自然语言处理中的参数高效迁移学习

尼尔霍尔斯比[1]安德烈朱尔吉乌[1] *Stanisław Jastrze,bski[2] *布鲁娜莫伦[1]昆汀·德·勒瓦西勒[1]安德里亚格斯蒙多[1]莫娜阿塔里扬[1]Sylwain果冻[1]

摘要

对大型预训练模型进行微调是自然语言处理中一种有效的传递机制。然而，在存在许多下游任务的情况下，微调是参数效率较低的：每个任务都需要一个全新的模型。作为一种替代方案，我们建议使用适配器模块进行传输。适配器模块产生一个紧凑和可扩展的模型；它们对每个任务只添加一些可训练的参数，并且可以添加新的任务，而无需重新访问以前的任务。原网络的参数保持不变，产生了高度的参数共享。为了证明适配器的有效性，我们将最近提出的BERT变压器模型转移到26个不同的文本分类任务中，包括GLUE基准测试。适配器达到接近最先进的性能，同时每个任务只添加几个参数。在GLUE上，我们获得了完全微调性能的0.4%以内，每个任务只添加3.6%的参数。相比之下，微调训练每个任务的100%的参数。[1]

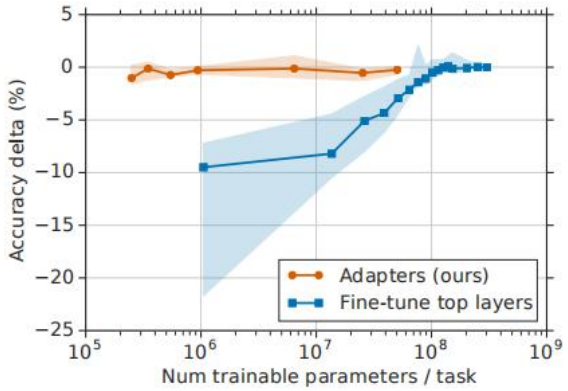

图1。在精度和训练过的任务特定参数的数量之间的权衡，用于适配器调整和微调。y轴通过完全微调的性能进行标准化，详见第3节。这些曲线显示了来自GLUE基准测试的9个任务中的第20、第50和第80个性能百分位数。基于适配器的调优获得了与完全微调相似的性能，使训练参数少了两个数量级。

1.介绍

从预先训练的模型转换在许多NLP任务上产生了强大的性能(Dai & Le，2015；Howard & Ruder，2018；雷德福等。，2018).BERT是一个在无监督损失的大型文本语料库上训练的变压器网络，在文本分类和提取问题回答方面取得了最先进的性能(Devlin等。，2018).

在本文中，我们讨论了在线设置，其中任务以流的形式到达。我们的目标是建立一个在所有系统上都表现良好的系统，但不需要为每一个新任务训练一个全新的模型。之间高度分享

任务对于云服务等应用程序特别有用，在这些应用程序中，需要对模型进行培训，以解决从客户那里依次到达的许多任务。为此，我们提出了一种迁移学习策略，它可以产生紧凑和可扩展的下游模型。紧凑模型是那些使用每个任务的少量附加参数来解决许多任务的模型。可扩展的模型可以通过逐步训练来解决新的任务，而不会忘记以前的任务。我们的方法可以在不牺牲性能的情况下产生这样的模型。

自然语言处理中最常见的两种迁移学习技术是基于特征的迁移和微调。相反，我们提出了一种基于适配器模块的替代传输方法(Rebuffi等。，2017).基于特征的转移涉及到预训练的实值嵌入向量。这些嵌入可能是在单词(Mikolov等人。，句子(Cer等人。或段落级（Le & Mikolov，2014）。然后，将嵌入的数据输入到自定义的下游模型中。微调包括从预先训练好的网络中复制权重，并在下游任务上对它们进行调整。最近的研究表明，微调往往能享受得更好

*相等贡献[1]谷歌研究[2]雅吉洛尼亚大学。通信地址：尼尔，来自<@谷歌。com>.

*36的程序*th*机器学习国际会议，长滩，加州，PMLR 97,2019。*

1代码在https://github。研究/适配器

性能比基于特性的传输更好（Howard & Ruder，2018）。

基于特性的传输和微调都需要为每个任务设置一组新的权重。如果网络的底层在任务之间共享，那么微调的参数效率就会更高。然而，我们提出的适配器调优方法的参数效率更高。图1演示了这种权衡。x轴表示每个任务训练的参数数量；这对应于解决每个额外任务所需的模型大小的边际增长。基于适配器的调优需要少训练两个数量级的参数来进行微调，同时获得类似的性能。

适配器是添加在预先训练过的网络层之间的新模块。基于适配器的调优不同于基于特性的传输和微调。考虑一个参数为w的函数（神经网络）：$\phi_{\mathbf{w}}(x)$.基于特征的传输组合$\phi_{\mathbf{w}}$有了一个新的函数，$x_{\mathbf{v}}$，以产生$x_{\mathbf{v}}(\phi_{\mathbf{w}}(x))$.然后只训练新的、特定于任务的参数v。微调包括为每个新任务调整原始参数w，从而限制紧凑性。对于适配器的调优，有一个新的功能，$_{\mathbf{w},\mathbf{v}}(x)$，被定义，其中参数w从训练前复制过来。初始参数$v_0$是否被设置为使新函数类似于原始函数：$_{\mathbf{w},\mathbf{v}_0}(x)\approx\phi_{\mathbf{w}}(x)$.在训练期间，只有v被调谐。对于深度网络，定义$_{\mathbf{w},\mathbf{v}}$通常包括向原始网络添加新的图层，$\phi_{\mathbf{w}\sim}$.如果选择$|v| < |w|$，则生成的模型需要许多任务的$|w|$参数。由于w是固定的，因此该模型可以扩展到新的任务，而不影响以前的任务。

*基于适配器的调优与多任务学习和持续学习有关。*多任务学习也会产生紧凑的模型。然而，多任务学习需要同时访问所有任务，而这却是基于适配器的调优所不需要的。持续学习系统的目标是从无穷无尽的任务流中学习。这种模式具有挑战性，因为网络在再训练后忘记了以前的任务（McCloskey & Cohen，1989；法语，1999）。适配器的不同之处在于任务不交互，共享参数被冻结。这意味着该模型使用少量的任务特定参数对之前的任务具有完美的记忆。

我们演示了一组大型和多样化的文本分类任务，适配器为NLP产生参数高效的调优。关键的创新是设计一个有效的适配器模块及其与基本模型的集成。我们提出了一个简单而有效的瓶颈架构。在GLUE基准测试中，我们的策略几乎与完全微调的BERT的性能相匹配，但只使用3%的特定于任务的参数，而微调使用100%的特定于任务的参数。我们在另外17个公共文本数据集和SQuAD提取的问题回答上观察到类似的结果。总之，基于适配器的调优就产生了一个单一的、可扩展的、在文本分类中达到接近最新性能的模型。

2.NLP适配器调谐

我们提出了一个在几个下游任务上调优大型文本模型的策略。我们的策略有三个关键属性：（i)它获得了良好的性能，（ii）它允许按顺序对任务进行训练，也就是说，它不需要同时访问所有数据集，并且（iii）它只为每个任务添加了少量的额外参数。这些属性在云服务的上下文中特别有用，在云服务中，许多模型需要对一系列下游任务进行训练，因此高度的共享是可取的。

为了实现这些特性，我们提出了一个新的瓶颈适配器模块。使用适配器模块进行调优需要向模型中添加少量的新参数，这些参数将在下游任务上进行训练(Rebuffi等人。，2017).当对深度网络进行普通的微调时，会对网络的顶层进行修改。这是必需的，因为上游和下游任务的标签空间和损耗不同。适配器模块执行更一般的架构修改，以重新将预先训练好的网络用于下游任务。特别是，适配器调优策略涉及到向原始网络中注入新的层。原始网络的权值是不变的，而新的适配器层是随机初始化的。在标准的微调中，新的顶层和原始的权重是共同训练的。相比之下，在适配器调优中，原始网络的参数被冻结，因此可能被许多任务共享。

适配器模块有两个主要特性：少量的参数和一个接近标识的初始化。与原始网络的图层相比，适配器模块需要较小。这意味着当添加更多的任务时，总模型大小增长相对较慢。对于适应模型的稳定训练，需要近恒等初始化；我们在3.6节对此进行实证研究。通过将适配器初始化为一个接近身份的函数，原始网络在训练开始时不受影响。在训练期间，适配器可能会被激活，以改变整个网络中激活的分布。如果不需要，适配器模块也可以被忽略；在第3.6节中，我们观察到一些适配器对网络的影响比其他的更大。我们还观察到，如果初始化偏离恒等函数太远，模型可能无法训练。

2.1.对变压器网络的实例化

我们为文本变形器实例化基于适配器的调优。这些模型在许多NLP任务中获得了最先进的性能，包括翻译、提取QA和文本

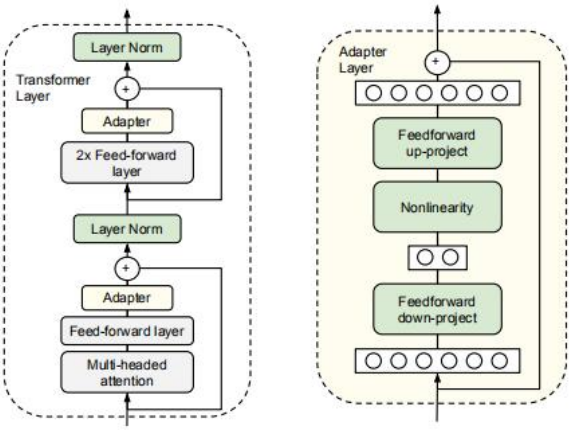

图2。适配器模块的体系结构及其与变压器的集成。左：我们向每个变压器层添加两次适配器模块：在多头注意之后的投影之后，在两个前馈层之后。右图：适配器包含一个瓶颈，它包含了相对于原始模型中的注意力和前馈层的少量参数。该适配器还包含一个跳过连接。在适配器调优过程中，绿色图层会对下游数据进行训练
包括适配器、层归一化参数和最终的分类层（图中未显示）。

分类问题(Vaswani等。，2017年；雷德福等人。，2018年；Devlin等人。，2018).我们考虑了标准的变压器架构，如Vaswani等人提出的那样。(2017).

适配器模块提供了许多体系结构上的选择。我们提供了一个简单的设计，以获得良好的性能。我们实验了一些更复杂的设计，见第3.6节，但我们发现以下策略在许多数据集上执行得和我们测试的任何其他策略一样好。

图2显示了我们的适配器架构，及其对变压器的应用。变压器的每一层都包含两个主要的子层：一个注意层和一个前馈层。两个层后面都有一个投影，该投影将特征大小映射回层输入的大小。在每个子层上都应用了一个跳跃式连接。每个子层的输出被输入到子层的归一化中。我们在每个子层之后插入两个串行适配器。适配器总是直接应用到子层的输出上，在投影回到输入大小之后，但在添加跳过连接回来之前。然后，将适配器的输出直接传递到下面的图层规范化中。

为了限制参数的数量，我们提出了一个瓶颈架构。适配器首先将原始的d维特征投影到一个更小的维度m中，应用一个非线性，然后投影回d维。每层添加的参数总数，包括偏差，为2md + d + m。通过设置m < d，我们限制了每个任务添加的参数的数量；在实践中，我们使用了原始模型中大约0 8%的参数。.5—瓶颈维度m提供了一种简单的方法来权衡性能和参数效率。适配器模块本身在内部有一个跳过连接。通过跳跃连接，如果投影层的参数被初始化为接近于零，则该模块被初始化为一个近似的恒等函数。

除了适配器模块中的层外，我们还训练每个任务的新的层标准化参数。这种技术

nique，类似于条件批处理标准化(De Vries等。，FiLM (Perez等人。和自调制(Chen等人。，2019)，也产生网络的参数效适应；每层只有2d参数。然而，仅训练层归一化参数并不足以获得良好的性能，见
第3.4节。

3. 实验

我们证明了适配器可以实现对文本任务的参数高效传输。在GLUE基准测试上(Wang等。，2018年)，适配器调优在BERT的完全微调的0.4%以内，但它只增加了通过微调训练的参数数量的3%。我们在进一步的17个公共分类任务和快速问题回答上确认了这一结果。分析表明，基于适配器的调优会自动关注网络的更高层次。

3.1. 实验设置

我们使用公共的，预先训练过的BERT变压器网络作为我们的基础模型。为了使用BERT进行分类，我们遵循Devlin等人的方法。(2018).每个序列中的第一个标记是一个特殊的“分类标记”。我们在这个标记的嵌入上附加一个线性层来预测类标签。

我们的训练程序也遵循了Devlin等人的方法。(2018).我们使用Adam（Kingma & Ba，2014）进行优化，其学习率在前10%的步骤中线性增加，然后线性衰减到零。所有的运行都在4个谷歌云tpu上进行训练，批处理大小为32。对于每个数据集和算法，我们运行一个超参数扫描，并根据验证集上的准确性选择最佳模型。对于GLUE任务，我们报告由提交网站提供的测试指标[2].对于其他的分类任务，我们报告了测试集的准确性。

2https://gluebenchmark.com/