

用于快速适应深度网络的模型无关元学习

切尔西·芬恩¹ 彼得·阿贝尔^{1,2} 谢尔盖·莱文¹

抽象的

我们提出了一种元学习算法
与模型无关,因为它与任何使用梯度去气味训练的
模型兼容,并且适用于各种不同的

学习问题,包括分类、回归和强化学习。目标

元学习的目的是在各种学习任务上训练模型,使其
能够解决新的问题
仅使用少量训练样本的学习任务。在我们的方法中,
参数
该模型经过显式训练,使得一个小的
少量的梯度步数
来自新任务的训练数据将产生
该任务的泛化性能良好。在
效果,我们的方法训练模型很容易
进行微调。我们证明这种方法
在两个少量图像分类基准上实现最先进的性能,产
生
在少样本回归上取得了良好的结果,并且加速器对
策略梯度强化进行了微调
使用神经网络策略进行学习。

一、简介

快速学习是人类智力的标志
无论是从几个例子中识别物体,还是在几分钟后快速学习新技
能
经验。我们的人工智能代理应该能够做到
同样,仅从几个示例中快速学习和适应,并随着更多数据的可用
而继续适应。这种快速灵活的学习具有挑战性,

因为代理人必须将其先前的经验与
少量的新信息,同时避免对新数据的过度拟合。此外,先前经验
和新数据的形式将取决于任务。像这样,

为了获得最大的适用性,学习机制
学习(或元学习)应该是任务的通用性,并且

1加州大学伯克利分校2OpenAI。一致
二:切尔西·芬恩<cbfinn@eecs.berkeley.edu>。

第34届国际机器会议论文集
学习,澳大利亚悉尼,PMLR 70,2017。版权所有 2017
由作者撰写。

完成任务所需的计算形式。

在这项工作中,我们提出了一种元学习算法
是通用的并且与模型无关,因为它可以
直接应用于任何学习问题和模型
使用梯度下降程序进行训练。我们的重点
是关于深度神经网络模型,但我们说明了如何
我们的方法可以轻松处理不同的架构
不同的问题设置,包括分类、回归和策略梯度强化学习,只需进
行最小的修改。在元学习中,受训者的目标

模型是从少量的数据中快速学习一个新任务
新数据,模型由元学习器训练
能够学习大量不同的任务。
我们方法的关键思想是训练模型的
初始参数,以便在通过一个或多个梯度步骤更新参数后,模型
在新任务上具有最佳性能

来自该新任务的少量数据。与之前不同
学习更新函数的元学习方法或
学习规则 (Schmidhuber, 1987; Bengio 等, 1992;
安德里霍维奇等人, 2016;拉维和拉罗谢尔, 2017),我们的
算法不会扩展学习参数的数量,也不会对模型架构施加限制(例
如
通过需要一个循环模型 (Santoro et al., 2016)或
连体网络 (Koch, 2015)) ,并且它可以很容易地与全连
接、卷积或循环神经网络相结合。它还可以与各种损失函数一
起使用,包括可微的监督损失和不可微的强化学习目标。

训练模型参数的过程,使得
少数梯度步骤,甚至单个梯度步骤,可以在新任务上产生良好
的结果,可以从特征学习的角度来看,构建广泛适合许多任务的
内部表示。如果内部

表示适用于许多任务,只需进行微调
稍微修改参数(例如通过主要修改顶部
前馈模型中的层权重)可以产生良好的结果
结果。实际上,我们的程序针对以下模型进行了优化
可以轻松快速地进行微调,从而可以适应
发生在适合快速学习的空间。从动态系统的角度来看,我们的学
习过程可以被视为
最大化新损失函数的敏感性
与参数有关的任务:当灵敏度
高,参数的微小局部变化可能会导致

任务损失有很大改善。

这项工作的主要贡献是用于元学习的简单模型和与任务无关的算法,该算法训练模型的参数,以便少量的梯度更新将导致对新任务的快速学习。我们在不同的模型类型(包括全连接网络和卷积网络)以及几个不同的领域(包括少样本回归、图像分类和强化学习)上演了该算法。我们的评估表明,我们的元学习算法与专门为监督分类设计的最先进的一次性学习方法相比,使用更少的参数,但它也可以很容易地应用于回归并可以加速在存在任务可变性的情况下强化学习,大大优于直接预训练作为初始化。

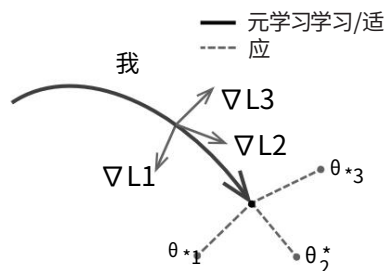


图 1. 我们的模型无关元学习算法 (MAML) 的图表, 该算法针对可以快速适应新任务的表示 θ 进行了优化。

2. 模型无关的元学习

我们的目标是训练能够实现快速适应的模型, 这种问题设置通常被形式化为小样本学习。在本节中, 我们将定义问题设置并呈现我们算法的一般形式。

2.1. 元学习问题设置

少样本元学习的目标是训练一个模型, 使其仅使用少量数据点和训练迭代即可快速适应新任务。为了实现这一目标, 模型或学习器在元学习阶段接受一组任务的训练, 这样训练后的模型只需使用少量的示例或试验就可以快速适应新任务。

实际上, 元学习问题将整个任务视为训练示例。在本节中, 我们以一般方式形式化此元学习问题设置, 包括不同学习领域的简短示例。我们将在第 3 节中详细讨论两个不同的学习领域。

我们考虑一个模型, 表示为 f , 它将观测值 x 映射到输出 a 。在元学习过程中, 模型被训练为能够适应大量或无限数量的任务。由于我们希望将我们的框架应用于从分类到强化学习的各种学习问题, 因此我们在下面引入学习任务的通用概念。形式上, 每个任务 $T = \{L(x_1, a_1, \dots, x_H, a_H), q(x_1), q(x_{t+1}|x_t, a_t), H\}$ 由损失函数 L 组成, 损失函数 L 为初始观测值 $q(x_1)$ 、转移分布 $q(x_{t+1}|x_t, a_t)$ 和情节长度 H 。在独立同分布监督学习问题中, 长度 $H = 1$ 。模型可以通过以下方式生成长度 H 的样本在每个时间 t 选择一个输出。损失 $L(x_1, a_1, \dots, x_H, a_H) \rightarrow R$ 提供特定于任务的反馈, 其形式可能是错误分类损失或马尔可夫决策过程中的成本函数。

在我们的元学习场景中, 我们考虑我们希望模型能够适应的任务 $p(T)$ 的分布。在 K -shot 学习设置中, 模型被训练为学习从 $p(T)$ 中提取的新任务 T_i , 该任务仅从 q_i 中提取的 K 个样本以及 T_i 生成的反馈 L_{T_i} 中提取。在元训练期间, 从 $p(T)$ 中采样任务 T_i , 使用 K 个样本和来自 T_i 的相应损失 L_{T_i} 的反馈来训练模型, 然后在来自 T_i 的新样本上进行测试。

然后通过考虑来自 q_i 的新数据的测试误差如何随参数变化而改进模型 f 。实际上, 采样任务 T_i 的测试误差充当元学习过程的训练误差。在元训练结束时, 从 $p(T)$ 中采样新任务, 并通过从 K 个样本学习后的模型性能来衡量元性能。通常, 用于元测试的任务会在元训练期间进行。

2.2. 与模型无关的元学习算法

与之前的工作相反, 之前的工作试图训练摄取整个数据集的循环神经网络 (San toro et al., 2016; Duan et al., 2016b) 或可以在测试时与非参数方法结合的特征嵌入 (Vinyals 等人, 2016; Koch, 2015), 我们提出了一种方法, 可以通过元学习来学习任何标准模型的参数, 从而为模型的快速适应做好准备。这种方法背后的直觉是, 某些内部表示比其他内部表示更容易转移。例如, 神经网络可能学习广泛适用于 $p(T)$ 中所有任务的内部特征, 而不是单个任务。我们如何鼓励这种通用表示的出现? 我们对这个问题采取了一种明确的方法: 由于模型将在新任务上使用基于梯度的学习规则进行微调, 因此我们的目标是以这种基于梯度的学习规则可以使从 $p(T)$ 中提取的新任务快速进展, 而不会过度拟合。实际上, 我们的目标是找到对任务变化敏感的模型参数, 这样, 当在以下方向改变时, 参数的微小变化将对从 $p(T)$ 得出的任何任务的损失函数产生很大的改进该损失的梯度 (见图 1)。我们

算法 1 与模型无关的元学习

要求: $p(T)$: 任务分配要求: α, β : 步长超参数 1: 随机初始化 θ 2: while not did do3: 任务样本批次 $T_i \sim p(T)$ 4: 对于所有 T_i 做的事情根据 K 个示例 \mathcal{S} 计算 $\nabla_{\theta} L_{T_i}(f_{\theta})$:6: 计算适应参数, 梯度 $de = \theta - \alpha \nabla_{\theta} L_{T_i}(f_{\theta})$ 7: $\theta \leftarrow de$ end for8: 更新 $\theta \leftarrow \theta - \beta \nabla_{\theta}$ 9: 结束 10: $T_i \sim p(T)$ $L_{T_i}(f_{\theta})$ while

对模型的形式不做任何假设, 只是假设它由某个参数向量 θ 参数化, 并且损失函数在 θ 中足够平滑, 以至于我们可以使用基于梯度的学习技术。

形式上, 我们考虑由参数化函数 f_{θ} 和参数 θ 表示的模型。当适应新任务 T_i 时, 在我们的方法中, 使用一个或多个梯度下降对任务 T_i 进行更新的参数向量 θ 进行更新。例如, 模型的参数 θ 变为 θ 。如, 当使用一次梯度更新时, $\theta \leftarrow \theta - \alpha \nabla_{\theta} L_{T_i}(f_{\theta})$ 。被计算为了

 θ_{T_i}

步长 α 可以被固定为超参数或元学习。为了简化符号, 我们将在本节的其余部分考虑一个梯度更新, 但使用多个梯度更新是一种简单的扩展。

通过优化 f_{θ} $p(T)$ 的性能来训练模型参数。更具体地说, 元目标如下: L_{T_i} 关于 θ 跨任务采样 $(f_{\theta} - \alpha \nabla_{\theta} L_{T_i}(f_{\theta}))$

$$\min_{\theta} \mathbb{E}_{T_i \sim p(T)} L_{T_i}(f_{\theta}) = \mathbb{E}_{T_i \sim p(T)} L_{T_i}(f_{\theta})$$

请注意, 元优化是在模型参数 θ 上执行的, 而目标使用更新的模型参数 θ 计算的。

实际上, 我们提出的方法旨在优化模型参数, 以便新任务上的一个或少量梯度步骤将在该任务上产生最有效的行为。

跨任务的元优化通过随机梯度下降 (SGD) 进行, 使得模型参数 θ 更新如下: $\theta \leftarrow \theta - \beta \nabla_{\theta}$

$$\nabla_{\theta} \mathbb{E}_{T_i \sim p(T)} L_{T_i}(f_{\theta}) \quad (1)$$

其中 β 是元步长。一般情况下, 完整的算法如算法 1 所示。

MAML 元梯度更新涉及通过梯度的梯度。在计算上, 这需要额外向后传递 f 来计算 Hessian 向量

产品, 由 TensorFlow 等标准深度学习库支持 (Abadi 等人, 2016)。在我们的实验中, 我们还对放弃反向传播和使用一阶近似进行了比较, 我们将在 5.2 节中对此进行讨论。

3. MAML 的种类在本节中, 我

们讨论用于监督学习和强化学习的元学习算法的具体实例。这些领域的不同之处在于损失函数的形式以及任务生成数据并将其呈现给模型的方式, 但是相同的基本适应机制可以应用于这两种情况。

3.1 监督回归和分类

少样本学习在监督任务领域得到了充分研究, 其目标是使用来自类似任务的先前数据进行元学习, 仅从该任务的几个输入/输出对中学习新函数。例如, 目标可能是在仅看到一个或几个赛格威示例后, 使用之前见过许多其他类型物体的模型对赛格威图像进行分类。同样, 在少样本回归中, 目标是在对具有相似统计特性的许多函数进行训练后, 仅根据从该函数采样的几个数据点来预测连续值函数的输出。

为了在第 2.1 节中元学习定义的背景形式化监督回归和分类问题, 我们可以定义地平线 $H = 1$ 并删除 x_t 上的时间步下标, 因为模型接受单个输入并产生单个输出, 而不是输入和输出的序列。任务 T_i 从 q_i 生成 K 个 iid 观测值 x , 任务损失由 x 的模型输出与该观测值和任务的相应目标值 y 之间的误差表示。

用于监督分类和回归的两个常见损失函数是交叉熵和均方误差 (MSE), 我们将在下面描述; 不过, 也可以使用其他监督损失函数。对于使用均方误差的回归任务, 损失的形式为: $(y - f_{\theta}(x))^2$

$$L_{T_i}(f_{\theta}) = \mathbb{E}_{x^{(j)}, y^{(j)} \sim T_i} (y^{(j)} - f_{\theta}(x^{(j)}))^2, \quad (2)$$

其中 x 是从任务 T_i 采样的输入/输出对。

在 K -shot 回归任务中, 提供 K 个输入/输出对用于每个任务的学习。

类似地, 对于具有交叉熵损失的离散分类任务, 损失的形式为: $-\log f_{\theta}(x^{(j)})$

$$L_{T_i}(f_{\theta}) = \mathbb{E}_{x^{(j)}, y^{(j)} \sim T_i} [-\log f_{\theta}(x^{(j)})] + (1 - y^{(j)}) \log(1 - f_{\theta}(x^{(j)})) \quad (3)$$

用于少样本监督学习的算法 2 MAML	
要求: $p(T)$:任务分配 要求: α, β :步长超参数 1:随机初始化 θ_2 : while not did do	
3: 任务样本批次 T_i $p(T)$ 4:对于所有 T_i do (j) (j) 5:	
从 T_i, y 中采样 K 个数据点 $D = \{x\}$ 使用公式(2) 6中的 D 和 L_{T_i} 计算 $\nabla \theta_{L_{T_i}}(f_{\theta})$: 或(3)	
7: 使用梯度下降计算自适应参数: $\theta = \theta - \alpha \nabla \theta_{L_{T_i}}(f_{\theta}) \theta_i(j)$ (j) = $\{x, y$	
8: 示例数据点 D 元更新 θ } 从 T_i 为 9: 结束	
10:更新 $\theta \leftarrow \theta - \beta \nabla \theta$ 等式2或 $L_{T_i}(f_{\theta})$ 使用每个 D θ 3中的 T_i $p(T)$ 和 L_{T_i} 11: end while	

根据传统术语, K -shot 分类任务使用每个类别的 K 个输入/输出对, 总共 NK 个数据点进行 N 路分类。给定任务 $p(T_i)$ 上的分布, 这些损失函数可以直接插入到2.2节中的方程中以执行元学习, 如算法2 中详述。

3.2.强化学习

在强化学习 (RL) 中, 小样本元学习的目标是使代理仅使用测试设置中的少量经验即可快速获取新测试任务的策略。新任务可能涉及实现新目标或在新环境中成功实现先前训练的目标。例如, 智能体可能会学会快速找出如何在迷宫中导航, 这样, 当面对新的迷宫时, 它就可以确定如何仅用少量样本就能可靠地到达出口。在本节中, 我们将讨论如何将 MAML 应用于 RL 元学习。

每个RL任务 T_i 包含一个初始状态分布 $q_i(x_1)$ 和一个转移分布 $q_i(x_{t+1}|x_t, a_t)$, 损失 L_{T_i} 对应于 (负) 奖励函数 R 。因此整个任务是一个马尔可夫决策具有水平线 H 的过程 (MDP), 其中允许学习者查询有限数量的样本轨迹以进行少样本学习。MDP 的任何方面都可能因 $p(T)$ 中的任务而变化。正在学习的模型 f_{θ} 是一种策略, 它将状态 x_t 映射到每个时间步 $t \in \{1, \dots, H\}$ 处的动作分布。任务 T_i 和模型 f_{θ} 的损失采用以下形式

$$L_{T_i}(f_{\theta}) = -\mathbb{E}_{x_t, a_t \sim f_{\theta}, q_{T_i}} \sum_{t=1}^H R_i(x_t, a_t)。$$

(4)

在 K -shot 强化学习中, 来自 f_{θ} 和任务 T_i (x_1, a_1, \dots, x_H) 的 K rollout 以及相应的奖励 $R(x_t, a_t)$ 可用于适应新任务 T_i 。

用于强化学习的算法 3 MAML	
要求: $p(T)$:任务分配 要求: α, β :步长超参数 1:随机初始化 θ_2 : while not did do	
3:对任务批次 T_i $p(T)$ 进行采样 4:对于 所有 T_i 执行的操作5:	
使用 T_i 中的 f_{θ} 采样 K 个轨迹 $D = \{(x_1, a_1, \dots, x_H)\}$ 使用 D 和评估 $\nabla \theta_{L_{T_i}}(f_{\theta})$	
6: 方程4中的 L_{T_i} 使用梯度下降计算适应参数: $\theta = \theta -$ $\alpha \nabla \theta_{L_{T_i}}(f_{\theta}) \theta_i = \{(x_1, a_1, \dots, x_H)\}$ 使用 f_{θ} T_i 9中的样本轨迹	
4: D :10结束:更新 θ 方程 8: 中的 $\theta \leftarrow \theta - \beta \nabla \theta$ 和 L_{T_i} 11: end while	

由于由于未知的动态, 预期奖励通常是不可微的, 因此我们使用策略梯度方法来估计模型梯度更新和元优化的梯度。由于策略梯度是一种同策略算法, 因此在 f_{θ} 适应过程中每个额外的梯度步骤都需要来自当前策略 f_{θ_i} 的新样本。我们在算法3 中详细介绍该算法。该算法与算法2 具有相同的结构, 主要区别在于步骤 5 和 8 需要从与任务 T_i 对应的环境中采样轨迹。

该方法的实际实现还可以使用最近提出的策略梯度算法的各种改进, 包括状态或动作相关的基线和信任区域 (Schulman et al., 2015)。

四、相关工作

我们在本文中提出的方法解决了元学习的一般问题 (Thrun & Pratt, 1998; Schmidhuber, 1987; Naik & Mammone, 1992), 其中包括小样本学习。元学习的一种流行方法是训练元学习器, 学习如何更新学习器模型的参数 (Bengio 等人, 1992; Schmidhuber, 1992; Bengio 等人, 1990)。该方法已应用于学习优化深度网络 (Hochreiter et al., 2001; Andrychowicz et al., 2016; Li & Malik, 2017), 以及学习动态变化的循环网络 (Ha et al., 2017)。最近的一种方法同时学习权重初始化和优化器, 用于小样本图像识别 (Ravi & Larochelle, 2017)。与这些方法不同的是, MAML 学习器的权重是使用梯度来更新的, 而不是学习更新; 我们的方法没有引入元学习的额外参数, 也不需要特定的学习器架构。

还开发了少样本学习方法

特定任务,例如生成建模 (Edwards & 斯托基, 2017; Rezende 等人, 2016)和图像识别 (Vinyals 等人, 2016)。一种成功的方法
少样本分类是学习使用 Siamese 网络 (Koch, 2015)或带有注意机制的递归 (Vinyals 等人, 2016; Shyam 等人, 2017; Snell)在学习的度量空间中比较新示例。

等人, 2017)。这些方法已经产生了一些最成功的结果,但很难直接推广到其他问题,例如强化学习。我们的相反,方法与模型的形式无关以及特定的学习任务。

元学习的另一种方法是在许多任务上训练记忆增强模型,其中循环

学习者接受培训以适应新任务的推出。这种网络已应用于少样本图像识别 (Santoro et al., 2016; Munkhdalai & Yu, 2017)学习“快速”强化学习代理 (Duan 等人, 2016b; 王等人, 2016)。我们的实验表明我们的方法在少量镜头分类方面优于循环方法。此外,与这些方法不同的是,我们的方法只是提供了良好的权重初始化和对两个学习器使用相同的梯度下降更新和元更新。因此,可以直接对学习器进行额外梯度步骤的微调。

我们的方法还与初始化方法有关
深层网络。在计算机视觉中,模型经过预训练
大规模图像分类已被证明可以学习
针对一系列问题的有效特征 (Donahue 等人, 2014)。相反,我们的方法显式优化了快速适应模型,使其能够适应新任务
仅举几个例子。我们的方法也可以查看明确最大化新任务损失的敏感性
模型参数。许多先前的工作已经探索了深层网络的敏感性,通常是在

初始化 (Saxe 等人, 2014; Kirkpatrick 等人, 2016)。这些作品中的大多数都考虑了良好的随机初始化,尽管许多论文已经解决了数据相关的初始化器 (Krahenbuhl 等人, 2017)。, 2016; 萨利曼 & Kingma, 2016),包括学习初始化 (Husken & 戈里克, 2000; 麦克劳林等人, 2015)。相比之下,我们的方法明确地训练参数的敏感性
给定的任务分配,允许极其高效
适应 K-shot 学习和快速学习等问题
只需一个或几个梯度步骤即可进行强化学习。

5. 实验评估

我们实验评估的目标是回答
以下问题: (1)MAML能否实现快速学习
新任务? (2) MAML可以用于元学习吗
在多个不同领域,包括监督回归、分类和强化学习? (3) 可以

通过 MAML 学习的模型可以通过额外的梯度更新和/或示例继续改进吗?

我们考虑的所有元学习问题都需要
在测试时对新任务进行一定程度的适应。什么时候可能的话,我们将我们的结果与接收到的预言机进行比较
任务的身份 (这是一个与问题相关的表示)作为附加输入,作为上限

模型的性能。所有的实验都是
使用 TensorFlow 执行 (Abadi 等人, 2016),它允许在元学习期间通过梯度更新进行自动微分。该代码可在线获取¹

5.1. 回归

我们从一个简单的回归问题开始,它说明了 MAML 的基本原则。每个任务都涉及从正弦波的输入到输出的回归,其中
正弦波的幅度和相位在之间变化
任务。因此, $p(T)$ 是连续的,其中幅度
变化范围为 $[0.1, 5.0]$,相位变化范围为 $[0, \pi]$,
输入和输出的维度均为 1。
在训练和测试期间,数据点 x 从 $[-5.0, 5.0]$ 中统一采样。损失是均方误差
预测值 $f(x)$ 与真实值之间。regressor 是一个具有 2 个隐藏层大小的神经网络模型
40 具有 ReLU 非线性。当使用 MAML 进行训练时,
我们使用一个梯度更新,其中 $K = 10$ 个示例
固定步长 $\alpha = 0.01$,并使用 Adam 作为元优化器 (Kingma & Ba, 2015)。基线同样是由 Adam 训练的。为了评估性能,我们在不同数量的 K 上微调单个元学习模型

示例,并将性能与两个基线进行比较:(a)
对所有任务进行预训练,这需要训练一个网络以回归到随机正弦函数,然后
测试时,使用自动调整的步长对 K 个提供的点进行梯度下降微调,以及

(b) 接收真实幅度和相位的预言机
作为输入。在附录C中,我们展示了与其他多任务和适应方法的比较。

我们通过微调学习的模型来评估性能
通过 MAML 和 $K = \{5, 10, 20\}$ 上的预训练模型
数据点。在微调期间,每个梯度步骤都是使用相同的 K 个数据点计算的。
定性结果,
如图2所示,并在附录B中进一步扩展
表明学习的模型能够快速适应
只有 5 个数据点,显示为紫色三角形,而
使用标准监督学习预训练的模型
在所有任务上都无法充分适应如此少的数据点,而不会出现灾难性的过度拟合。至关重要的是,当
 K 个数据点都在输入范围的一半内,

¹回归和监督实验的代码位于
github.com/cbfinn/maml RL 实验的代码位于github.com/cbfinn/maml_rl

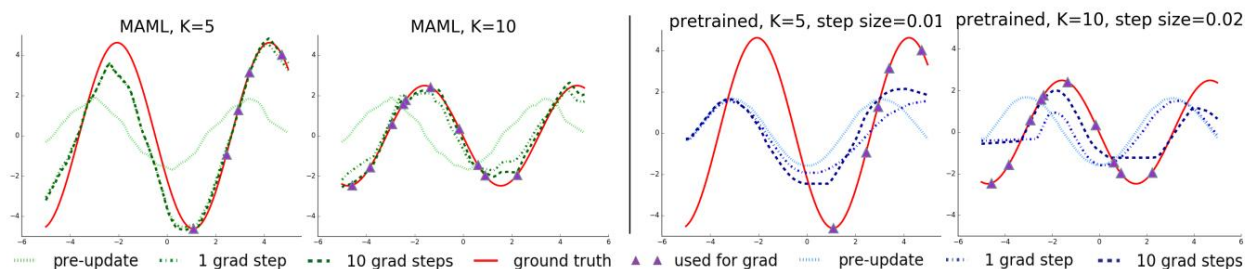


图 2. 简单回归任务的小样本适应。左: 请注意, MAML 能够估计曲线中存在的部分

没有数据点, 表明模型已经了解了正弦波的周期结构。右: 预训练模型的微调

在没有 MAML 的相同任务分布上, 具有调整的步长。由于预训练任务的输出经常相互矛盾, 该模型无法恢复合适的表示, 并且无法从少量测试时间样本中进行推断。

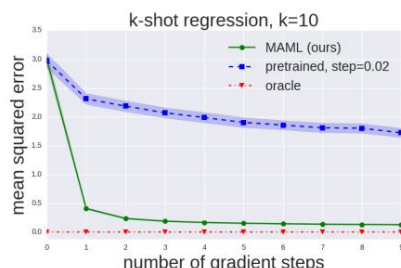


图 3. 定量正弦曲线回归结果显示

元测试时的学习曲线。请注意, MAML 继续通过额外的梯度步骤进行改进, 而不会过度拟合元测试期间的数据集非常小, 导致损失大大低于基线微调方法。

使用 MAML 训练的模型仍然可以推断振幅和范围的另一半相位, 表明 MAML 训练的模型 f 已学会对周期建模正弦波的性质。此外, 我们观察到模型通过 MAML 学习的定性和定量结果 (图3和附录B) 继续

通过额外的梯度步骤来改进, 尽管在一个梯度步骤后训练以获得最大性能。这一改进表明 MAML 优化了参数, 使它们位于适合的区域

快速适应并且对 $p(T)$ 的损失函数敏感, 正如第 2.2 节中所讨论的, 而不是过度拟合仅在一部后才有所改善的参数 θ 。

5.2. 分类

与之前的元学习相比评估 MAML 和少样本学习算法, 我们应用了我们的方法 Omniglot 上的少量图像识别 (Lake 等人, 2011) 和 Minilmagenet 数据集。Omniglot 数据集由来自 50 个不同字母表的 1623 个字符的 20 个实例组成。每个实例都是由不同的人绘制的。Minilmagenet 数据集由 Ravi 提出 & Larochelle (2017), 涉及 64 个培训班, 12 个验证类和 24 个测试类。万能者和 Minilmagenet 图像识别任务是最近使用的最常见的少样本学习基准 (Vinyals 等, 2016; 桑托罗等人, 2016; 拉维和拉罗谢尔, 2017)。

我们遵循 Vinyals 提出的实验方案

等人。 (2016), 其中涉及通过 1 或 5 个镜头快速学习 N 路分类。 N 路分类问题的设置如下: 选择 N 个未见过的类, 为模型提供 N 个类中每个类的 K 个不同实例

类, 并评估模型在 N 个类中对新实例进行分类的能力。对于 Omniglot, 我们随机选择 1200 个字符进行训练, 无论字母是什么, 并使用剩余的进行测试。 Omniglot 数据集是通过旋转 90 度的倍数来增强, 如由桑托罗等人提出。 (2016)。

我们的模型遵循与嵌入相同的架构 Vinyals 等人使用的函数。 (2016), 它有 4 个带有 3×3 卷积和 64 个滤波器的模块, 然后是批量归一化 (Ioffe & Szegedy, 2015)、ReLU 非线性和 2×2 最大池化。 Omniglot 图片被下采样到 28×28 , 所以维数最后一个隐藏层是 64。与使用的基线分类器一样由 Vinyals 等人。 (2016), 最后一层被输入到软最大值中。对于 Omniglot, 我们使用跨步卷积代替最大池化。对于 Minilmagenet, 我们每个使用 32 个过滤器层以减少过度拟合, 如 (Ravi & Larochelle, 2017)。为了也提供公平的比较记忆增强神经网络 (Santoro 等人, 2016) 为了测试 MAML 的灵活性, 我们还提供了非卷积网络的结果。为此, 我们使用具有 4 个隐藏层的网络, 大小分别为 256、128、64、64, 每个层都包括批量归一化和 ReLU 非线性, 后面是线性层和 softmax。对于所有型号, 损失函数是预测类别和真实类别之间的交叉熵误差。其他超参数详细信息是包含在附录 A.1 中。

我们在表 1 中列出了结果。卷积模型 MAML 学习到的结果与该任务的最先进结果相比较, 略胜于先前的方法。其中一些现有的方法, 例如匹配

网络、连体网络和记忆模型在设计时考虑了少样本分类, 而不是

很容易应用于强化学习等领域。此外, 通过 MAML 学习的模型使用

表 1. 对保留的 Omniglot 字符 (上)和 Minilmagenet 测试集 (下)进行少样本分类。MAML取得成果与最先进的卷积和循环模型相当或优于最先进的卷积和循环模型。连体网、匹配网和内存模块方法都是特定于分类的,并不直接适用于回归或强化学习场景。±表示95%任务的置信区间。请注意,Omniglot 结果可能不具有严格可比性,因为之前使用的训练/测试分割没有工作。Minilmagenet 对基线方法和匹配网络的评估来自Ravi & Larochelle (2017)。

	5 路精度 5 次 1 次		20路精度	
	94.9%	82.8%	1 次射击	5 发
Omniglot (Lake 等人, 2011)				
MANN,而非转化 (Santoro 等人, 2016)	89.7 ± 1.1%	97.5 ± 0.6%	-	-
MAML,无转化 (我们的)			-	-
连体网络(Koch, 2015)匹配网络	97.3%	98.4%	88.2%	97.0%
(Vinyals et al., 2016)神经统计学家(Edwards	98.1%	98.9%	93.8%	98.5%
& Storkey, 2017)记忆模型。(凯撒等人, 2017)	98.1%	99.5%	93.2%	98.1%
	98.4%	99.6%	95.0%	98.6%
MAML (我们的)	98.7 ± 0.4%	99.9 ± 0.1%	95.8 ± 0.3%	98.9 ± 0.2%

Minilmagenet (Ravi & Larochelle, 2017) 1-shot 5-shot		5路精度	
微调基线 最近邻基线		28.86 ± 0.54%	49.79 ± 0.79%
41.08 ± 0.70% 51.04 ± 0.65%			
匹配网络(Vinyals et al., 2016)元学习器		43.56 ± 0.84%	55.31 ± 0.73%
LSTM (Ravi & Larochelle, 2017) 43.44 ± 0.77% 60.60 ± 0.71%			
MAML,一阶约。(我们的) 48.07 ± 1.75% 63.15 ± 0.91%			
MAML (我们的)		48.70 ± 1.84%	63.11 ± 0.92%

与匹配网络相比,总体参数更少
和元学习器 LSTM,因为该算法不
引入权重之外的任何附加参数
分类器本身。与这些现有方法相比,
记忆增强神经网络 (Santoro 等人, 2016)
具体而言,一般来说,循环元学习模型代表了一类更广泛适用的方法,与 MAML 一样,可以用于其他任务,例如

强化学习 (Duan 等人, 2016b; Wang 等人, 2016)。然而,如比较所示,MAML 明显优于记忆增强网络,并且

元学习器 LSTM 在 5 路 Omniglot 和 Minilm Agenet 分类上的表现,无论是在 1-shot 还是 5-shot 情况下。

MAML 中的大量计算费用来自
通过梯度算子反向传播元梯度时使用二阶导数

元目标 (参见等式 (1))。在 Minilmagenet 上,
我们展示了与一阶近似的比较
MAML,其中这些二阶导数被省略。笔记
结果方法仍然计算元梯度
更新后的参数值 θ 我, 其中规定
有效的元学习。然而令人惊讶的是,该方法的性能与所获得的性能几乎相同
具有完整的二阶导数,表明大多数
MAML 的改进来自于梯度
目标是更新后的参数值,而不是
第二阶更新来自微分
梯度更新。过去的工作观察到 ReLU 神经网络局部几乎是线性的
(Goodfellow 等人, 2015),这表明二阶导数可能接近
在大多数情况下为零,部分解释了良好的性能

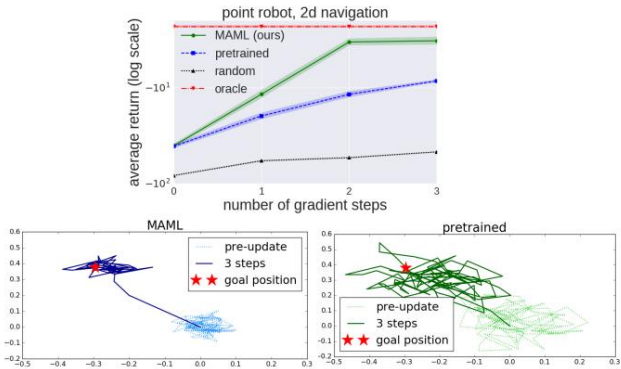


图 4. 上图:2D 导航任务的定量结果,下图:通过 MAML 学习的模型之间的定性比较

并通过预训练网络进行微调。

一阶近似的曼斯。这种近似消除了额外的向后传递中计算 Hessian 向量积的需要,我们发现这导致了

网络计算速度提高约 33%。

5.3.强化学习

为了评估强化学习问题上的 MAML,
我们根据 rllab 基准套件中的模拟连续控制环境构建了几组任务
(Duan 等人, 2016a)。我们讨论个人

下面的域。在所有领域中,模型都经过训练
MAML 是一种神经网络策略,具有两个大小为 100 的隐藏层,
具有 ReLU 非线性。梯度
更新是使用普通策略梯度 (RE INFORCE) (Williams, 1992)计算的,并且我们使用信任区域策略优化 (TRPO)作为元优化器 (Schulman 等人, 2015)。为了避免计算三阶导数,

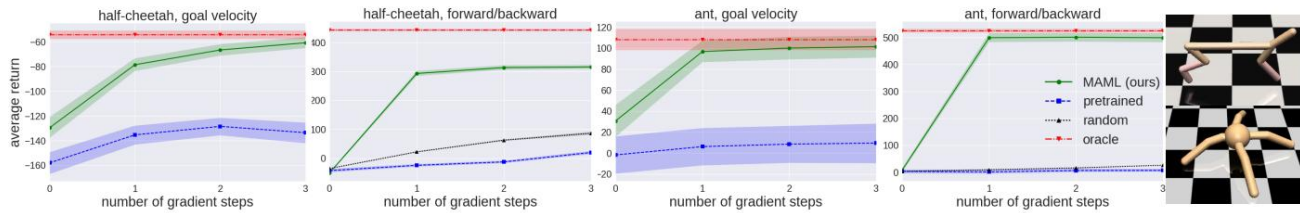


图 5.半猎豹和蚂蚁运动任务的强化学习结果,任务显示在最右侧。每个与监督学习任务不同,梯度步骤需要来自环境的额外样本。结果表明 MAML 可以比传统的预训练或随机初始化更快地适应新的目标速度和方向,取得了良好的效果只需两到三个梯度步骤即可执行。我们排除目标速度、随机基线曲线,因为回报要差得多(猎豹 < -200 , 蚂蚁 < -25)。

我们使用有限差分来计算 Hessian 向量 TRPO 的产品。对于学习和元学习更新时,我们使用 Duan 等人提出的标准线性特征基线。

(2016a), 分别安装在批次中每个采样任务的每次迭代。我们比较三个基线模型: (a) 预训练一项策略所有任务, 然后进行微调, (b) 训练策略来自随机初始化的权重, 以及 (c) 预言机策略它接收任务的参数作为输入, 其中对于以下任务, 对应于代理的目标位置、目标方向或目标速度。基线模型

(a) 和 (b) 通过梯度下降进行微调
手动调整步长。学习政策视频
可以在 sites.google.com/view/maml 查看

2D 导航。在我们的第一个元强化学习实验中, 我们研究一组任务, 点代理必须移动到不同的位置
2D 目标位置, 为每个任务随机选择一个单位正方形。观测值是当前的 2D 位置, 和动作对应于剪裁为的速度命令
在 $[-0.1, 0.1]$ 范围内。奖励是负平方到目标的距离, 当智能体在目标的 0.01 范围内或在 $H = 100$ 的范围内。
策略使用 MAML 进行训练, 以最大限度地提高性能使用 20 个轨迹进行 1 次策略梯度更新后。此问题的附加超参数设置

以下 RL 问题在附录 A.2 中。在我们的评估中, 我们将适应新任务与最多 4 个梯度更新进行比较, 每个梯度更新有 40 个样本。结果如图 4

显示使用 MAML 初始化的模型的适应性能, 在同一组上进行常规预训练

任务、随机初始化和预言机策略
接收目标位置作为输入。结果表明 MAML 可以学习适应速度更快的模型在一次梯度更新中, 并且继续通过额外的更新进行改进。

运动。研究 MAML 能够如何扩展到更多复杂的深度强化学习问题, 我们还研究适应使用 MuJoCo 模拟器执行高维运动任务 (Todorov 等人, 2012)。这些任务需要两个模拟机器人 平面猎豹和 3D 四足机器人 (

“ant”) 朝特定方向或特定方向奔跑速度。在目标速度实验中, 奖励是

当前速度之间的负绝对值

代理和目标, 统一随机选择

对于猎豹, 在 0.0 到 2.0 之间; 对于蚂蚁, 在 0.0 到 2.0 之间

3.0 对于蚂蚁来说。在目标方向实验中, 奖励是前向速度的大小

或向后方向, 为每个任务随机选择

$p(T)$ 。地平线为 $H = 200$, 除了蚂蚁向前/向后的所有问题, 每个梯度步骤 20 次推出

任务, 每个步骤使用 40 次部署。图 5 的结果表明 MAML 学习到的模型可以快速

即使仅通过一次梯度更新即可调整其速度和方向, 并通过更多梯度步骤继续改进。结果还表明, 在这些具有挑战性的

任务时, MAML 初始化的性能明显优于

随机初始化和预训练。事实上, 预训练

在某些情况下比随机初始化更糟糕, 这是一个事实

在之前的 RL 工作中观察到 (Parisotto 等人, 2016)。

6. 讨论和未来的工作

我们引入了一种基于学习的元学习方法

通过梯度去气味轻松调整模型参数。我们的方法有很多好处。很简单

并且不引入任何元学习的学习参数。它可以与任何模型表示相结合

适合基于梯度的训练, 以及任何可区分的目标, 包括分类、回归和

强化学习。最后, 由于我们的方法仅仅

产生权重初始化, 尽管我们展示了最先进的结果, 但可以使用任意数量的数据和任意数量的梯度步骤来执行适应

每类只有一个或五个示例的分类。

我们还表明, 我们的方法可以使用以下方法来调整 RL 代理政策梯度和非常有限的经验。

重用过去任务中的知识可能是制作高容量可扩展模型的关键, 例如

神经网络, 适合用小数据进行快速训练

数据集。我们相信这项工作迈向了简单且通用的元学习技术的一步, 该技术可以

适用于任何问题和任何模型。进一步的研究

在这一领域可以使多任务初始化成为深度学习和强化学习梯度的标准。

致谢

作者要感谢 Xi Chen 和 Trevor Darrell
如需有用的讨论,Yan Duan 和 Alex Lee 提供技术建议,Nikhil Mishra、Haoran Tang 和 Greg Kahn 提供技术建议
对论文初稿的反馈以及匿名
审稿人的意见。这项工作得到了支持
部分获得 ONR PECASE 奖和 NSF GRFP 奖。

参考

阿巴迪、马丁、阿加瓦尔、阿什什、巴勒姆、保罗、布雷夫多、尤金、陈、志峰、西特罗、克雷格、科拉多、格雷格·S、戴维斯、安迪、迪恩、杰弗里、德文、马蒂厄等人。+sorrow:异构上的大规模机器学习
分布式系统。 arXiv 预印本 arXiv:1603.04467, 2016年。

安德里霍维奇、马尔辛、丹尼尔、米沙、戈麦斯、塞尔吉奥、霍夫曼、马修·W、普福、大卫、斯考尔、汤姆和德弗雷塔斯、南多。学习通过梯度下降来学习
通过梯度下降。在神经信息处理中
系统（NIPS）,2016。

Bengio、Samy、Bengio、Yoshua、Cloutier、Jocelyn 和 Gecsei、Jan。关于突触学习的优化
规则。人工和生物神经最优性
网络,第 6-8 页,1992 年。

Bengio、Yoshua、Bengio、Samy 和 Cloutier、Jocelyn。学习突触学习规则。蒙特利尔大学计算机科学与运筹学系,1990 年。

多纳休、杰夫、贾、扬清、维尼亚尔斯、奥里奥尔、霍夫曼、朱迪、张、宁、曾、埃里克和达雷尔、特雷弗。 De caf:通用的深度卷积激活功能
视觉识别。国际机器学习会议（ICML）,2014 年。

Duan, Yan, Chen, Xi, Houthoof, Rein, Schulman, John, 还有阿比尔,彼得。深度强化基准测试
学习以实现持续控制。国际机器学习会议（ICML）,2016a。

段严、舒尔曼、约翰、陈曦、巴特利特、彼得 L、伊利亚·苏茨克维尔和彼得·阿比尔。 RL2:通过慢速强化学习实现快速强化学习。 arXiv
预印本 arXiv:1611.02779,2016b。

哈里森·爱德华兹和阿莫斯·斯托基。走向神经统计员。国际学习表征会议（ICLR）,2017 年。

伊恩·J·古德费洛、乔纳森·施伦斯和克里斯蒂安·塞格迪。解释和利用对抗性例子。
国际学习表征会议（ICLR）,2015。

Ha、David、Dai、Andrew 和 Le、Quoc V. 超网络。
国际学习表征会议（ICLR）,2017。

Hochreiter、Sepp、Younger、A Steven 和 Conwell、Pet ter R。学习使用梯度下降进行学习。在
国际神经网络会议。
施普林格,2001。

迈克尔·哈斯肯和克里斯蒂安·戈里克。快速学习
使用基于知识的网络初始化的问题类别。神经网络,2000。JCNN 2000, IEEE-INNS-ENNS 国际联合会议论文集
会议,第 6 卷,第 619-624 页。 IEEE,2000。

约夫、谢尔盖和克里斯蒂安·塞格迪。批量归一化：
通过减少内部加速深度网络训练
协变量平移。国际机器学习（ICML）,2015。

凯撒、卢卡斯、纳胡姆、奥菲尔、罗伊、奥尔科和本吉奥、萨米。学习记住罕见事件。国际学习表征会议（ICLR）,2017 年。

金玛、迪德里克和巴、吉米。亚当：一种方法
随机优化。国际会议
学习表示（ICLR）,2015。

柯克帕特里克、詹姆斯、帕斯卡努、拉兹万、拉比诺维茨、尼尔、维内斯、乔尔、德贾丁斯、纪尧姆、鲁苏、安德烈 A、Milan、Kieran、Quan、John、Ramalho、Tiago、Grabska Barwinska、Agnieszka 等。克服神经网络中的灾难性遗忘。 arXiv 预印本
arXiv:1612.00796,2016。

科赫、格雷戈里。用于一次性图像识别的连体神经网络。 ICML 深度学习研讨会,2015 年。

Krahenbühl、Philipp、Doersch、Carl、Donahue、Jeff 和 Darrell、Trevor。卷积神经网络的数据相关初始化。国际会议
学习表示（ICLR）,2016。

莱克、布伦登·M、萨拉胡迪诺夫、鲁斯兰、格罗斯、杰森、和 Tenenbaum、Joshua B. 一次性学习简单的
视觉概念。在认知科学会议上
社会（CogSci）,2011。

Li、Ke 和 Malik、Jitendra。学习优化。国际学习表征会议（ICLR）,2017年。

麦克劳林、杜格尔、杜维诺、大卫和亚当斯、瑞安。
通过可逆学习进行基于梯度的超参数优化。国际机器学习会议（ICML）,2015。

用于快速适应深度网络的模型无关元学习

Munkhdalai,Tsendsuren 和于洪。元网络有效。国际机器学习会议 (ICML),2017 年。

Naik,Devang K 和 Mammone,RJ。元神经网络通过学习来学习。在国际联席会议上神经网络 (IJCNN) ,1992。

Parisotto,Emilio,Ba,Jimmy Lei 和 Salakhutdinov,Ruslan。模仿演员:深度多任务和迁移强化学习。国际学习会议 陈述 (ICLR) ,2016。

拉维,萨钦和拉罗谢尔、雨果。优化作为少样本学习模型。在国际学习表征会议 (ICLR),2017 年。

雷伊、马克。当前神经语言模型中的在线表示学习。 arXiv 预印本 arXiv:1508.03854,2015。

雷森德、达尼洛·希门尼斯、穆罕默德、沙基尔、丹尼赫尔卡、伊沃、格雷戈尔、卡罗尔和维尔斯特拉、达安。深度生成模型中的一次性泛化。国际机器学习会议 (ICML),2016 年。

Salimans,Tim 和 Kingma,Diederik P. 权重标准化:加速训练的简单重新参数化深度神经网络。神经信息处理系统 (NIPS),2016 年。

桑托罗、亚当、巴图诺夫、谢尔盖、鲍特维尼克、马修、达安·维尔斯特拉和蒂莫西·利利克拉普。元学习使用记忆增强神经网络。国际机器学习会议 (ICML),2016 年。

安德鲁·萨克斯、詹姆斯·麦克莱兰和苏里亚·甘古利。学习中非线性动力学的精确解深度线性神经网络。国际会议学习表征 (ICLR) ,2014。

施米德胡贝尔,于尔根。自我参考学习的进化原则。关于学习如何学习:元-元-...钩子。)文凭论文,Institut f.信息技术,技术。大学。慕尼黑,1987 年。

施米德胡贝尔,于尔根。学习控制快速权重记忆:动态循环网络的替代方案。神经计算,1992。

舒尔曼、约翰·莱文、谢尔盖、阿贝尔、彼得、乔丹、迈克尔一世和莫里茨、菲利普。信托区域政策优化。在国际机器会议上学习 (ICML) ,2015。

Shyam,Pranav,Gupta,Shubham 和 Dukkipati,Ambekar。细心的经常性比较。国际机器学习会议 (ICML),2017 年。

Snell,Jake,Swersky,Kevin 和 Zemel,Richard S. Pro 构建了用于小样本学习的典型网络。 arXiv 预印本 arXiv:1703.05175,2017。

塞巴斯蒂安·特龙和洛里恩·普拉特。学习学习。施普林格科学与商业媒体,1998。

托多罗夫、伊曼纽尔、埃雷兹、汤姆和塔萨、尤瓦尔。穆乔科:用于基于模型的控制的物理引擎。国际智能机器人与系统会议 (IROS) ,2012。

维尼亚尔斯、奥里奥尔、布伦德尔、查尔斯、利利克拉普、蒂姆·维尔斯特拉、达安等人。用于一次性学习的匹配网络。在神经信息处理系统 (NIPS) ,2016。

王,Jane X,Kurth-Nelson,Zeb,Tirumala,Dhruva,索耶、休伯特、雷博、乔尔 Z、穆诺斯、雷米、布伦德尔、查尔斯、库马兰、达尔尚和博特维尼克,马特。学习强化学习。 arXiv 预印本 arXiv:1611.05763,2016。

Williams,Ronald J. 简单统计梯度跟踪连接主义强化学习算法。机器学习,8(3-4):229-256,1992。

A. 其他实验细节

在本节中,我们提供实验设置和超参数的更多详细信息。

A.1. 分类

对于 N 路、K 镜头分类,每个梯度都是使用 NK 个样本的批量大小来计算的。对于 Omniglot,5 路卷积和非卷积 MAML 模型均使用 1 个梯度步骤 (步长大小 $\alpha = 0.4$) 和 32 个任务的元批量大小进行训练。使用具有相同步长 $\alpha = 0.4$ 的 3 个梯度步骤对网络进行评估。使用步长 $\alpha = 0.1$ 的 5 个梯度步骤对 20 路卷积 MAML 模型进行训练和评估。在训练期间,元批量大小设置为 16 个任务。对于 MinImageNet,两个模型均使用 5 个大小为 $\alpha = 0.01$ 的梯度步长进行训练,并在测试时使用 10 个梯度步长进行评估。继 Ravi 和 Larochelle (2017) 之后,每个类使用 15 个示例来评估更新后的元梯度。我们分别使用 4 和 2 个任务的元批量大小进行 1-shot 和 5-shot 训练。

所有模型都在单个 NVIDIA Pascal Titan X GPU 上进行了 60000 次迭代训练。

A2. 强化学习

在所有强化学习实验中,MAML 策略均使用 $\alpha = 0.1$ 的单个梯度步骤进行训练。在评估过程中,我们发现在第一个梯度步骤之后将学习率减半会产生优异的性能。因此,第一步的适应过程中的步长设置为 $\alpha = 0.1$,所有后续步骤的步长设置为 $\alpha = 0.05$ 。针对每个域手动调整基线方法的步长。在 2D 导航中,我们使用的元批量大小为 20;在运动问题中,我们使用了 40 个任务的元批量大小。MAML 模型经过最多 500 次元迭代的训练,并使用训练期间平均回报最佳的模型进行评估。对于蚂蚁目标速度任务,我们在每个时间步添加了正奖励奖金,以防止蚂蚁结束情节。

B. 附加正弦曲线结果

在图6中,我们显示了 MAML 模型的完整定量结果,该模型经过 10 个镜头学习训练并在 5 个镜头、10 个镜头和 20 个镜头上进行评估。在图7中,我们显示了 MAML 的定性性能和随机采样正弦曲线上的预训练基线。

C. 附加比较

在本节中,我们对我们的方法进行了更彻底的评估,包括额外的多任务基线和 Rei (2015)方法的比较代表。

C.1. 多任务基线

正文中的预训练基线在所有任务上训练了单个网络,我们将其称为“所有任务上的预训练”。为了评估模型,与 MAML 一样,我们使用 K 个示例在每个测试任务上微调该模型。

在我们研究的领域中,不同的任务涉及相同输入的不同输出值。因此,通过对所有任务进行预训练,模型将学习输出特定输入值的平均输出。在某些情况下,该模型可能对实际域了解很少,而是了解输出空间的范围。

我们尝试了多任务方法来提供比较点,我们不是在输出空间中求平均值,而是在参数空间中求平均值。为了实现参数空间的平均,我们在从 $p(T)$ 中提取的 500 个任务上顺序训练了 500 个单独的模型。每个模型都是随机初始化的,并根据分配的任务中的大量数据进行训练。然后,我们采用模型的平均参数向量,并使用调整的步长对 5 个数据点进行微调。由于计算要求,我们对该方法的所有实验都是在正弦任务上进行的。各个回归系数的误差很低:在各自的正弦波上小于 0.02。

我们尝试了此设置的三种变体。在训练各个回归器期间,我们尝试使用以下方法之一:无正则化、标准 L2 权重衰减以及对迄今为止训练的回归器的平均参数向量进行 L2 权重正则化。后两种变体鼓励各个模型寻找简约的解决方案。使用正则化时,我们将正则化的幅度设置为尽可能高,而不会显着影响性能。在我们的结果中,我们将这种方法称为“多任务”。如表2中的结果所示,我们发现参数空间 (多任务) 中的平均性能比输出空间 (所有任务的预训练) 中的平均性能更差。这表明在单独训练任务时很难找到多个任务的简约解决方案,并且 MAML 正在学习比平均最优参数向量更复杂的解决方案。

C.2. 上下文向量适应

Rei (2015)开发了一种学习可以在线调整的上下文向量的方法,并应用于当前的语言模型。此上下文向量中的参数的学习和调整方式与 MAML 模型中的参数相同。为了与使用此类上下文向量解决元学习问题进行比较,我们将一组自由参数 z 连接到输入 x ,并且只允许梯度步骤修改 z ,而不是像 MAML 中那样修改模型参数 θ 。对于我来说

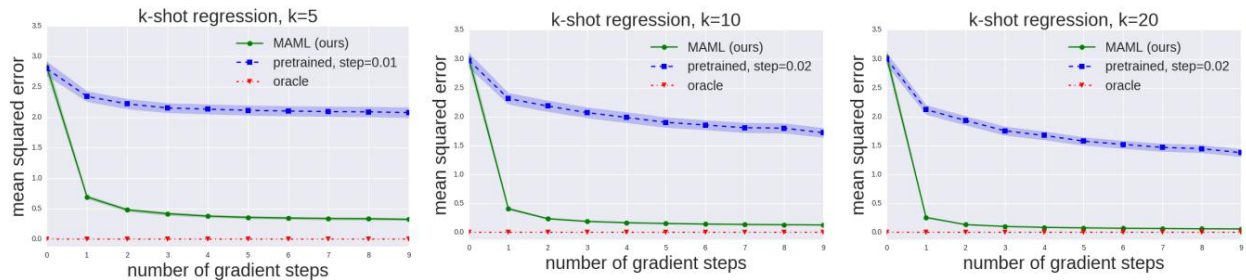


图 6. 定量正弦回归结果显示了具有不同数量的 K 个测试时间样本的测试时间学习曲线。每个使用相同的 K 个示例计算梯度步长。请注意,MAML 继续通过额外的梯度步骤进行改进,而无需在元测试期间过度拟合极小的数据集,并实现远低于基线微调的损失方法。

表 2. 正弦回归上的其他多任务基线,显示 5 次均方差。结果表明 MAML 正在学习一种比 MAML 更复杂的解决方案

平均最优参数向量。

编号grad 步骤多任	1		10
务,无 reg 多任务,l2	5 4.19 3.85 3.69		
reg 多任务,reg 均值	7.18 5.69 5.60		
0 2.91 2.72 2.71			
对所有任务进行预训练	2.41 2.23 2.19		
MAML (我们的)	0.67 0.38 0.35		

表 3. 5 路 Omniglot 分类		
1发 5发		
上下文向量	94.9 ± 0.9%	97.7 ± 0.3%
MAML	98.7 ± 0.4%	99.9 ± 0.1%

年龄输入,z 与输入按通道连接

图像。我们按照相同的实验协议在 Omniglot 和两个 RL do mains 上运行了这个方法。我们报道

结果如表3、表 4 和表 5 所示。学习自适应上下文向量在玩具点质量问题上表现良好，但在更困难的问题上低于标准,可能是由于较少灵活的元优化。

表 4. 2D 点质量,平均回报

编号梯度步长 2 上下文向量	0 -42.42	1 -13.90	2 -5.17	3
-3.18				
MAML (我们的)	-40.41 -11.68	-3.33 -3.23		

表 5 半猎豹前进/后退,平均回报

编号梯度步长上下文	0	1	2	3
向量	-40.49 -44.08	-38.27 -42.50		
MAML (我们的)	-50.69 293.19	313.48 315.65		

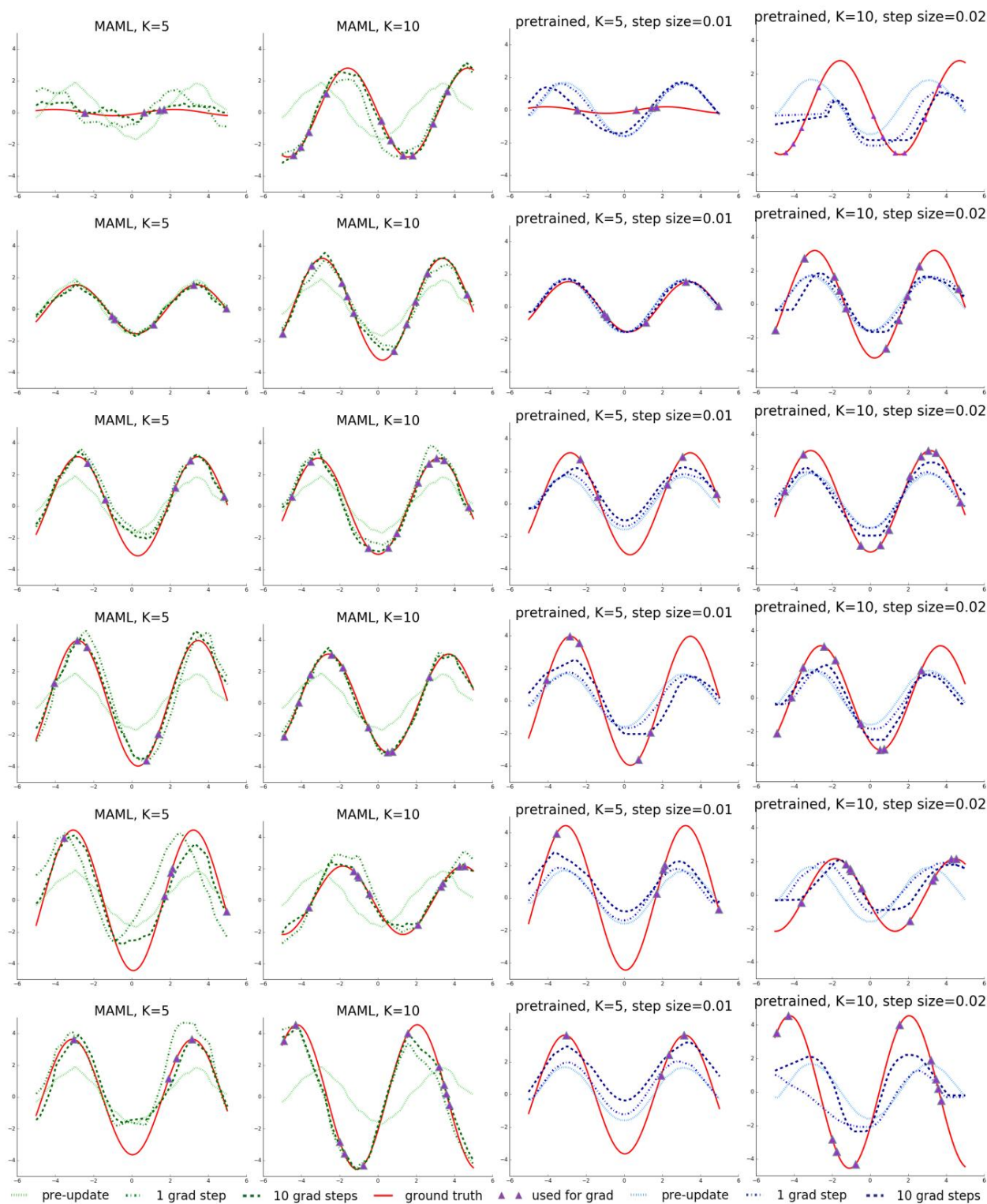


图 7. 正弦回归任务的定性结果的随机样本。