论文译文 SimAM- A Simple, Parameter-Free Attention Module for Convolutional Neural Networks

论文译文 SimAM: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks

Abstract

在本文中，我们为卷积神经网络（ConvNets）提出了一个概念简单但非常有效的注意力模块。与现有的通道注意力模块和空间注意力模块不同，我们的模块在没有给原始网络增加参数的情况下，在一个层中推导出特征图的三维注意力权重。具体来说，我们以一些著名的神经科学理论为基础，提出通过优化能量函数来找出每个神经元的重要性。我们进一步推导了能量函数的快速闭式解，并表明该解可以在不到十行代码中实现。该模块的另一个优点是，大多数运算符都是根据定义的能量函数的解来选择的，从而避免了过多的结构调整工作。对各种视觉任务的定量评估表明，所提出的模块既灵活又有效，可以提高许多 ConvNets 的表示能力。我们的代码可在 Pytorch-SimAM 上获取。

1 Introduction

在大规模数据集（如 ImageNet（Russakovsky 等人，2015 年））上训练的卷积神经网络（ConvNets）大大提高了许多视觉任务的性能，如图像分类（Krizhevsky 等人，2012 年；Simonyan & Zisserman，2014 年；He 等人，2016 年b；Huang 等人，2017 年；Szegedy 等人，2015 年；Sandler 等人，2018 年）、物体检测（Ren 等人，2015 年；Sandler 等人，2018 年）、2012；Simonyan & Zisserman，2014；He 等人，2016b；Huang 等人，2017；Szegedy 等人，2015；Sandler 等人，2018）、物体检测（Ren 等人，2015；Liu 等人，2016；He 等人，2017）和视频理解（Feichtenhofer 等人，2016；Wang 等人，2018a）。多项研究表明，更好的 ConvNet 结构可以显著提高各种问题的处理性能。因此，构建强大的 ConvNet 是视觉研究中的一项重要任务。

现代 ConvNet 通常有多个阶段，每个阶段由几个区块组成。这些区块由卷积、池化、激活或一些定制的元结构（本文中称为模块）等运算符构成。最近，许多作品不再像（Krizhevsky 等人，2012 年）那样设计整个架构，而是专注于构建高级模块，以提高 ConvNets 的表征能力。堆叠卷积（Simonyan & Zisserman，2014 年）、残差单元（He 等人，2016b;a；Zagoruyko & Komodakis，2016 年；Sandler 等人，2018 年）和密集连接（Huang 等人，2017 年；2018 年）是最具代表性的几种，已被广泛应用于现有架构中。然而，设计这些模块需要丰富的专业知识和大量的时间。为了避免这种情况，许多研究人员寻求一些搜索策略来自动构建体系结构(Zoph & Le , 2016 ; Liu et al . , 2018b ; Dong & Yang , 2019 ; Tan & Le , 2019 ;郭庆旺等, 2020 ; Liu et al . , 2019 ;费希滕霍夫, 2020 ; Tan et al , 2020)。

除了设计复杂的模块外，另一类研究重点是构建即插即用模块(Hu et al . , 2018b ; Woo et al . , 2018 ; Cao et al . , 2020 ; Lee et al . , 2019 ; Wang et al . , 2020 ; Yang et al , 2020)，该模块可以细化一个模块内的卷积输出，使整个网络能够学习到更多的信息特征。例如，挤压和激发（SE）模块（Hu 等人，2018b）可以让网络捕捉与任务相关的特征（见图 1 中的 "山地帐篷"），并抑制许多背景激活（见图 1 中的 "钢拱桥"）。该模块独立于网络结构，因此可插入各种网络，如 VGG（Simonyan & Zisserman，2014 年）、ResNets（He 等人，2016 年 b）和 ResNeXts（Xie 等人，2017 年）。最近，SE 模块作为一个组件被纳入 AutoML，以搜索更好的网络结构（Howard 等人，2019；Tan & Le，2019）。

然而，现有的注意力模块存在两个问题。首先，它们只能在通道或空间维度上提炼特征，限制了它们学习跨通道和空间变化的注意力权重的灵活性。其次，它们的结构是由一系列复杂的因素构建的，例如，池化的选择。我们通过提出一个基于成熟的神经科学理论的模块来解决这些问题。具体来说，为了使网络学习到更有判别力的神经元，我们提出从当前神经元直接推断3 - D权重(也就是说,同时考虑空间维度和通道维度)，然后反过来精化这些神经元。为了有效推断这种三维权重，我们以神经科学知识为指导，定义了一个能量函数，并得出了一个闭式解。如图 1 所示，我们的模块可以帮助网络捕捉到许多与图像标签一致的有价值线索（参见 "山地帐篷 "和 "灰鲸 "的例子）。此外，我们模块中使用的大多数运算符都是从能量函数的解中得到的，不需要其他繁琐的步骤。

值得强调的是，我们主要关注的是一个小型的即插即用模块，而不是超越现有 ConvNets 的新架构。之前的一项研究（Wang 等人，2017 年）也试图推断三维权重。他们基于手工制作的编码器-解码器结构取得了可喜的成果。与该研究相比，我们的工作提供了另一种生成三维权重的高效方法。我们的模块更灵活、更模块化，而且仍然保持轻量级。总之，我们的主要贡献如下：

- 受人脑注意力机制的启发，我们提出了一个具有全三维权重的注意力模块，并设计了一个能量函数来计算权重。

- 我们推导出了能量函数的闭式解，从而加快了权重计算速度，并使整个模块采用了轻量级形式。

- 我们将提出的模块集成到一些著名的网络中，并在各种任务中对它们进行评估。我们的模块在准确性、模型大小和速度方面都优于其他流行模块。

2 Related Work

.....

3 Method

在本节中，我们首先总结了一些具有代表性的注意力模块，如 SE（Hu 等人，2018b）、CBAM（Woo 等人，2018）、GC（Cao 等人，2020）。然后，我们介绍我们的新模块，它与之前的方法有着相似的理念，但表述方式却截然不同。

1. Overview of existing attention modules

现有的注意力模块通常被集成到每个模块中，以完善前几层的输出。这种细化步骤通常沿着通道维度(图5a)或空间维度(图5b)进行。因此，这些方法生成1 - D或2 - D权重，并对每个通道或空间位置的神经元一视同仁，这可能会限制它们学习更有辨别力的线索的能力。例如，图1显示SE丢失了'灰鲸'的一些主要成分。因此，我们认为全3 - D权重优于传统的1 - D和2 - D注意力，并提出使用全3 - D权重对特征进行优化，如图2c所示。

注意力模块的另一个重要因素是权重生成方法。现有的大多数作品都是根据一些毫无根据的启发式方法来计算注意力权重的。例如，SE 使用全局平均池化（GAP）来捕捉全局上下文。通过在 CBAM 和 GC 中分别添加全局最大池化（GMP）和基于软最大值的池化，进一步改进了上下文聚合。在表1中，我们列出了以往工作中使用的主要算子。可以看出，现有的模块建立在许多通用的操作符上，如FC，Conv2D，BN等，以及一些高度定制的操作符，如通道级全互连层(Channel-wise全互连layers，CFC)。总之，结构设计的选择具有重要的工程意义。我们认为，注意机制的实现应该遵循神经计算中的一些统一原则。因此，我们基于一些成熟的神经科学理论提出了一种新的方法。

2. Our attention module

正如我们之前所述，现有的计算机视觉中的注意力模块主要集中在通道域或空间域。这两种注意机制正好对应了人脑(卡拉斯科, 2011)中基于特征的注意和基于空间的注意。然而，在人类中，这两种机制共存，共同作用于视觉加工过程中的信息选择。因此，我们提出了一个注意力模块来进行类似的操作，使得每个神经元被赋予一个唯一的权重。然而，直接估计完整的三维权重是具有挑战性的。(Wang et al . , 2017)提出使用编码器-解码器框架来学习三维权重。但是这种方法从一个ResNet的低层到高层添加了不同的子网络，这不能很容易地扩展到其他模块化的p网络。另一个例子，CBAM，分别估计1 - D和2 - D权重，然后将它们组合起来。该方法并不直接生成真实的三维权重(Woo et al , 2018)。CBAM中的两步法计算时间太长。因此，我们认为3 - D权重的计算应该是直接的，同时允许模块保持轻量级特性。

根据上述论点，在这里我们提出了一个模块，可以有效地产生真正的3 - D权重。为了成功地实施注意，我们需要估计单个神经元的重要性。如何根据一层的特征图计算单个神经元的重要性?在视觉神经科学中，信息最丰富的神经元通常是那些表现出与周围神经元不同的放电模式的神经元。此外，一个活跃的神经元还可以抑制周围神经元的活动，这种现象被称为空间抑制(Webb et al , 2005)。也就是说，在视觉处理中，表现出明显空间抑制效应的神经元应该优先考虑(即,重要性)。寻找这些神经元最简单的实现方式是衡量一个目标神经元与其他神经元之间的线性可分性。基于这些神经科学发现，我们为每个神经元定义如下能量函数：

这里，(t = wtt + bt和(xi = wtxi + bt是t和xi的线性变换，其中t和xi是目标神经元和输入特征X∈RC × H × W的单个通道中的其他神经元。I为空间维度上的索引，M = H × W为该通道上的神经元个数，wt和bt为权重，偏置变换。方程(1)中的所有值都是标量。当(t = yt时，方程(1)取得最小值，其他所有的(xi都是yo，其中yt和yo是两个不同的值。通过极小化该方程，方程(1)等价于寻找目标神经元t与同一通道内所有其他神经元之间的线性可分性。为了简单起见，我们对yt和yo采用二进制标签(即1和- 1)，并在方程(1)中加入正则项。最终的能量函数为：

理论上，每个通道有M个能量函数。通过像SGD这样的迭代求解器来求解所有这些方程在计算上是很麻烦的。幸运的是，方程(2)有一个关于wt和bt的快速闭式解，可以很容易地得到：

μt = 1 M -1 ∑M -1 i=1 xi 和 σ2 t= 1 M -1 ∑M -1 i (xi -μt)2 是对该通道中除 t 以外的所有神经元计算的均值和方差。由于公式 (3) 和公式 (4) 中显示的现有解决方案都是在单通道上获得的，因此可以合理地假设单通道中的所有像素都遵循相同的分布。根据这一假设，可以计算出所有神经元的均值和方差，并重复用于该通道上的所有神经元（Hariharan 等人，2012 年）。避免为每个位置反复计算 μ 和 σ，可以大大降低计算成本。因此，最小能量的计算方法如下：

其中，ˆμ= 1 M ∑M i=1 xi，ˆσ2= 1 M ∑M i=1(xi - ˆμ)2。公式（5）表明，能量 et∗ 越低，神经元 t 与周围神经元的区别越明显，对视觉处理越重要。因此，每个神经元的重要性可以通过 1/et∗ 得到。与我们的方法类似，（Aubry 等人，2014 年）研究了语义部分匹配的类似函数。但他们的方法需要计算较大的协方差矩阵，不适合深度神经网络。与（Aubry 等人，2014 年）不同的是，我们对单个神经元进行操作，并将这种线性分离性整合到端到端框架中。此外，我们还提供了对神经科学的全面理解。

到目前为止，我们得出了一个能量函数，并发现了每个神经元的重要性。根据（Hillyard 等人，1998 年），哺乳动物大脑中的注意力调节通常表现为神经元反应的增益（即缩放）效应。因此，我们使用缩放算子而不是加法来进行特征细化。我们模块的整个细化阶段为

由于 sigmoid 是单调函数，它不会影响每个神经元的相对重要性。

事实上，除了通道均值 μ 和方差 σ 的计算外，我们模块中的所有运算都是元素向运算。因此，我们可以利用当前机器学习库（如 pytorch）的优势，只需几行代码就能实现我们的模块（式（6）），如图 3 所示。我们在每个区块内的第二个卷积层之后添加了这一实现。总之，我们提出的模块源于神经科学的基本理论，与以往的方法截然不同。此外，我们的模块易于实现，可与现有网络一起使用。

4 Experiments

在本节中，我们将在各种任务中进行一系列实验，以验证 SimAM 的有效性。为了进行公平比较，我们使用 pytorch 重新实现了所有比较方法，并采用了一致的设置。

1. CIFAR Classification

首先，我们在基于 CIFAR（Krizhevsky 等人，2009 年）的图像分类任务中测试了我们的方法。有两个变体：一个有 10 个类别，另一个包含 100 个类别。两个变体都有 50k 张训练图像和 10k 张验证图像。我们的主要重点是验证我们简单而有效的注意力模块。因此，我们将注意力模块纳入了一些成熟的架构，包括 ResNet（He 等人，2016b）、Pre-activation ResNet（He 等人，2016a）、WideResNet（Zagoruyko & Komodakis，2016）和 MobileNetV2（Sandler 等人，2018）1。

Implementation details.我们对所有模型都采用了标准的训练流程（Lee 等人，2015；He 等人，2016b）。具体来说，每幅图像每边填充 4 个像素，然后从填充图像或其水平翻转图像中随机裁剪出 32 × 32 图像用于训练。在评估过程中，所有模型都接受原始图像进行测试。优化由动量为 0.9、批量大小为 128、权重衰减为 0.0005 的 SGD 求解器完成。除 WideResNet 在两台 GPU 上进行优化外，所有网络均在单台 GPU 上进行训练。学习率从 0.1 开始，在 32,000 和 48,000 次迭代时除以 10（在 64,000 次迭代时停止除法）。对 λ 的详细分析将在后文讨论。对于其他模块，包括 SE（Hu 等人，2018b）、CBAM（Woo 等人，2018）、ECA（Wang 等人，2020）和 GC（Cao 等人，2020），我们使用了作者提供

的默认设置下的公开代码。由于随机性，我们报告了每种方法 5 次以上的平均准确率和标准推导。所有结果见表 2