Orthogonal Subspace Learning for Language Model Continual Learning

正交子空间学习在语言模型连续学习中的应用

太长不看版

本文介绍了一种语言模型用于连续学习 (Continual Learning, CL) 的方法， O-LoRA。

之前针对连续学习的方法主要有三大类： 基于回放 (rehearsal-based)、基于正则化 (regularization-based) 和基于 架构 (architecture-based) 的方法。基于回放的方法涉及到数据隐私的问题，；基于正则化的方法难以处理长任务序列， 并且可能引入大量需要存储的信息， 难以应用于大语言模型； 而基于架构的方法本质上仍是为不同的任务训练不同 的专家模型，限制了其对未见任务的泛化能力。

之前的方法对于所有的任务都在一个共享的向量空间中进行更新， 在针对新任务进行梯度更新的时候， 自然而 然地会对先前的知识产生遗忘， 因此， 有研究者提出了一种正交梯度下降 (OGD) 的方法， 通过将参数约束在与之前 任务正交的空间内移动， 来缓解遗忘。但是， 这样的方法需要存储之前所有数据的梯度， 这对于大模型而言是不切 实际的。

考虑到 LoRA 的本质， 即在特定的低秩子空间内对大模型进行微调， 那么之前任务的梯度子空间也应该可以由 LoRA 参数表示。在此基础上， 作者假设过去 LoRA 矩阵子空间也可以近似表示其梯度空间， 由此， 提出了 O-LoRA。

O-LoRA，即在学习新任务的时候， 约束其 LoRA 子空间与之前任务的子空间正交， 保证了新任务的梯度更新 不对之前的任务产生干扰。

具体来说， 针对每个任务， 都学习一组 LoRA 参数 {At , Bt }，利用 At 的列向量来近似第 t 个任务的参数更新子

空间 Ut ，同时令 b 表示 At 的列向量的线性权重系数。为了满足正交性， 也需要在训练目标中添加相应的正交损失

项。尽管在训练过程中， 随着任务量的增加， LoRA 的参数量也会增加， 但可以将对应于 LoRA 参数的更新合并到 初始参数中，从而固定推理成本。

摘要

受益于大规模语料库和先进的硬件， 大语言模型 (LLMs) 展现出了卓越的语言理解能力与生成能力， 然而， 当 遇到多个连续任务时， 它们的性能会有一定的下降。在本文中， 作者提出了一种名为“正交低秩适应 (orthogonal low-rank adaptation, O-LoRA)”的方法， 用于在语言模型中进行持续性学习， 有效地减轻了在学习新任务中遗忘带来 的灾难性的影响。

具体而言， O-LORA 在不同的 (低秩) 向量子空间中学习任务， 这些空间保持正交， 以此来最小化干扰。这种方 法只引入了极少的额外参数， 并且不需要对用户数据进行存储， 以进行回放。在连续学习基准测试上的实验结果表 明，这种方法优于最先进的方法，并且可以保持 LLMs 在未见任务上的出色泛化能力。

**1.** 引言

顺序学习对于开发能够实际应用的自然语言处理模型至关重要， 因为它使得自然语言处理模型在遇到新任务或 新知识时能够持续演化。尽管预训练模型在静态任务上表现优异， 但是如果连续学习多个任务， 会遇到诸多挑战。 随着模型对新任务的学习，它往往会对先前获得的知识有一定的遗忘，这是一种灾难性的现象。

现有的持续学习 (continual learning) 方法主要可以分为： 基于回放 (rehearsal-based)、基于正则化 (regularization- based) 和基于架构 (architecture-based) 的方法。

基于回放 (rehearsal-based) 的方法允许访问来自先前任务的示例的内存缓冲区， 并与当前任务联合在一起训练 模型。然而， 存储、回放来自先前任务的数据可能引发一些隐私问题， 特别是这些数据可能涉及敏感的个人信息， 或可能包含以用于识别个人身份的信息。

基于正则化的方法 (regularization-based) 在损失函数中引入了额外的内容， 用以惩罚重要权重的变化， 保护先前 学习到的任务。但是，这种方法通常难以处理长任务序列。

基于架构 (architecture-based) 的方法通过动态扩展模型容量或隔离现有的模型权重，以减少干扰。 总之，这些方法本质上是为不同的任务学习不同的专家模型，限制了它们对未见任务的泛化能力。

现有的方法通常在共享的向量空间中更新所有的任务， 能够直接影响模型的隐藏层 (hidden layer) 输出。通过在 与过去任务相关的梯度子空间的正交方向上采取梯度步骤， 可以有效地减轻遗忘， 因为这可以防止与过去任务的损 失函数产生干扰。然而， 之前的方法要么需要存储历史数据 (涉及隐私问题)，要么需要历史数据的梯度 (这对大模 型而言是不切实际的)。

因此， 作者提出了 O-LoRA， 以进行持续学习。这种方法灵感来自于 LoRA，即在低秩子空间内对大型预训练 模型进行微调。基于这种方法，作者假设之前任务的梯度子空间可以有效地由 LoRA 参数进行捕捉。

O-LoRA 具有三个优点： 数据隐私友好 (不需要存储、回放用户数据)、模型参数友好 (引入参数成本很小)、泛 化友好 (保持了 LLMs 在未见任务上的泛化能力)。

**2.** 背景

**2.1** 持续学习

持续学习侧重于开发学习算法， 以在非静态数据上进行知识的积累。在监督式学习中， 一系列任务 {D1 , · · · , Dt }

以流方式给出即，数据或任务不是一次性全部提供，而是连续逐渐地、一个接一个地到达)，每个任务

包含一个独立的目标数据集， 其中，一个单一的模型需要按照顺序依次适应它们，并且只能够在

第 t 个任务的时候访问 Dt。

总的来说，给定一个由 Θ 参数化的预测模型 hΘ ,持续学习旨在优化跨所有任务的目标：

在本文中， 作者采用了更加严格的限制。在训练阶段， 模型被禁止访问任何历史数据； 在测试阶段， 模型将在 不知道样本属于哪个任务的情况下进行预测。

**2.2** 低秩适配器 **(LoRA)**

当预训练模型适应特定任务的时候， 其权重更新表现出低的“内在维度”，这意味着这些更新可以通过较低纬 度的变化来有效表示，而不需要更改整个权重矩阵的所有元素。

在 LoRA 中，对于一个预训练的权重矩阵 Wini ∈ Rd×k，其更新被限制为用一个低秩分解来表示，即：Winit +∆W = Winit x + ABx，其中， A ∈ Rd×r, B ∈ Rr×k，并且秩 r 远小于 d 和 k 的最小值 (即， r << min(d, k))。

在这样的设置中， 保持 Winit 不变， 即其不接受梯度更新， 而 A 和 B 则包含可训练的参数， 这样， 就可以以较 低的参数成本实现对预训练模型的有效微调。

在引入了 LoRA 后，前向传播就相应地变为了：

h = Winit + ∆W = Winit x + ABx

这种方式允许模型在保持原始预训练权重不变的同时，通过调整 A 和 B 来适应新的任务。

**3. O-LoRA**

**3.1** 指令模式

对于 LLMs 而言， 遵循指令的能力是人类与 AI 模型之间交互的重要桥梁。考虑到以下两点原因， 作者采用了 指令调优作为其训练范式：

1. 整合人类专业知识： 模型可以通过提供明确的指令利用先前的知识， 并受益于人类专业知识， 从而实现更高 效的学习；

2. 增强泛化能力：明确的指导有助于模型捕捉潜在原则，使其能够更好地泛化到未曾见过的情况。

所有的指令都遵循统一的模式， 由以下几个部分组成： 任务定义 (如何将输入文本映射到输出文本)、选项 (输 出标签的约束)、文本 (任务实例的输入)、答案 (给定样本的预期输出)。

**3.2** 正交子空间中的持续学习

之前的方法都具有一个共同特点： 所有任务都在共享的向量空间内进行更新， 直接影响模型的隐藏层输出。那 么，当在神经网络中应用相对于新任务的梯度更新而不考虑以前的任务时，就自然地会发生遗忘。

针对这个问题， 研究者提出了一种正交梯度下降 (OGD) 方法， 用以缓解遗忘。该方法将参数约束在与以前任务 的梯度正交的空间内移动， 由于对之前任务数据的访问有限， OGD 通过之前收敛参数中的梯度来近似之前数据的当 前梯度。然而， OGD 需要存储所有以前数据的梯度，这对于拥有数十亿参数的大规模语言模型而言是不切实际的。

因此， 作者利用 LoRA 的低秩子空间作为之前任务梯度子空间的代理， 灵感来自 LoRA 的本质——大型预训练 模型主要在特定的低秩子空间内进行微调。这种特征行为表明 LoRA 参数不仅仅是数值调整， 而且包含了重要的 模型更新方向。因此， 作者假设之前任务的梯度子空间可以简洁地由 LoRA 参数表示， 通过在与之前任务相关的 LoRA 子空间正交的子空间内进行学习，可以防止对以前任务的损失函数产生干扰，从而缓解遗忘。

O-LoRA 在保持之前参数不变的情况下， 在与之前任务的 LoRA 子空间正交的方向上逐步学习新的任务。对于 每个任务， 都引入一组 LoRA 参数， 表示为 {At , Bt }，其中， A ∈ Rd×r, B ∈ Rr×k，秩 r << min(d, k)。可以将第 t 个任务的参数更新子空间 Ut 近似为由 At 的列向量扩张成的子空间：

At = [a, a , · · · , a]

令 Bt = [b, b , · · · , b], 其中,b 表示 At 中列向量的线性权重系数。为了确保子空间 U 和 W 之间的正交性， 需

要满足：

< u, w >= 0, ∀u ∈ U, w ∈ W

因此，实现任务 i(Ui ) 和任务 t(Ut ) 的 LoRA 子空间之间的正交性可以表示为：

Oi,t = A, At = 0

最后，训练目标可以定义为：

其中， Oi,t [j, k] 表示 Oi,t 的第 j 行和第 k 列的元素， λ 1 是正交损失的权重。在训练过程中， 为了减少对以前知 识的遗忘，固定之前的 LoRA 参数 {Ai , Bi |i < t}。与之前的研究相似，作者仅将 LoRA 应用于 Wq 和 Wv 的权重。

虽然在训练过程中 LoRA 参数的数量随任务数增加， 但可以将与 LoRA 参数对应的更新合并到初始参数中， 以 避免 GPU 内存膨胀，即：

**3.3 O-LoRA** 与其他方法的比较

数据隐私友好

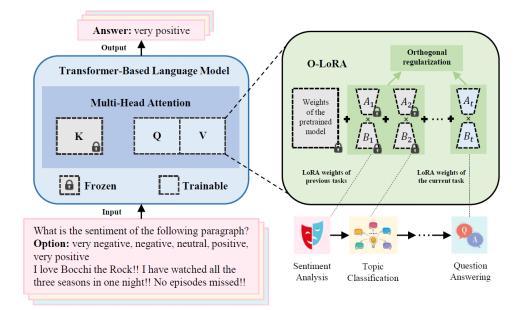
基于回放的方法依赖于在缓冲区中存储以前的任务数据并在训练期间回放， 不适用于涉及数据隐私的情况。此 外， 随着训练任务数量的增加， 使用基于回放的方法训练新任务的成本也会增加。而 O-LoRA 则不需要存储历史数 据，减轻了与数据隐私有关的担忧，同时，由于仅修改了训练损失，因此没有额外的训练成本。

模型参数友好

许多方法对每个任务都训练整个模型参数， 而 O-LoRA 仅为每个任务引入了微小的额外参数， 在计算资源和 GPU 内存方面， O-LoRA 要求较低。此外， 由于 LoRA 的训练会冻结预训练模型参数， 因此也不容易遗忘在预训练 期间获得的知识。

泛化友好

传统方法主要设计用于分类任务， 往往由于具有狭窄的任务特定重点， 在泛化到未见任务时表现不佳。而 O-LoRA 采用指令调整作为其训练范式， 有助于模型在训练数据中的特定示例之外泛化， 使其能够更有效地处理未 曾见过的情况。



O-LoRA 结构示意。首先， 通过指令调整允许整合人类专业知识并增强泛化能力。接下来， 分别使用 LoRA 来近似 每个任务的梯度子空间。对于每个按顺序到来的任务， 在增量学习新的 LoRA 的同时， 强制当前任务的 LoRA 与以 前的任务之间保持正交关系。

**4.** 实验

**4.1** 实验设置

**4.1.1** 数据集

标准持续学习基准测试

作者使用语言模型的持续学习基准测试来评估该方法， 该测试包括五个文本分类数据集： AG News 、Amazon reviews 、Yelp reviews, DBpedia 和 Yahoo Answers。作者采用了 T5 模型的 CL 设置， 遵循 LFPT5，并探索了基准测 试的三种不同顺序。附录 A.2 展示了任务的详细信息，任务序列在附录 A.3 中展示。

大量任务

作者通过对包含 15 个数据集的持续学习基准测试进行实验来评估该方法在更长的任务序列上的性能。其中包 括来自 CL 基准测试的五个任务、来自 GLUE 基准测试的四个任务 (MNLI ，QQP ，RTE ，SST2)、来自 SuperGLUE 基准测试的五个任务 (WiC ，CB ，COPA ，MultiRC ，BoolQ) 以及 IMDB 电影评论数据集。按照[之前研究者](https://arxiv.org/abs/2301.12314)提出过的 方法，作者为每个任务选择了 1000 个随机样本用于训练，并为每个类别保留了 500 个样本用于验证。

未见任务生成

为了评估该方法对 LLMs 的泛化能力的影响， 作者首先在 Alpaca 数据集上训练了一个 LLM，该数据集是一个 开源的多任务指令调整数据集。之后， 使用该预训练的 LLM 对标准的 CL 基准测试进行顺序训练。在顺序训练之 后， 采用 MMLU 数据集 (涵盖 STEM、人文科学和社会科学等各个领域的 57 个主题) 零样本基准测试， 评估了在各 种难度水平上的知识水平和问题解决能力。

**4.1.2** 指标

设 ai,j 表示在训练第 i 个任务后， 在第 j 个任务上的测试准确性， 使用平均准确性 (AA) 进行评估， 即在训练最 后一个任务后所有任务的平均准确性，计算方式为：

**4.1.3** 基线

作者使用了十种方法来评估 O-LoRA，除了基于提示词 (prompt-based) 的方法， 其余方法都使用了 LoRA 框架 (原文如此，但似乎下面许多方法并非基于提示词，也并未采用 LoRA)，这确保了一致的参数设置。

十种方法包括：

• SeqFT：在一系列任务上训练所有模型参数 (不添加任何正则化或回放来自之前任务的样本)

• SeqLoRA：在一系列任务上训练固定大小的 LoRA 参数 (不添加任何正则化或回放来自之前任务的样本)

• IncLoRA：在顺序任务系列上增量学习新的 LoRA 参数 (不添加任何正则化或回放来自之前任务的样本)

• Replay：使用记忆缓冲区对整个模型进行微调，并在学习新任务时回放旧任务的样本以避免遗忘。

• EWC：使用正则化损失对整个模型进行微调，该损失阻止更新可能干扰先前学习任务的参数。

• LwF：限制共享表示层与学习新任务前的原始状态相似。

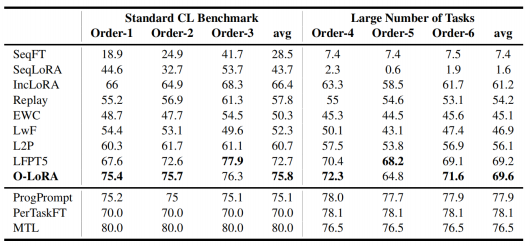
• L2P：使用输入动态地从提示池中选择和更新提示，以实例为基础进行更新。

• LFPT5：持续训练一个软提示，该提示同时学习解决任务和生成训练样本，这些样本随后用于经验回放。

• ProgPrompt：为每个不同任务采用特定的软提示， 顺序地将其添加到之前学习的提示中。本质上， 它为每个任 务训练单独的模型，在推理时利用任务 ID 选择适当的模型。

• PerTaskFT：为每个任务训练一个独立的模型。

• MTL：将所有任务作为多任务学习同时训练一个模型。这种方法是持续学习的上限。 结果如图所示：

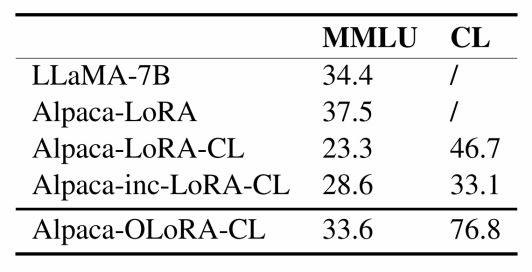


使用 T5-large 模型的两个标准 CL 基准测试的结果总结。展示了对最后一项任务进行训练后的平均准确性。所有的 结果都是平均自超过 3 次运行的结果。

**4.1.4** 实现细节

O-LoRA 是一种与模型无关的 CL(Continual Learning) 方法， 可以与任何基于 Transformer 的模型一起使用。在 实验中， 为了将 O-LoRA 与最近的 CL 方法进行比较， 采用了预训练的 T5-large 模型； 为了验证该方法对 LLMs 在 未见任务的泛化能力的影响， 采用了预训练的 LLaMA-7B 模型。所有实验结果都是 3 次运行的平均值。更详细的设 置请参阅附录 A.1。

**4.2** 主要结果



不同持续学习方法应用于 Alpaca-LoRA-LLaMA 模型的性能比较。这些方法在 (MMLU 零样本) 和 CL 基准测试 (顺 序 1) 上进行评估

标准的持续学习基准测试结果

在标准持续学习基准测试的所有任务顺序上， 与 LFPT5 相比， O-LoRA 的性能提高了超过 24%。该方法表现与 多任务学习相媲美， 并明显优于 PerTaskFT，这表明这种方法不仅有效地避免了灾难性遗忘， 而且利用了来自过去 任务的知识，以有效学习新任务。

大量任务的表现

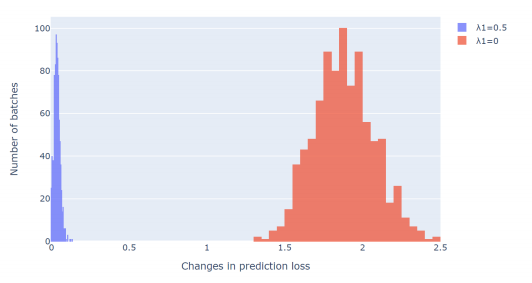
在涉及大量任务的基准测试中， O-LoRA 在三个任务顺序的平均性能方面优于最先进的方法 LFPT5。ProgPrompt 在处理长序列任务方面表现更佳， 但在推断过程中严重依赖任务 ID，限制了其泛化能力。此外， 几乎所有现有的持 续学习方法的性能都远低于 PerTaskFT 和 MTL，这表明针对大量任务的持续学习仍然是一个具有挑战性的问题。

泛化能力的影响

作者通过持续学习实验来调查 O-LoRA 对大型语言模型泛化能力的影响。从在 Alpaca 数据集上精调的 LLaMA- 7B 语言模型开始， 在 MMLU 基准测试中测试具有和不具有 O-LoRA 约束的模型。由上图可见， 没有 O-LoRA 的模 型正确率与随机猜测相当，而 O-LoRA 则表现出了一定的泛化能力。

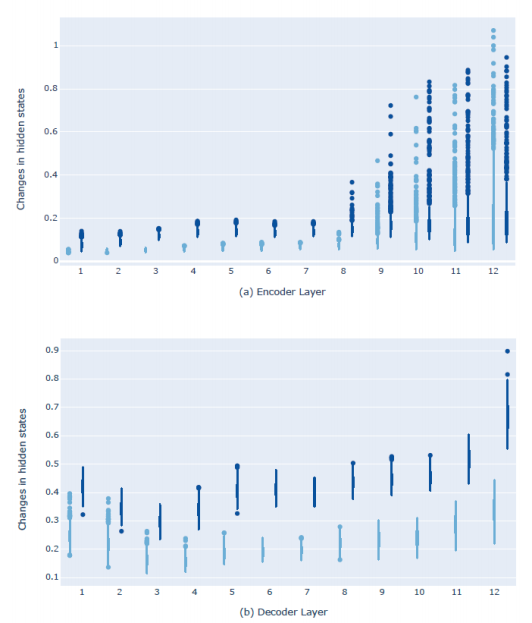
**4.3** 结论

1. O-LoRA 约束 (λ) 有助于保持以前样本的损失较低，有效地抵消了遗忘。



在训练新任务后， 预测损失变化的直方图。O-LoRA 约束 (λ = 0.5) 有助于减小变化， 相比之下， 没有该约束 (λ = 0) 时，变化较大。

2. O-LoRA 约束可以减少变化， 从而减少内部知识的遗忘。模型的低层编码更通用的语义知识， 而高层编码任务 特定的语义知识，并在学习新任务时发生较大变化。



T5-base 模型在具有和不具有 O-LoRA 约束的情况下， 不同层之间隐藏状态的变化。 (a) 展示了在训练新任务后 T5 编码器中每个层的隐藏状态变化。 (b) 显示了 T5 解码器中每个层的隐藏状态变化。浅蓝色表示在训练目标中使用正 交性损失，而深蓝色表示在训练目标中不包含正交性损失。

3. 不同的预训练语言模型对性能的影响：

(1). (在 T5 中) 随着参数大小的增加， O-LoRA 的平均准确率提高。

(2). 较大的网络尺寸似乎可以对抗灾难性遗忘，接近多任务学习的水平。

(3). 编解码器架构可能对遗忘更有抵抗力。

4. 通过在标准持续学习基准上使用 T5-Base 进行实验， 发现增加秩 r 可以在一定程度上提高模型的平均准确率。 然而， r = 2 与 r = 16 之间的性能差异不显著，表明模型的梯度空间具有相对较低的内在维度。

**5.** 相关工作

**5.1** 持续学习 **( Continual Learning,CL)**

持续学习旨在开发可以积累非静态数据上知识的学习算法。现有的工作可以广泛分类为基于回放、基于正则化 和基于架构的方法。

基于回放的方法

利用存储来自以前任务的示例的内存缓冲区， 与当前任务一起训练模型。经验回放 (ER) 是回放方法中常用的 策略， 可以作为一个强有力的基准。然而， 存储和回放来自以前任务的数据会引发隐私问题， 特别是在处理敏感信 息时。

基于正则化的方法

在损失函数中引入额外的项， 以惩罚关键权重的变化。例如， 正交梯度下降 (OGD) 将参数约束在由以前任务的 梯度定义的正交空间内移动。然而， OGD 需要存储所有历史数据的梯度， 这对于大型语言模型来说是不切实际的。 另一项工作引入了 C-LoRA，用于文本条件图像的持续学习， 它正则化了新 LoRA 参数与历史版本的相似性， 限制 了它们对新任务的学习可塑性。

基于架构的方法

侧重于动态扩展模型容量或隔离现有模型权重， 以减轻新旧任务之间的干扰。 ProgressivePrompt 为每个进入的 任务学习单独的提示， 并将它们顺序连接到以前学习的提示中。然而， 这样的方法本质上是为不同的任务训练不同 的专家模型，这限制了它们对未见任务的泛化能力。

**5.2** 参数高效微调 **(Parameter Efficient Tuning, PET)**

PET 旨在在最小化计算资源和标注工作的同时优化模型性能， 目前已经有多种方法实现参数效率调整， 其中一 个很有前景的方法是使用低秩适配器 (LoRA)。在 LoRA 的基础上， 作者提出了一种高效的持续学习神经架构， 涉 及将低秩适配器层叠在转换块的键和值投影矩阵上。正因为 LoRA 的优势， 在持续学习的背景下在模型性能和计算 效率之间取得平衡成为了可能。

**6.** 结论

本文介绍了一种名为 O-LoRA 的新方法， 用于语言模型的持续学习。 O-LoRA 通过在正交子空间内采用增量学 习策略， 有效地解决了灾难性遗忘问题。该方法注重数据隐私、高效利用模型参数， 并且对新任务具有强大的泛化 能力。实验证明了 O-LoRA 在持续学习中的有效性。

**7.** 不足

尽管 O-LoRA 在实验中表现出了一定的有效性，但仍存在一些限制：

1. 需要进一步研究它在更复杂情景下的性能和适用性；

2. 在训练过程中仍需要任务识别来为每个任务训练不同的 LoRA 参数。

未来的一个重要方向是探索任务无关训练方法。通过解决这个限制， 可以提高该方法的可扩展性和适用于各种 任务的能力，从而进一步推动语言模型的持续学习领域的发展。