19]。在这些应用中，梯度信息通常是不可用的。基于种群和无梯度的多目标进化算法 [20,21] 是在单次运行中找到一组分布良好的 Pareto 解的常用方法。然而，它不能用于解决大规模和基于梯度的 MTL 问题。

2.3 多目标梯度下降

多目标梯度下降 [22–24] 是一种有效的多目标优化方法，当梯度信息可用时。Sener 和 Koltun [12] 提出了一种求解 MTL 的新方法，将其视为多目标优化。然而，与自适应权重方法类似，该方法在优化过程中试图平衡不同的任务，没有系统地纳入权衡偏好。在本文中，本文将其推广到寻找 MTL 问题中具有不同任务权衡的一组具有良好代表性的 Pareto 解。

2.4 多任务学习作为多目标优化

2.4.1 MTL 作为多目标优化

MTL 问题涉及一组 m 个具有损失向量的相关任务：

$$\min_\theta \mathcal{L}(\theta) = (\mathcal{L}_1(\theta), \mathcal{L}_2(\theta), \cdots, \mathcal{L}_m(\theta))^{\mathrm{T}}, \tag{1}$$

(1) 式中，$\mathcal{L}_i(\theta)$ 为第 i 个任务的损失。MTL 算法是利用所有任务之间的共享结构和信息来同时优化所有任务。问题 (1) 是一个多目标优化问题。没有一个解决方案可以同时优化所有目标。本文能得到的是一组所谓的 Pareto 最优解，它在所有目标之间提供了不同的最优权衡。在本文中，本文的重点是寻找一组很有代表性的 Pareto 解，可以近似 Pareto 前沿。

2.4.2 线性标量化

线性标量化是解决多任务学习问题最常用的方法。该方法采用线性加权和方法，将所有任务的损失合并为单个代理损失：

$$\min_\theta \mathcal{L}(\theta) = \sum_{i=1}^{m} \boldsymbol{w}_i \mathcal{L}_i(\theta), \tag{2}$$

(2) 式中，ω_i 为第 i 个任务的权值。这种方法简单直接，但从多任务学习和多目标优化的角度来看，它都存在一些缺点。在典型的多任务学习应用程序中，在优化之前需要手动分配权重 ω_i，并且整体性能高度依赖于分配的权重。选择一个合适的权重向量可能是非常困难的，即使是对给定问题具有专业知识的经验丰富的 MTL 从业者。

2.4.3 基于梯度的多目标优化方法

目前已经提出了许多基于梯度的方法来解决多目标优化问题 [22,23]。Fliege 和 Svaiter [24] 提出了一种简单的基于梯度的方法，该方法是对单目标最陡下降算法的推广。算法的更新规则为 $\theta_{t+1} = \theta_t + \eta d_t$，其中 η 为步长，搜索方向 d_t 可以通过以下式子求解：

$$(d_t, \alpha_t) = \arg \min_{d \in R^n, \alpha \in R} \alpha + \frac{1}{2}||d||^2, s.t. \ \ \nabla \mathcal{L}_i(\theta_t)^T d \leq \alpha, i = 1, ..., m. \tag{3}$$

上述问题的解满足：

引理 1 [24]：(d^k,α^k) 为问题 (3) 的解。

1. 如果 θ_t 是 Pareto 临界，那么 d_t=0 且 α_t=0。
2. 若 θ_t 非 Pareto 临界，则

$$\begin{aligned} &\alpha_t \leq -(1/2)||d_t||^2 < 0, \\ &\nabla \mathcal{L}_i(\theta_t)^T d_t \leq \alpha_t, i = 1, ..., m, \end{aligned} \tag{4}$$

当 θ 的邻域内没有其他解在所有目标函数中都有更好的值时，θ 称为 Pareto 临界。也就是说，如果 $d_t = 0$，没有一个方向可以同时提高所有任务的性能。如果想提高某一任务的性能，另一任务的性能就会下降，因此当前解为 Pareto 临界点。当 $d_t \neq 0$ 时，它是所有任务的有效下降方向。当前解应沿着得到的方向 $\theta_{t+1} = \theta_t + \eta d_t$ 进行更新。

最近，Sener 和 Koltun [12] 使用多重梯度下降算法 (multiple gradient descent algorithm, MGDA) [22] 求解 MTL 问题，并取得了令人满意的结果。然而，这种方法并没有一个系统的方法来整合不同的权衡偏好信息。本文对该方法进行了推广，并提出了一种新的 Pareto MTL 算法来寻找所有任务之间具有不同权衡的分布良好的 Pareto 解集。

3 本文方法

3.1 MTL 分解

Pareto MTL 的主要思想是将 MTL 问题分解为若干具有不同权衡偏好的约束多目标子问题。通过并行地解决这些子问题，MTL 实践者可以获得一组具有不同权衡的很有代表性的解决方案。

基于分解的多目标进化算法 [25,26] 将多目标优化问题分解为若干子问题并同时求解，是目前最流行的无梯度多目标优化方法之一。本文提出的 Pareto MTL 算法推广了求解大规模和基于梯度的 MTL 的分解思想。

本文采用 [27] 中的思想，将 MTL 分解为 K 个子问题，这些子问题具有一组分布良好的单位偏好向量 $\{u_1, u_2, ..., u_K\}$。假设 MOP 中的所有目标都是非负的，则偏好向量 u_k 对应的多目标子问题为:

$$\min_{\theta} \mathcal{L}(\theta) = (\mathcal{L}_1(\theta), \mathcal{L}_2(\theta), \cdots, \mathcal{L}_m(\theta))^{\mathrm{T}}, s.t. \quad \mathcal{L}(\theta) \in \Omega_k, \tag{5}$$

式中 Ω_k(k = 1, ..., K) 为目标空间中的子区域:

$$\Omega_k = \{\boldsymbol{v} \in R_+^m | \boldsymbol{u}_j^T \boldsymbol{v} \leq \boldsymbol{u}_k^T \boldsymbol{v}, \forall j = 1, ..., K\} \tag{6}$$

$u_j^T v$ 是偏好向量 u_j 和给定向量 v 之间的内积，也就是说，v $\in \Omega_k$ 当且仅当 v 与 u_k 的锐角最小。因此在所有 K 个偏好向量中，$u_k^T v$ 是最大的内积。

子问题 (5) 可进一步表述为:

$$\begin{aligned} &\min_{\theta} \mathcal{L}(\theta) = (\mathcal{L}_1(\theta), \mathcal{L}_2(\theta), \cdots, \mathcal{L}_m(\theta))^{\mathrm{T}} \\ &s.t. \quad \mathcal{G}_j(\theta_t) = (\boldsymbol{u}_j - \boldsymbol{u}_k)^T \mathcal{L}(\theta_t) \leq 0, \forall j = 1, ..., K, \end{aligned} \tag{7}$$

如图 2所示，偏好向量将目标空间划分为不同的子区域。每个子问题的解将被相应的偏好向量吸引，从而被引导到其代表性的子区域。所有子问题的解决方案集将位于不同的子区域，并表示任务之间的不同权衡。

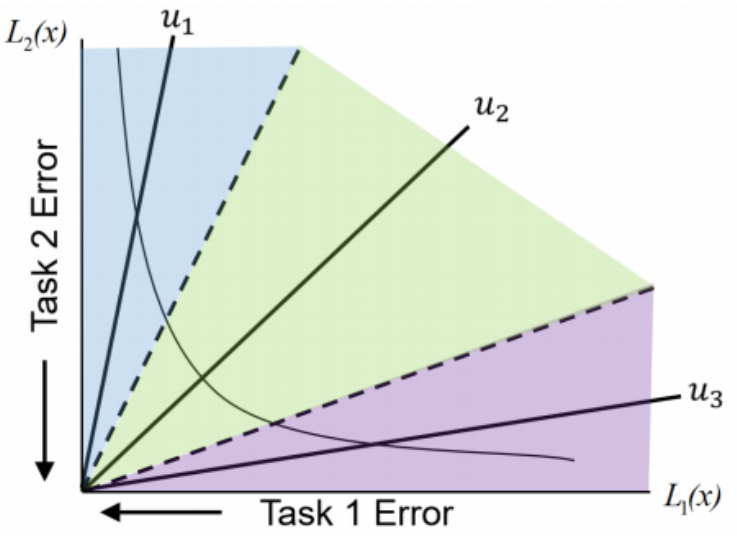


图 2. Pareto MTL 将给定的 MTL 问题分解为具有一组偏好向量的子问题。

3.1.1 寻找初始解

为了用基于梯度的方法求解约束多目标子问题 (5)，本文需要找到一个可行的或至少满足大多数约束的初始解。对于随机生成的解 θ_r，一种直接的方法是找到一个可行的初始解 θ_0，它满足:

$$\min_{\theta_0} ||\theta_0 - \theta_r||^2 \quad s.t. \quad \mathcal{L}(\theta_0) \in \Omega_k. \tag{8}$$

然而，这种方法是一个 n 维约束优化问题 [28]。直接解决这个问题是低效的，特别是对于具有数百万个参数的深度神经网络。在提出的 Pareto MTL 算法中，本文将该问题重新表述为无约束优化，并使用基于序列梯度的方法来寻找初始解 θ_0。

对于一个解 θ_r，本文定义所有激活约束的集合为 $I(\theta_r) = \{j|\mathcal{G}_j(\theta_r) \geq 0, \text{j=1, ..., K}\}$。可以通过求解问题 (9) 找到一个有效的下降方向 dr 来减小所有激活约束的值:

$$(d_r, \alpha_r) = \arg \min_{d \in R^n, \alpha \in R} \alpha + \frac{1}{2}||d||^2, s.t. \nabla \mathcal{G}_j(\theta_r)^T d \leq \alpha, j \in I(\theta_r). \tag{9}$$

该方法类似于基于无约束梯度的方法 (3)，但它减少了所有激活约束的值。基于梯度的更新规则为 $\theta_{r_t+1} = \theta_{r_t} + \eta d_t$，一旦找到可行的解决方案或满足预定义的迭代次数，更新将停止。

3.1.2 求解子问题

一旦有了一个初始解，就可以使用基于梯度的方法来解决约束子问题。根据 [24, 28]，我们可以通过求解类似于无约束情况下的子问题 (3) 的子问题来找到这个受约束的 MOP 的下降方向:

$$\begin{aligned} (d_t, \alpha_t) = &\arg \min_{d \in R^n, \alpha \in R} \alpha + \frac{1}{2}||d||^2 \\ &s.t. \quad \nabla \mathcal{L}_i(\theta_t)^T d \leq \alpha, i = 1, ..., m. \\ &\nabla \mathcal{G}_j(\theta_t)^T d \leq \alpha, j \in I_\epsilon(\theta_t), \end{aligned} \tag{10}$$

其中 $I_\epsilon(\theta)$ 为所有激活约束的索引集:

$$I_\epsilon(\theta) = \{j \in I | \mathcal{G}_j(\theta) \geq -\epsilon\}. \tag{11}$$

此处增加了一个小阈值 ϵ 来处理约束边界附近的解。与无约束情况类似，对于可行解 θ_t，通过求解问题 (10)，要么得到 $d_t = 0$ 并确认 θ_t 是限制在 Ω_k 上的 Pareto 临界点，要么得到 $d_t \neq 0$ 作为约束多目标问题 (7) 的下降方向。在后一种情况下，如果所有约束都不激活，d_t 是所有任务的有效下降方向。否则，d_t 是减少所有任务和所有激活约束值的有效方向。
引理 2 [28]：(d^k,α^k) 为问题 (10) 的解。

1. 如果 θ_t 是限制在 Ω_k 上的 Pareto 临界，则 d_t=0 且 α_t=0。
2. 若 θ_t 不约束于 Ω_k 上的 Pareto 临界，则

$$\begin{aligned} &\alpha_t \leq -(1/2)||d_t||^2 < 0, \\ &\nabla \mathcal{L}_i(\theta_t)^T d_t \leq \alpha_t, i = 1, ..., m \\ &\nabla \mathcal{G}_j(\theta_t)^T d_t \leq \alpha_t, j \in I_\epsilon(\theta_t). \end{aligned} \tag{12}$$

因此，可以用简单的基于迭代梯度的更新规则 $\theta_{t+1} = \theta_t + \eta_r d_t$ 得到每个子问题的受限 Pareto 临界解。通过求解所有子问题，可以得到一组限制在不同子区域上的不同 Pareto 临界解，这些解可以表示原始 MTL 问题的所有任务之间的不同权衡。

3.1.3 可扩展的优化方法

通过求解约束优化问题 (10)，可以得到每个多目标约束子问题的有效下降方向。然而，在高维决策空间中，优化问题本身并不能很好地扩展。例如，在训练深度神经网络时，通常有超过数百万个参数需要优化，在这个规模下解决约束优化问题 (10) 将会非常缓慢。在本节中，提出了一种可扩展的优化方法来解决约束优化问题。受 [24] 的启发，首先将优化问题 (10) 改写为对偶形式。根据 KKT 条件，有

$$d_t = -(\sum_{i=1}^{m} \lambda_i \nabla \mathcal{L}_i(\theta_t) + \sum_{j \in I_{\epsilon(\boldsymbol{x})}} \beta_i \nabla \mathcal{G}_j(\theta_t)), \quad \sum_{i=1}^{m} \lambda_i + \sum_{j \in I_{\epsilon(\theta)}} \beta_j = 1, \tag{13}$$

其中 $\lambda_i \geq 0$ 和 $\beta_i \geq 0$ 为线性不等式约束的朗日乘子。因此，对偶问题为:

$$\begin{aligned} &\max_{\lambda_i, \beta_j} \ -\frac{1}{2}||\sum_{i=1}^{m} \lambda_i \nabla \mathcal{L}_i(\theta_t) + \sum_{j \in I_{\epsilon(\boldsymbol{x})}} \beta_i \nabla \mathcal{G}_j(\theta_t)||^2 \\ &s.t. \quad \sum_{i=1}^{m} \lambda_i + \sum_{j \in I_{\epsilon(\theta)}} \beta_j = 1, \lambda_i \geq 0, \beta_j \geq 0, \forall i = 1, ..., m, \forall j \in I_\epsilon(\theta). \end{aligned} \tag{14}$$

对于上述问题，决策空间不再是参数空间，而成为目标约束空间。对于具有 2 个目标函数和 5 个激活约束的多目标优化问题，问题 (14) 的维数为 7，明显小于问题 (10) 可能超过一百万的维数。

Pareto MTL 算法框架如算法 1 所示。在优化过程中，由于子问题之间没有通信，所以所有子问题都可以并行解决。每个子问题的唯一偏好信息是偏好向量集。在没有任何 MTL 问题的先验知识的情况下，一组均匀分布的单位偏好向量将是一个合理的默认选择。

Algorithm 1 Pareto MTL Algorithm

Input: A set of evenly distributed vectors $\{\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_K\}$
Update Rule:
(can be solved in parallel)
for $k = 1$ to K **do**
  randomly generate parameters $\theta_r^{(k)}$
  find the initial parameters $\theta_0^{(k)}$ from $\theta_r^{(k)}$ using gradient-based method
  for $t = 1$ to T **do**
    obtain $\lambda_{ti}^{(k)} \geq 0, \beta_{ti}^{(k)} \geq 0, \forall i = 1, ..., m, \forall j \in I_\epsilon(\theta)$ by solving subproblem (14)
    calculate the direction $d_t^{(k)} = -(\sum_{i=1}^m \lambda_{ti}^{(k)} \nabla \mathcal{L}_i(\theta_t^{(k)}) + \sum_{j \in I_{\epsilon(\boldsymbol{\alpha})}} \beta_{ti}^{(k)} \nabla \mathcal{G}_j(\theta_t^{(k)})$
    update the parameters $\theta_{t+1}^{(k)} = \theta_t^{(k)} + \eta d_t^{(k)}$
  end for
end for
Output: The set of solutions for all subproblems with different trade-offs $\{\theta_T^{(k)} | k = 1, \cdot, K\}$

4 复现细节

4.1 与已有开源代码对比

本文的开源代码地址：`https://github.com/Xi-L/ParetoMTL`。本人复现了原文大部分实验，并在 Multi-Fashion-MNIST 实验中，增加了 ResNet18 网络进行对比，根据实验结果可以看出，得到了相对更好的结果。

4.2 实验环境搭建

anaconda3
python3.9
torch 2.0.0+cu118
torchaudio 2.0.1+cu118
torchvision 0.15.1+cu118

4.3 创新点

本文所提出的 Pareto 多任务学习算法 (Pareto MTL) 首先将多任务学习问题表述为多目标优化问题，然后将多目标优化问题分解为一组具有不同权衡偏好的约束子问题。通过并行求解这些子问题，Pareto MTL 可以找到一组具有较好代表性的 Pareto 最优解，这些最优解在所有任务之间具有不同的权衡。

5 实验结果分析

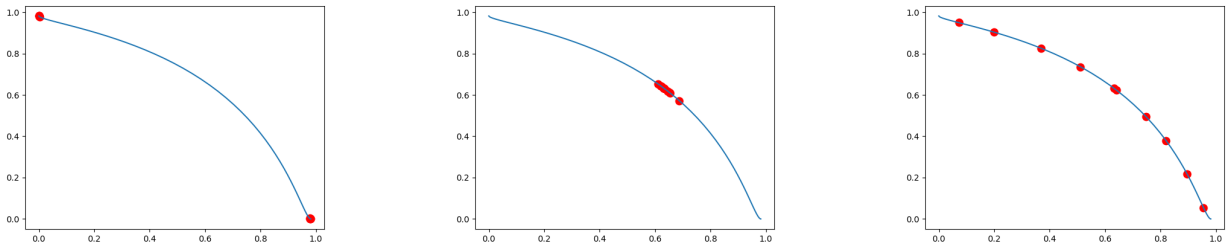
5.1 合成问题

为了更好地分析所提出的 Pareto MTL 的收敛性，首先在一个简单的合成多目标优化问题上将其与两种常用的方法进行了比较，即线性标量化方法和使用了多重梯度下降方法的 MOO-MTL 算法 [12]：

$$\min_{\boldsymbol{x}} f_1(\boldsymbol{x}) = 1 - \exp\left(-\sum_{i=1}^{d}(\boldsymbol{x}_d - \frac{1}{\sqrt{d}})^2\right)$$

$$\min_{\boldsymbol{x}} f_2(\boldsymbol{x}) = 1 - \exp\left(-\sum_{i=1}^{d}(\boldsymbol{x}_d + \frac{1}{\sqrt{d}})^2\right)$$

具体的实验结果如图 3所示，观察实验结果可以得到，所提出的 Pareto MTL 成功地生成了一组具有不同权衡的广泛分布的 Pareto 解。



(a) 随机线性标量化　(b) MOO MTL　(c) Pareto MTL (Ours)

图 3. 不同算法在一个综合算例上的收敛行为。(a) 运行 100 次后得到的随机线性标量化解。(b) 运行 10 次后得到的 MOO MTL [12] 方法解。(c) 本文提出的 Pareto MTL 方法经过 10 次运行得到的解。

5.2 Multi-Fashion-MNIST

为了评估 Pareto MTL 在具有不同任务关系的多任务学习问题上的性能，在 MultiMNIST [29] 和两个类似 MultiMNIST 的数据集上进行了实验。

为了构建 MultiMNIST 数据集，首先从原始 MNIST 数据集 [30] 中随机选取两幅不同数字的图像，然后将这两幅图像组合成一个新的图像，将一个数字放在左上角，另一个数字放在右下角。使用同样的方法，可以构建一个具有重叠 FashionMNIST 图像的 MultiFashionMINST 数据集 [31]，以及一个具有重叠 MNIST 和 FashionMNIST 图像的 Multi-(Fashion + MNIST) 数据集。

对于每个数据集，有一个两目标 MTL 问题，即对左上角的图像进行分类（任务 1）和对右下角的物品进行分类（任务 2）。实验中，分别构建了一个基于 LeNet [30] 以基于 ResNet18 [32] 的 MTL 神经网络，在三个数据集上进行训练，具体的结果如图 4所示。

根据实验结果可以看出，在所有的实验中，Pareto MTL 算法可以为所有实验生成多个分布良好的 Pareto 解，这些解在任务之间的权衡不同。对比 GradNorm 算法，Pareto MTL 算法的综合性能更好。这些结果证实了 Pareto MTL 算法可以成功地为 MTL 问题提供一组很有代表性的 Pareto 解。

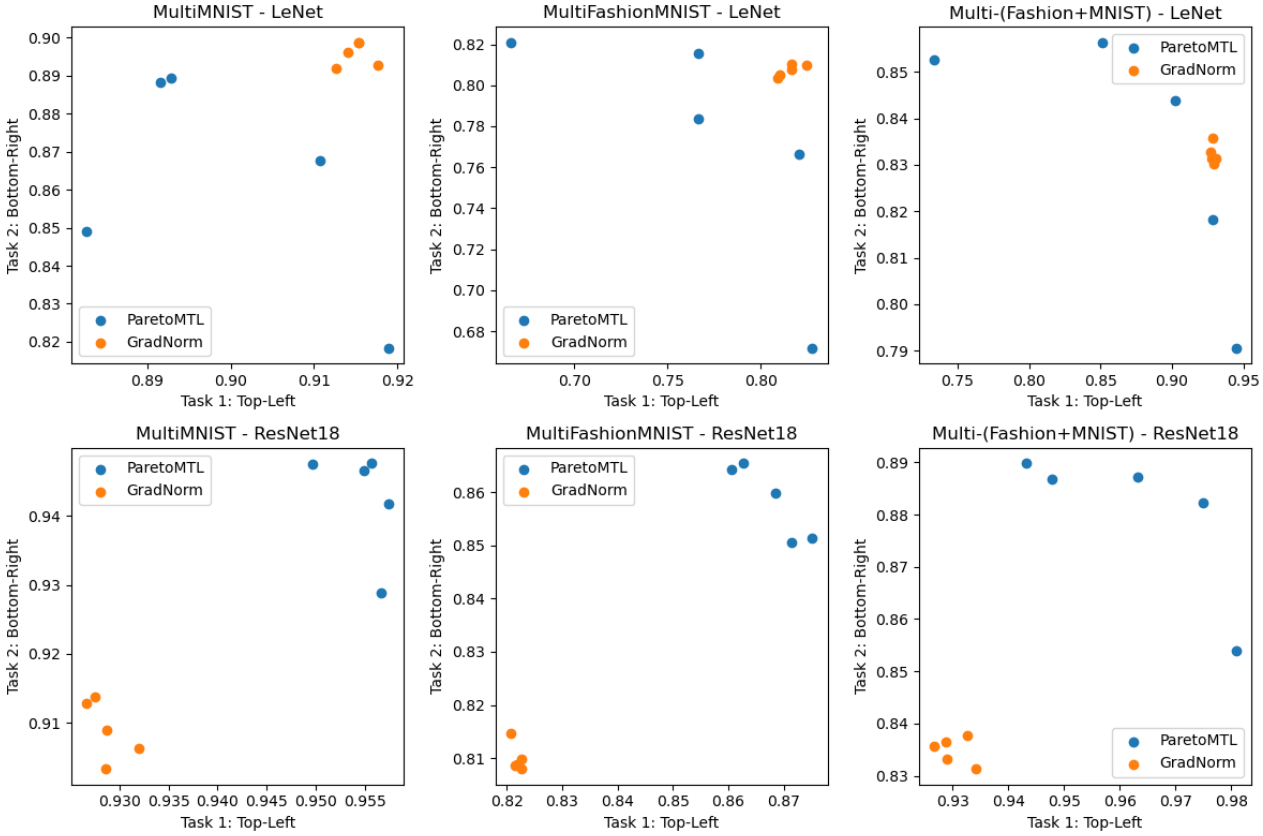


图 4. 任务 1 和任务 2 的准确率结果

6 总结与展望

本文提出的一种新的 Pareto 多任务学习 (Pareto Multi-Task Learning, Pareto MTL) 算法，针对给定的多任务学习 (Multi-Task Learning, MTL) 问题，生成一组具有不同任务间权衡的分布良好的 Pareto 解。然后，MTL 从业者可以很容易地从这些 Pareto 解决方案中选择他们喜欢的解决方案。实验结果证实，本算法可以成功地为不同的 MTL 应用找到一组很有代表性的解。

Pareto MTL 采用具有简单硬参数共享结构的神经网络作为 MTL 问题的基础模型。将 Pareto MTL 推广到其他软参数共享体系结构将是非常有趣的 [5]。一些关于任务关系学习的研究 [33–35] 也可能有助于 ParetoMTL 对不太相关的任务做出更好的权衡。

参考文献

[1] Rich Caruana. Multitask learning. Machine learning, 28(1):41–75, 1997.

[2] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6129–6138, 2017.

[3] Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. Learning general purpose distributed sentence representations via large scale multi-task learning. In International Conference on Learning Representations, 2018.

[4] Zhen Huang, Jinyu Li, Sabato Marco Siniscalchi, I-Fan Chen, Ji Wu, and Chin-Hui Lee. Rapid adaptation for deep neural networks through multi-task learning. In Sixteenth Annual Conference of the International Speech Communication Association, 2015.

[5] Sebastian Ruder. An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098, 2017.

[6] Yu Zhang and Qiang Yang. A survey on multi-task learning. arXiv preprint arXiv:1707.08114, 2017.

[7] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[8] Peng Wang, Ruigang Yang, Binbin Cao, Wei Xu, and Yuanqing Lin. Dels-3d: Deep localization and segmentation with a 3d semantic map. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5860–5869, 2018.

[9] Jaebok Kim, Gwenn Englebienne, Khiet P. Truong, and Vanessa Evers. Towards speech emotion recognition "in the wild" using aggregated corpora and deep multi-task learning. In 18th Annual Conference of the International Speech Communication Association, pages 1113–1117, 2017.

[10] Jin-Dong Dong, An-Chieh Cheng, Da-Cheng Juan, Wei Wei, and Min Sun. Dpp-net: Device-aware progressive search for pareto-optimal neural architectures. In Proceedings of the European Conference on Computer Vision (ECCV), pages 517–531, 2018.

[11] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. In International Conference on Learning Representations, 2019.

[12] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In Advances in Neural Information Processing Systems, pages 525–536, 2018.

[13] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In Proceedings of the 35th International Conference on Machine Learning, pages 794–803, 2018.

[14] Kaisa Miettinen. Nonlinear multiobjective optimization, volume 12. Springer Science & Business Media, 2012.

[15] Kristof Van Moffaert and Ann Nowé. Multi-objective reinforcement learning using sets of pareto dominating policies. The Journal of Machine Learning Research, 15(1):3483–3512, 2014.

[16] Marcela Zuluaga, Guillaume Sergent, Andreas Krause, and Markus Püschel. Active learning for multiobjective optimization. In International Conference on Machine Learning, pages 462–470, 2013.

[17] Daniel Hernández-Lobato, Jose Hernandez-Lobato, Amar Shah, and Ryan Adams. Predictive entropy search for multi-objective bayesian optimization. In International Conference on Machine Learning, pages 1492–1501, 2016.

[18] Amar Shah and Zoubin Ghahramani. Pareto frontier learning with expensive correlated objectives. In International Conference on Machine Learning, pages 1919–1927, 2016.

[19] Thomas Elsken, Jan Metzen, and Frank Hutter. Efficient multi-objective neural architecture search via lamarckian evolution. In International Conference on Learning Representations, 2019.

[20] Eckart Zitzler. Evolutionary algorithms for multiobjective optimization: Methods and applications, volume 63. Citeseer.

[21] Kalyanmoy Deb. Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Inc., New York, NY, USA, 2001.

[22] Jean-Antoine Désidéri. Mutiple-gradient descent algorithm for multiobjective optimization. In European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012), 2012.

[23] Jorg Fliege and A Ismael F Vaz. A method for constrained multiobjective optimization based on sqp techniques. SIAM Journal on Optimization, 26(4):2091–2119, 2016.

[24] Jörg Fliege and Benar Fux Svaiter. Steepest descent methods for multicriteria optimization. Mathematical Methods of Operations Research, 51(3):479–494, 2000.

[25] Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on evolutionary computation, 11(6):712–731, 2007.

[26] Anupam Trivedi, Dipti Srinivasan, Krishnendu Sanyal, and Abhiroop Ghosh. A survey of multiobjective evolutionary algorithms based on decomposition. IEEE Transactions on Evolutionary Computation, 21(3):440–462, 2016.

[27] Hai-Lin Liu, Fangqing Gu, and Qingfu Zhang. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. IEEE Trans. Evolutionary Computation, 18(3):450–455, 2014.

[28] Bennet Gebken, Sebastian Peitz, and Michael Dellnitz. A descent method for equality and inequality constrained multiobjective optimization problems. In Numerical and Evolutionary Optimization, pages 29–61. Springer, 2017.

[29] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In Advances in Neural Information Processing Systems, pages 3856–3866, 2017.Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In Advances in Neural Information Processing Systems, pages 3856–3866, 2017.

[30] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[31] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747, 2017.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[33] Amir R. Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3712–3722, 2018.

[34] Jiaqi Ma, Zhao Zhe, Xinyang Yi, Jilin Chen, and Ed H. Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining, 2018.

[35] Yu Zhang, Ying Wei, and Qiang Yang. Learning to multitask. In Advances in Neural Information Processing Systems, pages 5771–5782, 2018.