论文译文 Squeeze-and-Excitation Networks

论文译文 Squeeze-and-Excitation Networks

Abstract

卷积神经网络(convolutional neural networks，CNNs)的核心模块是卷积算子，通过在每层局部感受野内融合空间和通道信息，使网络能够构建信息丰富的特征。广泛的先前研究已经调查了这种关系的空间成分，试图通过在其特征层次结构中增强空间编码的质量来加强CNN的代表性。在这项工作中，我们将重点放在通道关系上，并提出了一种新的架构单元，我们称之为"挤压和激励" (SE)块，它通过显式地模拟通道之间的相互依赖来自适应地调整通道的特征响应。我们表明，这些块可以堆叠在一起，形成SENet架构，在不同的数据集上非常有效地泛化。我们进一步证明了SE块以微小的额外计算成本为现有最先进的CNN带来了性能的显著提升。挤激网络构成了我们的ILSVRC 2017分类提交的基础，它赢得了第一名，并将前5位的误差降低到2.251 %，超过了2016年的优胜条目，相对提高了25 %。模型和代码可在https://github.com/hujie-frank/SENet.上获得

1 Introduction

卷积神经网络(CNNs)已被证明是处理广泛的视觉任务的有用模型[1]，[2]，[3]，[4]。在网络的每个卷积层，一组滤波器沿着输入通道表达邻域空间连接模式- -在局部感受野内将空间和通道信息融合在一起。通过将一系列卷积层与非线性激活函数和下采样操作符交织在一起，CNN能够产生捕获分层模式的图像表示，并获得全局理论接受域。计算机视觉研究的一个中心主题是寻找更强大的表示，这些表示只捕获图像中对给定任务最突出的属性，从而提高性能。作为视觉任务中广泛使用的模型家族，新的神经网络架构设计的开发现在代表了这一研究的一个关键前沿。最近的研究表明，通过将学习机制集成到有助于捕捉特征之间空间相关性的网络中，可以加强CNN产生的表示。其中一种由Inception系列架构[5] [6]所推广的方法，将多尺度过程融入网络模块以实现性能的提升。进一步的工作试图更好地建模空间依赖关系[7] [8]，并将空间注意力纳入网络结构[9]。

在本文中，我们研究网络设计的一个不同方面- -通道之间的关系。我们引入了一个新的架构单元，我们称之为Squeeze - and Excite (SE)模块，其目标是通过显式建模卷积特征通道之间的相互依赖来提高网络生成的表示的质量。为此，我们提出了一种机制，允许网络执行特征重新校准，通过这种机制，它可以学习使用全局信息，有选择地强调信息丰富的特征，并抑制不太有用的特征。

SE构件的结构如图1所示。对于任意给定的变换Ftr将输入X映射到特征映射U，其中U∈RH × W × C，例如卷积，我们可以构造一个对应的SE块来执行特征重新校准。特征U首先通过一个挤压操作，该操作通过在其空间维度(H × W)上聚合特征图生成通道描述符。该描述符的功能是产生通道特征响应的全局分布的嵌入，允许来自网络全局感受野的信息被其所有层使用。聚合之后是一个激励操作，该操作采取简单的自门控成像机制的形式，将嵌入作为输入并产生每个通道的调制权重的集合。将这些权重应用到特征图U中，生成SE块的输出，该输出可以直接反馈到网络的后续层。

可以通过简单地堆叠SE块的集合来构造SE网络(SENet)。此外，这些SE块还可以在网络体系结构中的一定深度范围内作为原始块的替换(6.4节)。虽然构建块的模板是通用的，但在整个网络中，它在不同深度上的作用是不同的。在早期层，它以类不可知的方式激发信息特征，加强共享的低层表示。在后面的层中，SE块变得越来越专业化，并以高度类特定的方式响应不同的输入(7.2节)。因此，SE块执行的特征重标定的收益可以通过网络积累。

新型CNN架构的设计和开发是一项艰巨的工程任务，通常需要选择许多新的超参数和层配置。相比之下，SE块的结构非常简单，可以直接在现有的最先进的体系结构中使用，通过将组件替换为相应的SE组件，可以有效地提高性能。SE块在计算上也是轻量级的，只增加了模型复杂性和计算负担。

为了为这些说法提供证据，我们开发了几个SENet，并在ImageNet数据集上进行了广泛的评估[10]。我们还给出了ImageNet以外的结果，表明我们的方法的好处并不局限于特定的数据集或任务。通过使用SENets，我们在ILSVRC 2017分类竞赛中排名第一。我们的最佳模型集成在测试集上取得了2.251 %的top - 5误差。与前一年(top - 5误差为2.991 %)的获胜者相比，这代表了大约25 %的相对改善。

2 Related Work

Deeper architectures.VGGNets [11]和Inception模型[5]表明，增加网络的深度可以显著提高其能够学习的表征质量。批归一化(Batch Normalization，BN) [6]通过调节输入到各层的分布，为深度网络的学习过程增加了稳定性，并产生了更平滑的优化曲面[12]。在这些工作的基础上，深度残差网络证明了通过使用基于身份的跳跃连接[13]，[14]可以学习到更深更强的网络。高速公路网[15]引入了门机制来调节信息沿捷径连接的流动。在这些工作的基础上，网络层间的连接关系得到了进一步的重新制定[16]，[17]这表明深度网络的学习和表示性能有了很大的提高。

一个替代的，但密切相关的研究路线集中在改进网络中包含的计算元素的功能形式的方法上。分组卷积已被证明是增加学习到的变换基数的流行方法[18] [19]。通过多分支卷积[5]，[6]，[20]，[21]可以实现更灵活的算子组合，可以看作是分组算子的自然扩展。在以前的工作中，跨通道相关性通常被映射为新的特征组合，或者独立于空间结构[22]，[23]或者通过使用标准卷积滤波器[24]和1 × 1卷积来联合。这些研究大多集中在降低模型和计算复杂度的目标上，反映了一个假设，即信道关系可以表示为具有局部感受野的实例无关函数的组合。相比之下，我们声称，为单元提供一种利用全局信息显式建模通道间动态、非线性依赖关系的机制可以简化学习过程，并显著增强网络的表示能力。

Algorithmic Architecture Search.除了上述工作之外，还有丰富的研究历史，其目的是放弃手工架构设计，转而寻求自动学习网络的结构。该领域的早期工作大部分是在神经进化社区中进行的，该社区使用进化方法建立了跨网络拓扑搜索的方法[25]，[26]。虽然进化搜索通常需要大量的计算，但已经取得了显著的成功，包括为序列模型找到良好的记忆细胞[27] [28]，以及为大规模图像分类学习复杂的架构[29] [30] [31]。为了减少这些方法的计算负担，基于Lamarckian继承[32]，已经提出了这种方法的有效替代方案。

通过将架构搜索建模为超参数优化、随机搜索[34]和其他更复杂的基于模型的优化技术[35] [36]，也可以用来解决问题。拓扑选择作为通过可能的设计[37]和直接架构预测[38]的路径，[39]已经被提出作为额外的可行架构搜索工具。强化学习[40]、[41]、[42]、[43]、[44]等技术取

得了特别强的效果。SE块可以用作这些搜索算法的原子构建块，并且在并行工作中被证明在这种能力上是非常有效的[45]。

Attention and gating mechanisms.注意力可以解释为将可用的计算资源偏向于信号中最具信息量的成分[46]，[47]，[48]，[49]，[50]，[51]。注意力机制在序列学习[52]、[53]、图像中的定位和理解[9]、[54]、图像描述[55]、[56]和唇语阅读[57]等任务中得到了广泛的应用。在这些应用中，它可以作为一个操作符被整合到一个或多个层之后，代表更高层次的抽象，用于模态之间的适配。一些工作对空间和通道注意的联合使用提供了有趣的研究[58]，[59]。Wang et al。[58]引入了一个强大的基于沙漏模块的主干和掩码注意力机制[8]，它插入在深度残差网络的中间阶段之间。相比之下，我们提出的SE块包含一个轻量级的门控机制，它侧重于通过以计算高效的方式建模通道关系来增强网络的表征能力。

3 Squeeze And Excitation Blocks

挤压激励块是一个计算单元，它可以建立在输入X∈RH′× W′× C′到特征映射U∈RH × W × C的变换Ftr上。在下面的符号中，我们取Ftr为卷积算子，用V = [v1 , v2 , … , vC]表示学习到的滤波核集合，其中vc表示第c个滤波器的参数。然后将输出写成U = [u1 , u2 , … , uC]，其中

这里*表示卷积，vc = [v1c , v2c , … , vC′c]，X = [x1 , x2 , … , xC′]，uc∈RH × W。Vsc是一个2D空间核，代表vc的一个通道，作用在X的相应通道上。为了简化符号，省略了偏置项。由于输出是通过所有通道求和产生的，因此通道依赖隐式地嵌入在vc中，但与滤波器捕获的局部空间相关性纠缠在一起。由卷积建模的信道关系本质上是隐式的和局部的(除了最上层的那些)。我们期望通过显式建模通道间的依赖关系来增强卷积特征的学习，从而使网络能够提高对信息特征的敏感性，这些信息特征可以通过后续的转换来利用。因此，我们希望为它提供获取全局信息的途径，并在输入到下一个变换之前，通过挤压和激励两个步骤重新校准滤波器响应。一个SE模块的结构示意图如图1所示。

1. Squeeze: Global Information Embedding

为了解决通道依赖问题，我们首先考虑输出特征中每个通道的信号。每个学习到的滤波器都使用一个局部感受野进行操作，因此转换输出U的每个单元都无法利用该区域之外的上下文信息。为了缓解这个问题，我们提出将全局空间信息压缩到通道描述符中。这是通过使用全局平均池化来生成通道统计来实现的。形式上，通过U的空间维数H × W收缩生成一个统计量z∈RC，使得z的第c个元素的计算公式为：

讨论。变换U的输出可以理解为局部描述符的集合，其统计量对整幅图像具有表达能力。利用这些信息在以前的特征工程工作中普遍存在[60]，[61]，[62]。我们选择最简单的聚合技术，全局平均池化，注意到更复杂的策略也可以在这里使用。

2. Excitation: Adaptive Recalibration

为了利用挤压操作中聚集的信息，我们接着进行了第二个操作，该操作旨在完全捕获通道依赖。为了实现这一目标，该功能必须满足两个标准：第一，它必须是灵活的(特别是,它必须能够学习通道之间的非线性相互作用)，第二，它必须学会一个非互斥的关系，因为我们希望确保允许多个渠道被强调(而不是强制一热激活)。为了满足这些条件，我们选择使用一个sigmoid激活的简单门控机制：

其中δ是ReLU [63]函数，W1∈RC × Cr × C和W2∈RC × Cr .为了限制模型复杂度和帮助泛化，我们通过在非线性周围形成一个具有两个全连接(FC)层的瓶颈来参数化门机制，即一个具有还原比r (这个参数的选择在6.1节中讨论)的降维层，一个ReLU，然后一个增维层返回变换输出U的通道维度。块的最终输出通过使用激活s重新调整U的大小来获得：

其中' X = [' x1 , ' x2 , ... , ' xC]和Fscale(uc , sc)表示标量sc和特征映射uc∈RH × W之间的通道乘法。

讨论。激励算子将输入的特定描述符z映射为一组信道权重。就此而言，SE块内在地引入了以输入为条件的动态，可以将其视为通道上的自注意力函数，其关系不局限于卷积滤波器响应的局部感受野。

3. Instantiations

通过在每个卷积之后的非线性之后插入，SE块可以集成到标准架构中，比如VGGNet [11]。此外，SE块的灵活性意味着它可以直接应用于标准卷积之外的转换。为了说明这一点，我们通过将SE块合并到下面描述的更复杂体系结构的几个示例中来开发SENet。

我们首先考虑Inception网络SE模块的构造[5]。在这里，我们简单地将转换Ftr看作一个完整的Inception模块(见图2)，通过对体系结构中的每个这样的模块进行这种更改，我们得到一个SE - Inception网络。SE块也可以与残差网络(图3描述了一个SE - ResNet模块的结构示意图)直接使用。这里取SE块变换Ftr为残差模块的非恒等分支。挤压和激发都是在与身份分支求和之前起作用。进一步将SE块与ResNeXt [19]、Inception - ResNet [21]、MobileNet [64]和混洗网[65]集成的变体可以按照类似的方案构建。对于SENet架构的具体实例，表1给出了SE - ResNet - 50和SE - ResNeXt - 50的详细描述。

4 Model And Computational Complexity

为了使所提出的SE模块设计具有实用性，它必须在提高性能和增加模型复杂度之间提供良好的折衷。为了说明与该模块相关的计算负担，我们以ResNet - 50和SE - ResNet - 50的比较为例。对于224 × 224像素的输入图像，ResNet - 50在单个前向通道中需要≥3.86 GFLOPs。每个SE块在压缩阶段使用一个全局平均池化操作，在激励阶段使用两个小的FC层，然后使用一个便宜的通道尺度缩放操作。总体而言，当设置缩减比例r (在3.2节中介绍)为16时，SE - ResNet - 50所需FFLOPs为3.87 G，相对于原始ResNet - 50提升了0.26 %。为了换取这一点额外的计算负担，SE-ResNet-50的准确率超过了ResNet - 50，甚至接近了需要7.58 GFLOPs的更深层ResNet - 101网络(表2)。

在实际应用中，通过ResNet - 50进行一次前向和后向传递需要190 ms，而SE - ResNet - 50的训练小批量数据为256张图片(两种计时都是在带有8个NVIDIA Titan X GPU的服务器上进行的)，则需要209 ms。我们认为这是一个合理的运行时开销，随着全局池化和小的内积操作在流行的GPU库中得到进一步优化，这个开销可能会进一步降低。由于ResNet - 50对于嵌入式设备应用的重要性，我们进一步测试了每个模型的CPU推理时间：对

于224 × 224像素的输入图像，ResNet - 50需要164 ms，而SE - ResNet - 50需要167 ms。我们认为SE块产生的较小的额外计算成本是由其对模型性能的贡献所证明的。

在实际应用中，通过ResNet - 50进行一次前向和后向传递需要190 ms，而SE - ResNet - 50的训练小批量数据为256张图片(两种计时都是在带有8个NVIDIA Titan X GPU的服务器上进行的)，则需要209 ms。我们认为这是一个合理的运行时开销，随着全局池化和小的内积操作在流行的GPU库中得到进一步优化，这个开销可能会进一步降低。由于ResNet - 50对于嵌入式设备应用的重要性，我们进一步测试了每个模型的CPU推理时间：对于224 × 224像素的输入图像，ResNet - 50需要164 ms，而SE - ResNet - 50需要167 ms。我们认为SE块产生的较小的额外计算成本是由其对模型性能的贡献所证明的。

我们接下来考虑所提出的SE块引入的额外参数。这些额外参数仅来自门控机制的两个FC层，因此构成了网络总容量的一小部分。具体地，这些FC层的权重参数引入的总数为：

其中r表示缩减比，S表示级数(阶段是指在一个公共空间维度的特征图上操作的块的集合)，Cs表示输出通道的维数，Ns表示级数s (当FC层使用偏置项时,引入的参数和计算成本通常可以忽略不计)的重复块数。SE - ResNet - 50在ResNet - 50所需的25万个参数之外引入了25万个额外参数，对应增加了10 %。在实际中，这些参数大部分来自网络的最后阶段，其中激励操作是在最大数量的通道上执行的。然而，我们发现，在性能(ImageNet上top - 5误差< 0.1 %)降低相对参数增加到- 4 %的情况下，仅以较小的代价就可以删除SE块的这一相对昂贵的最后阶段，这可能证明在参数使用是关键考虑因素(详见6.4和7.2节)的情况下是有用的。

5 Experiments

在本节中，我们通过实验来研究SE模块在一系列任务、数据集和模型架构中的有效性。

1. Image Classification

为了评估SE块的影响，我们首先在ImageNet 2012数据集[10]上进行实验，该数据集包含128万张训练图像和来自1000个不同类别的50K验证图像。我们在训练集上训练网络，并在验证集上报告top - 1和top - 5错误。

每个基线网络架构及其相应的 SE 对应架构均采用相同的优化方案进行训练。我们遵循标准做法，使用比例和长宽比[5]对数据进行随机裁剪，使其大小为 224 × 224 像素（Inception-ResNet-v2 [21] 和 SE-Inception-ResNet-v2 为 299 × 299），并进行随机水平翻转。通过平均 RGB 通道减法对每张输入图像进行归一化处理。所有模型都在我们的分布式学习系统 ROCS 上进行训练，该系统旨在处理大型网络的高效并行训练。优化采用同步 SGD 算法，动量为 0.9，最小批量为 1024。初始学习率设定为 0.6，每 30 个历元降低 10 倍。使用 [66] 中描述的权重初始化策略，从头开始训练模型 100 个历元。缩减率 r（第 3.2 节）默认设置为 16（除非另有说明）。

在评估模型时，我们对每幅图像进行了中心裁剪，在将其较短的边缘调整为 256 像素后，裁剪出 224 × 224 像素（对于 Inception-ResNet-v2 和 SE-Inception-ResNet-v2，每幅图像的较短边缘调整为 352 像素，裁剪出 299 × 299 像素）。

Network depth.我们首先将 SE-ResNet 与不同深度的 ResNet 架构进行比较，结果见表 2。我们发现，在不同深度下，SE 块都能持续提高性能，而计算复杂度的增加却非常小。值得注意的是，SE-ResNet-50 的单作物 Top-5 验证误差为 6.62%，比 ResNet-50（7.48%）高出 0.86%，接近深度更大的 ResNet-101 网络的性能（top-5 误差为 6.52%），而总计算量仅为其一半（3.87 GFLOPs 对 7.58 GFLOPs）。这种模式在更大的深度上重复出现，SE-ResNet-101（6.07% 的前五名错误率）不仅与更深的 ResNet-152 网络（6.34% 的前五名错误率）相匹配，而且还优于后者 0.27%。需要指出的是，虽然 SE 区块本身增加了深度，但它们的计算效率极高，即使在扩展基础架构的深度实现收益递减时，也能产生良好的回报。此外，我们还发现，在不同的网络深度范围内，收益都是一致的，这表明 SE 区块所带来的改进可能与简单增加基础架构深度所获得的改进相辅相成。

Integration with modern architectures.接下来，我们研究了将 SE 模块与 Inception-ResNet-v2 [21] 和 ResNeXt（使用 32 × 4d 设置）[19] 这两种最新架构整合的效果，这两种架构都在基础网络中引入了额外的计算构件。我们构建了与这些网络等价的 SENet，即 SE-Inception-ResNet-v2 和 SE-ResNeXt（SE-ResNeXt-50 的配置见表 1），并在表 2 中报告了结果。与之前的实验一样，我们观察到在这两种架构中引入 SE 块后，性能有了显著提高。特别是，SE-ResNeXt-50 的前 5 名误差为 5.49%，优于其直接对应的 ResNeXt-50（前 5 名误差为 5.90%）以及更深入的 ResNeXt-101（前 5 名误差为 5.57%），后者的模型参数总数和计算开销几乎是前者的两倍。我们注意到，我们重新实现的 Inception-ResNet-v2 与 [21] 中报告的结果在性能上略有不同。不过，我们也观察到了与 SE 区块效果类似的趋势，发现 SE 对应区块（4.79% 的最高 5 级错误率）比我们重新实现的 Inception-ResNet-v2 基线（5.21% 的最高 5 级错误率）高出 0.42%，也比 [21] 中报告的结果高出 0.42%。

我们还通过对 VGG-16 [11] 和 BN-Inception 架构 [6] 进行实验，评估了 SE 块在非残差网络上运行时的效果。为了便于从头开始训练 VGG-16，我们在每次卷积后添加了批量归一化层。我们对 VGG-16 和 SE-VGG-16 采用了相同的训练方案。比较结果如表 2 所示。与残差基线架构的结果类似，我们发现 SE 块在非残差设置上的性能也有所提高。

为了让我们更深入地了解 SE 模块对这些模型优化的影响，图 4 展示了基线架构及其 SE 对应模块运行的训练曲线示例。我们观察到，在整个优化过程中，SE 模块都能产生稳定的改进。此外，这一趋势在作为基线的一系列网络架构中也相当一致。

Mobile setting.最后，我们考虑了移动优化网络中的两个代表性架构，即 MobileNet [64] 和 ShuffleNet [65]。在这些实验中，我们使用了 256 个小批量，并使用了与 [65] 相同的稍为宽松的数据增强和正则化方法。我们在 8 个 GPU 上使用 SGD 训练模型，模型的动量（设置为 0.9）和初始学习率为 0.1，每当验证损失趋于稳定时，学习率降低 10 倍。整个训练过程需要 400 个历元（使我们能够重现 [65] 的基线性能）。表 3 中报告的结果表明，SE 块能以最小的计算成本提高很大的精确度。

Additional datasets.接下来，我们将研究 SE 区块的优势是否适用于 ImageNet 以外的数据集。我们在 CIFAR-10 和 CIFAR-100 数据集 [70] 上

使用几种流行的基准架构和技术（ResNet-110 [14]、ResNet-164 [14]、WideResNet-16-8 [67]、Shake-Shake [68] 和 Cutout [69]）进行了实验。这些数据集包括 5 万张训练图像和 1 万张 32 × 32 像素 RGB 测试图像，分别标有 10 个和 100 个类别。将 SE 块整合到这些网络中的方法与第 3.3 节中描述的方法相同。每个基线及其对应的 SENet 均采用标准的数据增强策略进行训练[24], [71]。在训练过程中，图像会被随机水平翻转，并在两侧各填充 4 个像素，然后再进行 32 × 32 的随机裁剪。此外，还对平均值和标准差进行归一化处理。训练超参数的设置（如最小批量大小、初始学习率、权重衰减）与原始论文中建议的一致。我们在表 4 中报告了每个基线及其 SENet 在 CIFAR-10 上的性能，在表 5 中报告了在 CIFAR-100 上的性能。我们发现，在每次比较中，SENet 的性能都优于基线架构，这表明 SE 块的优势并不局限于 ImageNet 数据集。

2. Scene Classification

我们还在 Places365-Challenge 数据集[73]上进行了场景分类实验。该数据集包括 800 万张训练图像和 365 个类别中的 36500 张验证图像。相对于分类，场景理解任务为评估模型的泛化和抽象能力提供了另一种方法。这是因为它通常要求模型处理更复杂的数据关联，并对更大程度的外观变化具有鲁棒性。

我们选择使用 ResNet-152 作为评估 SE 区块有效性的强大基线，并遵循 [72] 和 [74] 中描述的训练和评估协议。在这些实验中，模型都是从头开始训练的。我们在表 6 中报告了实验结果，并与之前的工作进行了比较。我们观察到，SE-ResNet-152（前五名误差为 11.01%）的验证误差低于 ResNet-152（前五名误差为 11.61%），这证明 SE 块也能改进场景分类。该 SENet 超越了之前的最先进模型 Places-365-CNN [72]，后者在该任务中的前 5 名误差为 11.48%。

3. Object Detection on COCO

我们使用 COCO 数据集[75]进一步评估了 SE 区块在物体检测任务中的泛化能力。与之前的工作[19]一样，我们使用 minival 协议，即在 80k 训练集和 35k val 子集的联合集上训练模型，然后在剩余的 5k val 子集上进行评估。权重由在 ImageNet 数据集上训练的模型参数初始化。我们使用 Faster R-CNN [4] 检测框架作为评估模型的基础，并遵循 [76] 中描述的

超参数设置（即使用 "2 倍 "学习计划进行端到端训练）。我们的目标是评估用 SE-ResNet 替换物体检测器中的主干架构（ResNet）的效果，以便将性能上的任何变化归因于更好的表示。表 7 报告了使用 ResNet-50、ResNet-101 及其 SE 对应主干架构的对象检测器的验证集性能。在 COCO 的标准 AP 指标上，SE-ResNet-50 比 ResNet-50 高出 2.4%（相对提高 6.3%），在 AP@IoU=0.5 上，SE-ResNet-50 比 ResNet-50 高出 3.1%。在 AP 指标上，SE 块也比更深的 ResNet-101 架构提高了 2.0%（相对提高 5.0%）。总之，这组实验证明了 SE 区块的通用性。诱导的改进可以在广泛的架构、任务和数据集上实现。

4. ILSVRC 2017 Classification Competition

SENets 是我们在 ILSVRC 竞赛中获得第一名的基础。我们的获奖作品包括一个小型 SENets 组合，该组合采用了标准的多尺度和多作物融合策略，在测试集上获得了 2.251% 的前五名误差。作为本次提交的一部分，我们通过将 SE 块与修改后的 ResNeXt [19]（架构详情见附录）进行整合，构建了一个额外的模型 SENet-154。在表 8 中，我们使用标准裁剪尺寸（224 × 224 和 320 × 320）将该模型与之前在 ImageNet 验证集上的工作进行了比较。我们发现，在使用 224 × 224 中心裁剪评估时，SENet-154 的误差率为 18.68%，前五名的误差率为 4.47%，这是目前报道的最强结果。

挑战赛之后，ImageNet 基准取得了很大进展。为了便于比较，我们在表 9 中列出了目前已知的最强结果。最近 [79] 报道了仅使用 ImageNet 数据的最佳性能。这种方法在训练过程中使用强化学习来开发新的数据增强策略，以提高 [31] 所搜索架构的性能。文献[80]报道了使用 ResNeXt-101 32×48d 架构的最佳总体性能。这是通过在约 10 亿张弱标签图像上对其模型进行预训练，并在 ImageNet 上进行微调实现的。更复杂的数据扩充[79]和广泛的预训练[80]所带来的改进，可能与我们对网络架构提出的修改相辅相成。

6 ABLATION STUDY

在本节中，我们将进行消融实验，以更好地了解使用不同配置对 SE 块组件的影响。所有消融实验都是在单机（配备 8 个 GPU）的 ImageNet 数据集上进行的。ResNet-50 被用作骨干架构。我们根据经验发现，在 ResNet

体系结构中，在激励操作中消除 FC 层的偏差有助于通道依赖性建模，因此在接下来的实验中使用了这种配置。数据增强策略沿用了第 5.1 节中描述的方法。为了研究每种变体的性能上限，我们将学习率初始化为 0.1，并持续训练直到验证损失达到峰值2 （总共 300 个历元）。然后将学习率降低 10 倍，并重复这一过程（共三次）。在训练过程中使用标签平滑正则化 [20]。

1. Reduction ratio

公式 5 中引入的缩减率 r 是一个超参数，允许我们改变网络中 SE 区块的容量和计算成本。为了研究这个超参数在性能和计算成本之间的权衡，我们用 SE-ResNet-50 进行了一系列不同 r 值的实验。表 10 中的对比结果表明，性能对一系列缩减率都很稳定。复杂度的增加并不会单调地提高性能，而较小的比率则会显著增加模型的参数大小。设置 r = 16 可以很好地平衡精度和复杂度。在实践中，在整个网络中使用相同的比率可能并不是最佳选择（因为不同的层扮演着不同的角色），因此可以通过调整比率来进一步提高性能，以满足特定基础结构的需求。

2. Squeeze Operator

我们研究了使用全局平均集合而不是全局最大集合作为挤压运算符的意义（因为这 种方法效果很好，所以我们没有考虑更复杂的替代方法）。结果见表 11。虽然最大池化和平均池化都很有效，但平均池化的性能略胜一筹，因此有理由选择它作为挤压操作的基础。不过，我们注意到，SE 区块的性能对特定聚合算子的选择相当稳健。

3. Excitation Operator

接下来，我们将对激发机制的非线性选择进行评估。我们考虑了另外两种选择：ReLU 和 tanh，并尝试用这两种非线性替代 sigmoid。结果见表 12。我们看到，用 tanh 代替 sigmoid 会略微降低性能，而使用 ReLU 则会大幅降低性能，实际上会导致 SE-ResNet-50 的性能低于 ResNet-50 基线。这表明，要使 SE 块有效，必须仔细构建激励算子。

4. Different stages

我们通过在 ResNet-50 中逐级集成 SE 区块来探索 SE 区块在不同阶段的影响。具体来说，我们在中间阶段（阶段 2、阶段 3 和阶段 4）添加了 SE 块，并在表 13 中报告了结果。我们发现，在架构的每个阶段引入 SE 块都能带来性能优势。此外，不同阶段的 SE 区块所带来的收益是互补的，它们可以有效地结合在一起，进一步提高网络性能。

5. Integration strategy

最后，我们进行了一项消融研究，以评估将 SE 区块整合到现有架构中时 SE 区块位置的影响。除了建议的 SE 设计外，我们还考虑了三种变体：(1) SE-PRE 块，其中 SE 块被移到残差单元之前；(2) SE-POST 块，其中 SE 单元被移到与身份分支相加之后（ReLU 之后）；(3) SE-Identity 块，其中 SE 单元被置于与残差单元平行的身份连接上。图 5 展示了这些变体，表 14 报告了每个变体的性能。我们观察到，SE-PRE、SE-Identity 和建议的 SE 模块性能相似，而使用 SE-POST 模块则导致性能下降。该实验表明，只要在分支聚合之前应用 SE 单元，SE 单元所产生的性能改进对其位置相当稳健。

在上述实验中，每个 SE 块都被置于残差单元结构之外。我们还构建了一个设计变体，将 SE 块移至残差单元内部，直接置于 3 × 3 卷积层之后。由于 3 × 3 卷积层的通道数较少，相应的 SE 模块引入的参数数也减少了。表 15 中的比较显示，与标准 SE 模块相比，SE 3×3 变体以更少的参数达到了相当的分类精度。虽然这超出了本文的研究范围，但我们预计，通过针对特定架构调整 SE 块的使用，还能进一步提高效率。

7 Role of SE