Squeeze-and-Excitation Networks

摘要

卷积神经网络建立在卷积运算的基础上，通过融合局部感受野内的空间信息和通道信息来提取信息特征。为了提高网络的表示能力，许多现有的工作已经显示出增强空间编码的好处。在这项工作中，我们专注于通道，并提出了一种新颖的架构单元，我们称之为“Squeeze-and-Excitation” （SE）块，通过显式地建模通道之间的相互依赖关系，自适应地重新校准通道式的特征响应。通过将这些块堆叠在一起，我们证明了我们可以构建SENet架构，在具有挑战性的数据集中可以进行泛化地非常好。关键的是，我们发现SE块以微小的计算成本为现有的最先进的深层架构产生了显著的性能改进。SENets是我们ILSVRC 2017分类提交的基础，它赢得了第一名，并将 top-5 错误率显著减少到2.251%2.251%，相对于2016年的获胜成绩取得了~ 25%~25%的相对改进。

1. 引言

卷积神经网络（CNNs）已被证明是解决各种视觉任务的有效模型[19,23,29,41]。对于每个卷积层，沿着输入通道学习一组滤波器来表达局部空间连接模式。换句话说，期望卷积滤波器通过融合空间信息和信道信息进行信息组合，而受限于局部感受野。通过叠加一系列非线性和下采样交织的卷积层，CNN能够捕获具有全局感受野的分层模式作为强大的图像描述。最近的工作已经证明，网络的性能可以通过显式地嵌入学习机制来改善，这种学习机制有助于捕捉空间相关性而不需要额外的监督。Inception架构推广了一种这样的方法[14,39]，这表明网络可以通过在其模块中嵌入多尺度处理来取得有竞争力的准确度。最近的工作在寻找更好地模型空间依赖[1,27]，结合空间注意力[17]。

与这些方法相反，通过引入新的架构单元，我们称之为“Squeeze-and-Excitation” (SE)块，我们研究了架构设计的一个不同方向——通道关系。我们的目标是通过显式地建模卷积特征通道之间的相互依赖性来提高网络的表示能力。为了达到这个目的，我们提出了一种机制，使网络能够执行特征重新校准，通过这种机制可以学习使用全局信息来选择性地强调信息特征并抑制不太有用的特征。

SE构建块的基本结构如图1所示。对于任何给定的变换$F_{tr}: X \rightarrow U$ Ftr:X→U, $X \in R^{W' \times H' \times C'}, U \in R^{W \times H \times C}$X∈RW′×H′×C′,U∈RW×H×C，(例如卷积或一组卷积)，我们可以构造一个相应的SE块来执行特征重新校准，如下所示。特征U U首先通过squeeze操作，该操作跨越空间维度$W \times H$W×H聚合特征映射来产生通道描述符。这个描述符嵌入了通道特征响应的全局分布，使来自网络全局感受野的信息能够被其较低层利用。这之后是一个excitation操作，其中通过基于通道依赖性的自门机制为每个通道学习特定采样的激活，控制每个通道的激励。然后特征映射U U被重新加权以生成SE块的输出，然后可以将其直接输入到随后的层中。

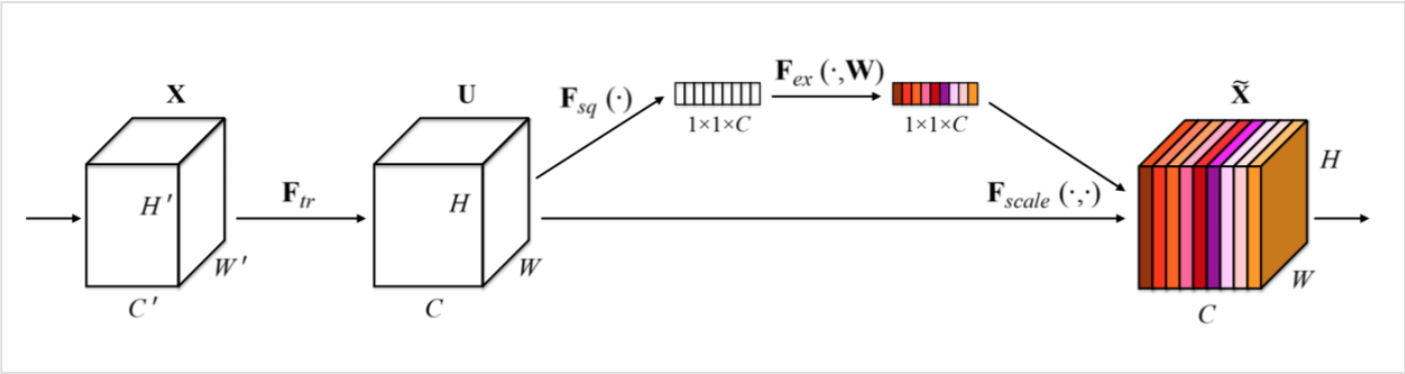

图1. Squeeze-and-Excitation块

SE网络可以通过简单地堆叠SE构建块的集合来生成。SE块也可以用作架构中任意深度的原始块的直接替换。然而，虽然构建块的模板是通用的，正如我们6.3节中展示的那样，但它在不同深度的作用适应于网络的需求。在前

面的层中，它学习以类不可知的方式激发信息特征，增强共享的较低层表示的质量。在后面的层中，SE块越来越专业化，并以高度类特定的方式响应不同的输入。因此，SE块进行特征重新校准的好处可以通过整个网络进行累积。

新CNN架构的开发是一项具有挑战性的工程任务，通常涉及许多新的超参数和层配置的选择。相比之下，上面概述的SE块的设计是简单的，并且可以直接与现有的最新架构一起使用，其卷积层可以通过直接用对应的SE层来替换从而进行加强。另外，如第四节所示，SE块在计算上是轻量级的，并且在模型复杂性和计算负担方面仅稍微增加。为了支持这些声明，我们开发了一些SENets，即SE-ResNet，SE-Inception，SE-ResNeXt和SE-Inception-ResNet，并在ImageNet 2012数据集[30]上对SENets进行了广泛的评估。此外，为了证明SE块的一般适用性，我们还呈现了ImageNet之外的结果，表明所提出的方法不受限于特定的数据集或任务。

使用SENets，我们赢得了ILSVRC 2017分类竞赛的第一名。我们的表现最好的模型集合在测试集上达到了2.251%2.251%的 top-5 错误率。与前一年的获奖者（2.991%2.991%的 top-5 错误率）相比，这表示 ~ 25%~25%的相对改进。我们的模型和相关材料已经提供给研究界。

2. 近期工作

深层架构。大量的工作已经表明，以易于学习深度特征的方式重构卷积神经网络的架构可以大大提高性能。VGGNets[35]和Inception模型[39]证明了深度增加可以获得的好处，明显超过了ILSVRC 2014之前的方法。批标准化（BN）[14]通过插入单元来调节层输入稳定学习过程，改善了通过深度网络的梯度传播，这使得可以用更深的深度进行进一步的实验。He等人[9,10]表明，通过重构架构来训练更深层次的网络是有效的，通过使用基于恒等映射的跳跃连接来学习残差函数，从而减少跨单元的信息流动。最近，网络层间连接的重新表示[5,12]已被证明可以进一步改善深度网络的学习和表征属性。

另一种研究方法探索了调整网络模块化组件功能形式的方法。可以用分组卷积来增加基数（一组变换的大小）[13,43]以学习更丰富的表示。多分支卷积可以解释为这个概念的概括，使得卷积算子可以更灵活的组合[14,38,39,40]。跨通道相关性通常被映射为新的特征组合，或者独立的空间结构[6,18]，或者联合使用标准卷积滤波器[22]和1 × 11×1卷积，然而大部分工作的目标是集中在减少模型和计算复杂度上面。这种方法反映了一个假设，即通道关系可以被表述为具有局部感受野的实例不可知的函数的组合。相比之下，我们声称为网络提供一种机制来显式建模通道之间的动态、非线性依赖关系，使用全局信息可以减轻学习过程，并且显著增强网络的表示能力。

注意力和门机制。从广义上讲，可以将注意力视为一种工具，将可用处理资源的分配偏向于输入信号的信息最丰富的组成部分。这种机制的发展和理解一直是神经科学社区的一个长期研究领域[15,16,28]，并且近年来作为一个强大补充，已经引起了深度神经网络的极大兴趣[20,25]。注意力已经被证明可以改善一系列任务的性能，从图像的定位和理解[3,17]到基于序列的模型[2,24]。它通常结合门功能（例如softmax或sigmoid）和序列技术来实现[11,37]。最近的研究表明，它适用于像图像标题[4,44]和口头阅读[7]等任务，其中利用它来有效地汇集多模态数据。在这些应用中，它通常用在表示较高级别抽象的一个或多个层的顶部，以用于模态之间的适应。高速网络[36]采用门机制来调节快捷连接，使得可以学习非常深的架构。王等人[42]受到语义分割成功的启发，引入了一个使用沙漏模块[27]的强大的trunk-and-mask注意力机制。这个高容量的单元被插入到中间阶段之间的深度残差网络中。相比之下，我们提出的SE块是一个轻量级的门机制，专门用于以计算有效的方式对通道关系进行建模，并设计用于增强整个网络中模块的表示能力。

3. Squeeze-and-Excitation块

Squeeze-and-Excitation块是一个计算单元，可以为任何给定的变换构建：
$F_{tr} : X \rightarrow U, X \in R^{W' \times H' \times C'}, U \in R^{W \times H \times C}$ Ftr:X→U,X∈RW′×H′×C′,U∈RW×H×C。为了简化说明，在接下来的表示中，我们将F_{tr}Ftr看作一个标准的卷积算子。$V = [v_1, v_2, \ldots, v_C]$ V=[v1,v2,...,vC]表示学习到的一组滤波器核，$v_c$vc指的是第cc个滤波器的参数。然后我们可以将F_{tr}Ftr的输出写作
$U = [u_1, u_2, \ldots, u_C]$ U=[u1,u2,...,uC]，其中

$$u_c = v_c * X = \sum_{s=1}^{C'} v_c^s * x^s.$$
uc=vc∗X=∑s=1C′vcs∗xs.

这里∗∗表示卷积，$v_c = [v_c^1, v_c^2, \dots, v_c^{C'}]$ vc=[vc1,vc2,...,vcC′]，$X = [x^1, x^2, \dots, x^{C'}]$ X=[x1,x2,...,xC′]（为了简洁表示，忽略偏置项）。这里v_c^svcs是2D空间核，因此表示$v_c$vc的一个单通道，作用于对应的通道X X。由于输出是通过所有通道的和来产生的，所以通道依赖性被隐式地嵌入到$v_c$vc中，但是这些依赖性与滤波器捕获的空间相关性纠缠在一起。我们的目标是确保能够提高网络对信息特征的敏感度，以便后续转换可以利用这些功能，并抑制不太有用的功能。我们建议通过显式建模通道依赖性来实现这一点，以便在进入下一个转换之前通过两步重新校准滤波器响应，两步为：squeeze和excitation。SE构建块的图如图1所示。

3.1. Squeeze:全局信息嵌入

为了解决利用通道依赖性的问题，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对局部感受野进行操作，因此变换输出U U的每个单元都无法利用该区域之外的上下文信息。在网络较低的层次上其感受野尺寸很小，这个问题变得更严重。

讨论。转换输出U U可以被解释为局部描述子的集合，这些描述子的统计信息对于整个图像来说是有表现力的。特征工程工作中[31,34,45]普遍使用这些信息。我们选择最简单的全局平均池化，同时也可以采用更复杂的汇聚策略。

3.2. Excitation:自适应重新校正

为了利用压缩操作中汇聚的信息，我们接下来通过第二个操作来全面捕获通道依赖性。为了实现这个目标，这个功能必须符合两个标准：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交互）；第二，它必须学习一个非互斥的关系，因为独热激活相反，这里允许强调多个通道。为了满足这些标准，我们选择采用一个简单的门机制，并使用sigmoid激活：

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2\delta(W_1 z))$$
s=Fex(z,W)=σ(g(z,W))=σ(W2δ(W1z))

，其中δδ是指ReLU[26]函数，$W_1 \in \mathbb{R}^{\frac{C}{r}\times C}$W1∈RCr×C和$W_2 \in \mathbb{R}^{C\times\frac{C}{r}}$W2∈RC×Cr。为了限制模型复杂度和辅助泛化，我们通过在非线性周围形成两个全连接（FC）层的瓶颈来参数化门机制，即降维层参数为W_1W1，降维比例为rr（我们把它设置为16，这个参数选择在6.3节中讨论），一个ReLU，然后是一个参数为W_2W2的升维层。块的最终输出通过重新调节带有激活的变换输出U U得到：

$$\tilde{x}_c = F_{scale}(u_c, s_c) = s_c \cdot u_c$$
x~c=Fscale(uc,sc)=sc·uc

其中$\tilde{X} = [\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_C]$ X~=[x~1,x~2,...,x~C]和$F_{scale}(u_c, s_c)$ Fscale(uc,sc)指的是特征映射$u_c \in \mathbb{R}^{W\times H}$uc∈RW×H和标量$s_c$sc之间的对应通道乘积。

讨论。激活作为适应特定输入描述符z z的通道权重。在这方面，SE块本质上引入了以输入为条件的动态特性，有助于提高特征辨别力。

3.3. 模型：SE-Inception和SE-ResNet

SE块的灵活性意味着它可以直接应用于标准卷积之外的变换。为了说明这一点，我们通过将SE块集成到两个流行的网络架构系列Inception和ResNet中来开发SENets。通过将变换F_{tr}Ftr看作一个整体的Inception模块（参见图

2），为Inception网络构建SE块。通过对架构中的每个模块进行更改，我们构建了一个SE-Inception网络。

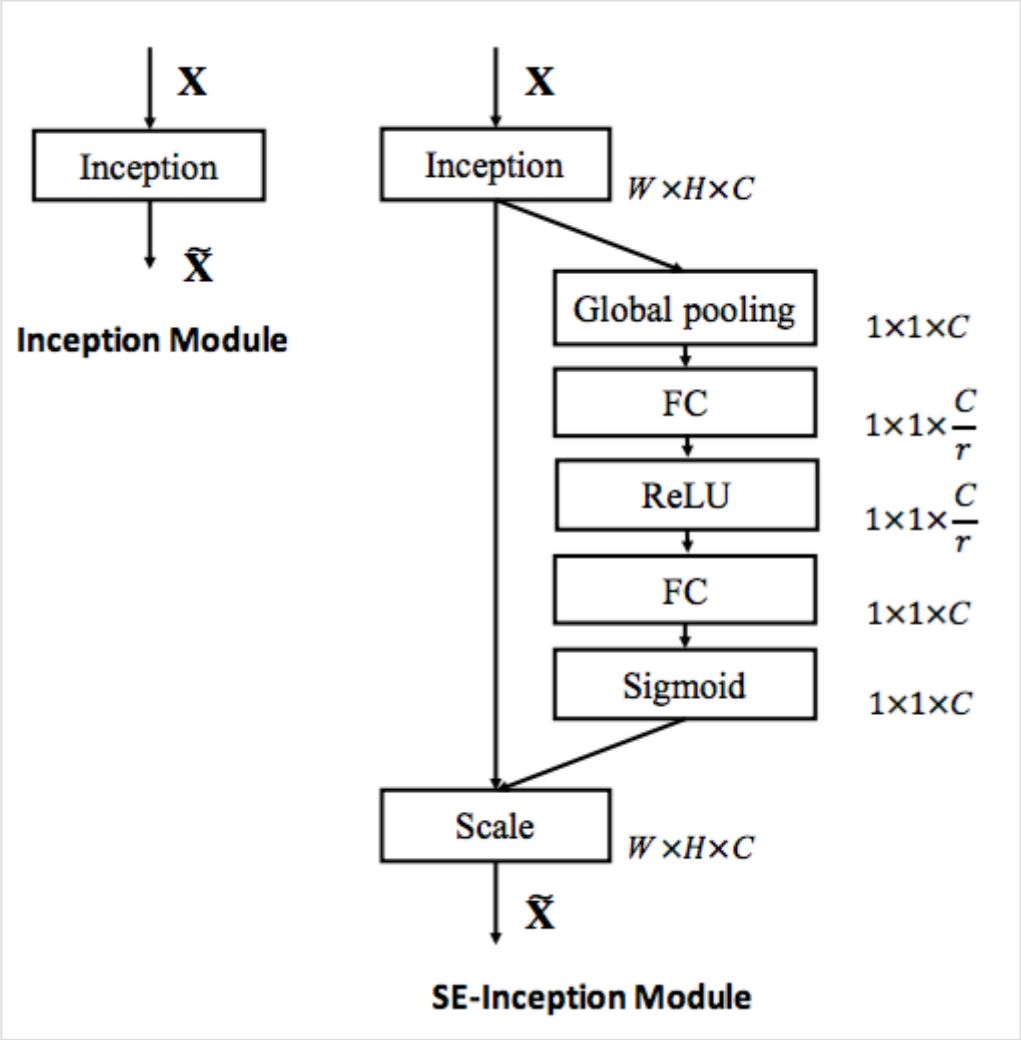

图2。最初的Inception模块架构(左)和SE-Inception模块(右)。

残留网络及其变种已经证明在学习深度表示方面非常有效。我们开发了一系列的SE块，分别与ResNet[9]，ResNeXt[43]和Inception-ResNet[38]集成。图3描述了SE-ResNet模块的架构。在这里，SE块变换F_{tr}Ftr被认为是残差模块的非恒等分支。压缩和激励都在恒等分支相加之前起作用。

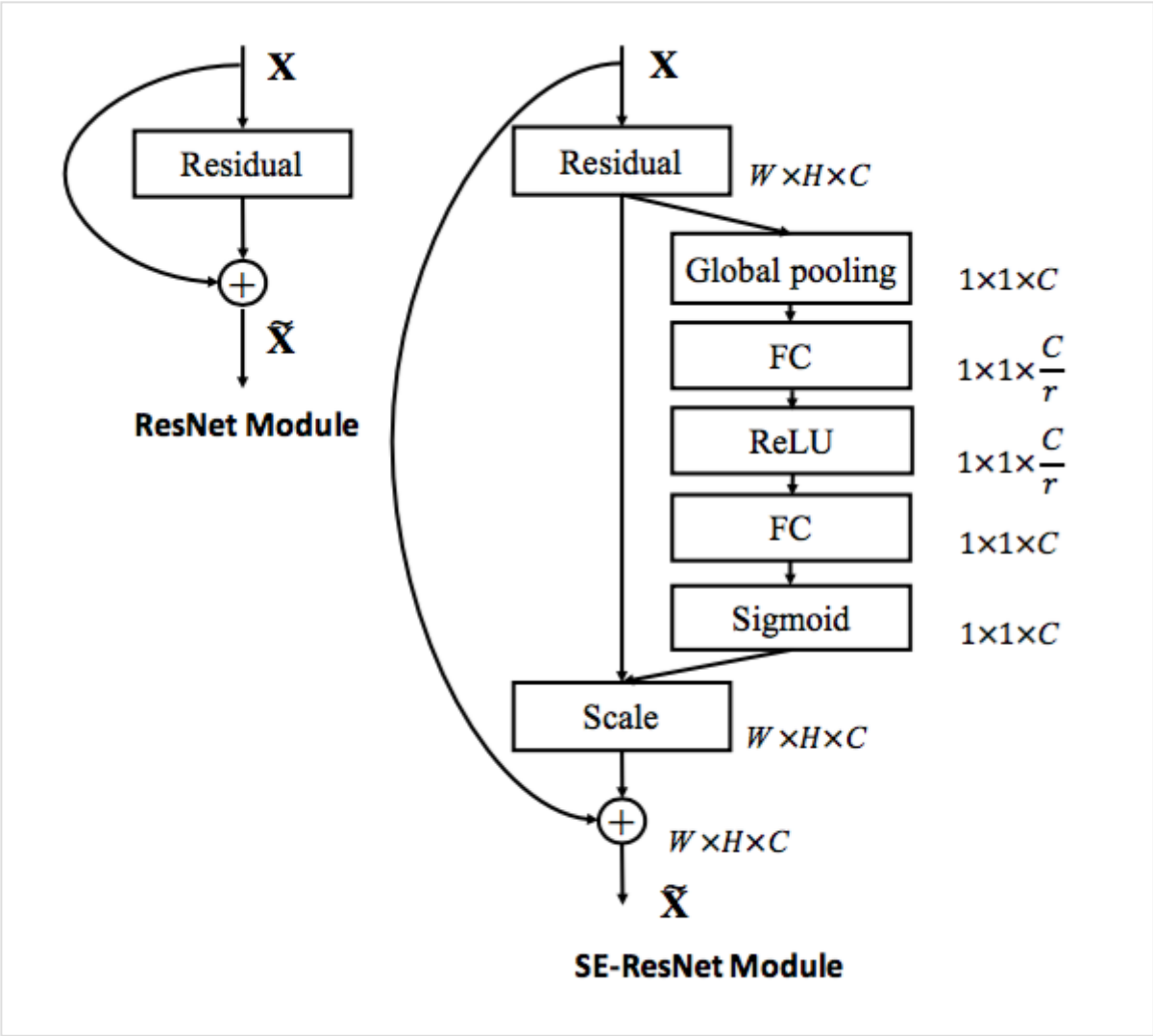

图3。 最初的Residual模块架构(左)和SE-ResNet模块架构(右)。

4. 模型和计算复杂度

SENet通过堆叠一组SE块来构建。实际上，它是通过用原始块的SE对应部分（即SE残差块）替换每个原始块（即残差块）而产生的。我们在表1中描述了SE-ResNet-50和SE-ResNeXt-50的架构。

Output size	ResNet-50	SE-ResNet-50	SE-ResNeXt-50 (32×4d)
112×112	$conv, 7\times7, 64$, stride 2		
56×56	$max\ pool, 3\times3$, stride 2		
	$\begin{bmatrix} conv, 1\times1, 64 \\ conv, 3\times3, 64 \\ conv, 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} conv, 1\times1, 64 \\ conv, 3\times3, 64 \\ conv, 1\times1, 256 \\ fc, [16, 256] \end{bmatrix} \times 3$	$\begin{bmatrix} conv, 1\times1, 128 \\ conv, 3\times3, 128 \quad C=32 \\ conv, 1\times1, 256 \\ fc, [16, 256] \end{bmatrix} \times 3$
28×28	$\begin{bmatrix} conv, 1\times1, 128 \\ conv, 3\times3, 128 \\ conv, 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} conv, 1\times1, 128 \\ conv, 3\times3, 128 \\ conv, 1\times1, 512 \\ fc, [32, 512] \end{bmatrix} \times 4$	$\begin{bmatrix} conv, 1\times1, 256 \\ conv, 3\times3, 256 \quad C=32 \\ conv, 1\times1, 512 \\ fc, [32, 512] \end{bmatrix} \times 4$
14×14	$\begin{bmatrix} conv, 1\times1, 256 \\ conv, 3\times3, 256 \\ conv, 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} conv, 1\times1, 256 \\ conv, 3\times3, 256 \\ conv, 1\times1, 1024 \\ fc, [64, 1024] \end{bmatrix} \times 6$	$\begin{bmatrix} conv, 1\times1, 512 \\ conv, 3\times3, 512 \quad C=32 \\ conv, 1\times1, 1024 \\ fc, [64, 1024] \end{bmatrix} \times 6$
7×7	$\begin{bmatrix} conv, 1\times1, 512 \\ conv, 3\times3, 512 \\ conv, 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} conv, 1\times1, 512 \\ conv, 3\times3, 512 \\ conv, 1\times1, 2048 \\ fc, [128, 2048] \end{bmatrix} \times 3$	$\begin{bmatrix} conv, 1\times1, 1024 \\ conv, 3\times3, 1024 \quad C=32 \\ conv, 1\times1, 2048 \\ fc, [128, 2048] \end{bmatrix} \times 3$
1×1	$global\ average\ pool$, 1000-d fc, $softmax$		

表1。(左)ResNet-50，(中)SE-ResNet-50，(右)具有32 × 4d32×4d模板的SE-ResNeXt-50。在括号内列出了残差构建块特定参数设置的形状和操作，并且在外部呈现了一个阶段中堆叠块的数量。fc后面的内括号表示SE模块中两个全连接层的输出维度。

在实践中提出的SE块是可行的，它必须提供可接受的模型复杂度和计算开销，这对于可伸缩性是重要的。为了说明模块的成本，作为例子我们比较了ResNet-50和SE-ResNet-50，其中SE-ResNet-50的精确度明显优于ResNet-50，接近更深的ResNet-101网络（如表2所示）。对于224 × 224224×224像素的输入图像，ResNet-50单次前向传播需要~ ~ 3.86 GFLOP。每个SE块利用压缩阶段的全局平均池化操作和激励阶段中的两个小的全连接层，接下来是廉价的通道缩放操作。总的来说，SE-ResNet-50需要~ ~ 3.87 GFLOP，相对于原始的ResNet-50只相对增加了0.26%0.26%。

	original		re-implementation			SENet		
	top-1 err.	top-5 err.	top-1 err.	top-5 err.	GFLOPs	top-1 err.	top-5 err.	GFLOPs
ResNet-50 [9]	24.7	7.8	24.80	7.48	3.86	$23.29_{(1.51)}$	$6.62_{(0.86)}$	3.87
ResNet-101 [9]	23.6	7.1	23.17	6.52	7.58	$22.38_{(0.79)}$	$6.07_{(0.45)}$	7.60
ResNet-152 [9]	23.0	6.7	22.42	6.34	11.30	$21.57_{(0.85)}$	$5.73_{(0.61)}$	11.32
ResNeXt-50 [43]	22.2	-	22.11	5.90	4.24	$21.10_{(1.01)}$	$5.49_{(0.41)}$	4.25
ResNeXt-101 [43]	21.2	5.6	21.18	5.57	7.99	$20.70_{(0.48)}$	$5.01_{(0.56)}$	8.00
BN-Inception [14]	25.2	7.82	25.38	7.89	2.03	$24.23_{(1.15)}$	$7.14_{(0.75)}$	2.04
Inception-ResNet-v2 [38]	19.9†	4.9†	20.37	5.21	11.75	$19.80_{(0.57)}$	$4.79_{(0.42)}$	11.76

表2。ImageNet验证集上的单裁剪图像错误率（%）和复杂度比较。 `original` 列是指原始论文中报告的结果。为了进行公平比较，我们重新训练了基准模型，并在 `re-implementation` 列中报告分数。 `SENet` 列是指已添加SE块后对应的架构。括号内的数字表示与重新实现的基准数据相比的性能改善。†表示该模型已经在验证集的非黑名单子集上进行了评估（在[38]中有更详细的讨论），这可能稍微改善结果。

在实践中，训练的批数据大小为256张图像，ResNet-50的一次前向传播和反向传播花费190190 ms，而SE-ResNet-50则花费209209 ms（两个时间都在具有88个NVIDIA Titan X GPU的服务器上执行）。我们认为这是一个合理的开销，因为在现有的GPU库中，全局池化和小型内积操作的优化程度较低。此外，由于其对嵌入式设备应用的重要性，我们还对每个模型的CPU推断时间进行了基准测试：对于224 × 224224×224像素的输入图像，ResNet-50花费了164164ms，相比之下，SE-ResNet-5050花费了167167ms。SE块所需的小的额外计算开销对于其对模型性能的贡献来说是合理的（在第6节中详细讨论）。

接下来，我们考虑所提出的块引入的附加参数。所有附加参数都包含在门机制的两个全连接层中，构成网络总容量的一小部分。更确切地说，引入的附加参数的数量由下式给出：

$$\frac{2}{r}\sum_{s=1}^{S} N_s \cdot C_s^2$$

2r∑s=1SNs·Cs2

其中rr表示减少比率（我们在所有的实验中将rr设置为1616），SS指的是阶段数量（每个阶段是指在共同的空间维度的特征映射上运行的块的集合），C_sCs表示阶段ss的输出通道的维度，N_s Ns表示重复的块编号。总的来说，SE-ResNet-50在ResNet-50所要求的~ ~2500万参数之外引入了~ ~250万附加参数，相对增加了~ 10%~10%的参数总数量。这些附加参数中的大部分来自于网络的最后阶段，其中激励在最大的通道维度上执行。然而，我们发现SE块相对昂贵的最终阶段可以在性能的边际成本（ImageNet数据集上< 0.1%<0.1%的top-1错误率）上被移除，将相对参数增加减少到~ 4%~4%，这在参数使用是关键考虑的情况下可能证明是有用的。

5. 实现

在训练过程中，我们遵循标准的做法，使用随机大小裁剪[39]到224 × 224224×224像素（299 × 299299×299用于Inception-ResNet-v2[38]和SE-Inception-ResNet-v2）和随机的水平翻转进行数据增强。输入图像通过通道减去均值进行归一化。另外，我们采用[32]中描述的数据均衡策略进行小批量采样，以补偿类别的不均匀分布。网络在我们的分布式学习系统“ROCS”上进行训练，能够处理大型网络的高效并行训练。使用同步SGD进行优化，动量为0.9，小批量数据的大小为1024（在4个服务器的每个GPU上分成32张图像的子批次，每个服务器包含8个GPU）。初始学习率设为0.6，每30个迭代周期减少10倍。使用[8]中描述的权重初始化策略，所有模型都从零开始训练100个迭代周期。

6. 实验

在这一部分，我们在ImageNet 2012数据集上进行了大量的实验[30]，其目的是：首先探索提出的SE块对不同深度基础网络的影响；其次，调查它与最先进的网络架构集成后的能力，旨在公平比较SENets和非SENets，而不是推动性能。接下来，我们将介绍ILSVRC 2017分类任务模型的结果和详细信息。此外，我们在Places365-Challenge场景分类数据集[48]上进行了实验，以研究SENets是否能够很好地泛化到其它数据集。最后，我们研究激励的作用，并根据实验现象给出了一些分析。

6.1. ImageNet分类

ImageNet 2012数据集包含来自1000个类别的128万张训练图像和5万张验证图像。我们在训练集上训练网络，并在验证集上使用中心裁剪图像评估来报告 top-1 和 top-5 错误率，其中每张图像短边首先归一化为256，然后从每张图像中裁剪出224 × 224224×224个像素，（对于Inception-ResNet-v2和SE-Inception-ResNet-v2，每幅图像的短边首先归一化到352，然后裁剪出299 × 299299×299个像素）。

网络深度。我们首先将SE-ResNet与一系列标准ResNet架构进行比较。每个ResNet及其相应的SE-ResNet都使用相同的优化方案进行训练。验证集上不同网络的性能如表2所示，表明SE块在不同深度上的网络上计算复杂度极小增加，始终提高性能。

值得注意的是，SE-ResNet-50实现了单裁剪图像6.62%6.62%的 top-5 验证错误率，超过了ResNet-50（7.48%7.48%）0.86%0.86%，接近更深的ResNet-101网络（6.52%6.52%的 top-5 错误率），且只有ResNet-101一半的计算开销（3.873.87GFLOPs vs. 7.587.58 GFLOPs）。这种模式在更大的深度上重复，SE-ResNet-101（6.07%6.07%的 top-5 错误率）不仅可以匹配，而且超过了更深的ResNet-152网络（6.34%6.34%的 top-5 错误率）。图4分别描绘了SE-ResNets和ResNets的训练和验证曲线。虽然应该注意SE块本身增加了深度，但是它们的计算效率极高，即使在扩展的基础架构的深度达到收益递减的点上也能产生良好的回报。而且，我们看到通过对各种不同深度的训练，性能改进是一致的，这表明SE块引起的改进可以与增加基础架构更多深度结合使用。

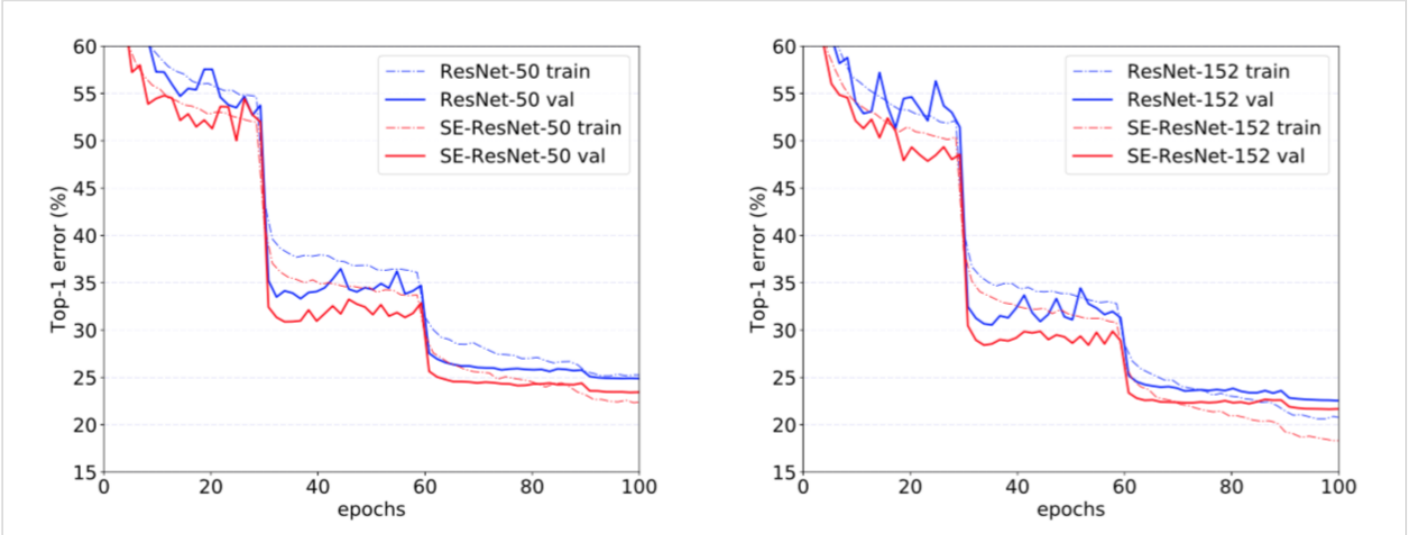

图4。ImageNet上的训练曲线。(左)：ResNet-50和SE-ResNet-50；(右)：ResNet-152和SE-ResNet-152。

与现代架构集成。接下来我们将研究SE块与另外两种最先进的架构Inception-ResNet-v2[38]和ResNeXt[43]的结合效果。Inception架构将卷积模块构造为分解滤波器的多分支组合，反映了Inception假设[6]，可以独立映射空间相关性和跨通道相关性。相比之下，ResNeXt体架构断言，可以通过聚合稀疏连接（在通道维度中）卷积特征的组合来获得更丰富的表示。两种方法都在模块中引入了先前结构化的相关性。我们构造了这些网络的SENet等价物，SE-Inception-ResNet-v2和SE-ResNeXt（表1给出了SE-ResNeXt-50（32 × 4d32×4d）的配置）。像前面的实验一样，原始网络和它们对应的SENet网络都使用相同的优化方案。

表2中给出的结果说明在将SE块引入到两种架构中会引起显著的性能改善。尤其是SE-ResNeXt-50的 top-5 错误率是5.49%5.49%，优于于它直接对应的ResNeXt-50（5.90%5.90%的 top-5 错误率）以及更深的ResNeXt-101（5.57%5.57%的 top-5 错误率），这个模型几乎有两倍的参数和计算开销。对于Inception-ResNet-v2的实验，我们猜测可能是裁剪策略的差异导致了其报告结果与我们重新实现的结果之间的差距，因为它们的原始图像大小尚未在[38]中澄清，而我们从相对较大的图像（其中较短边被归一化为352）中裁剪出299 × 299299×299大小的区域。SE-Inception-ResNet-v2（4.79%4.79%的 top-5 错误率）比我们重新实现的Inception-ResNet-v2（5.21%5.21%的 top-5 错误率）要低0.42%0.42%（相对改进了8.1%8.1%）也优于[38]中报告的结果。每个网络的优化曲线如图5所示，说明了在整个训练过程中SE块产生了一致的改进。

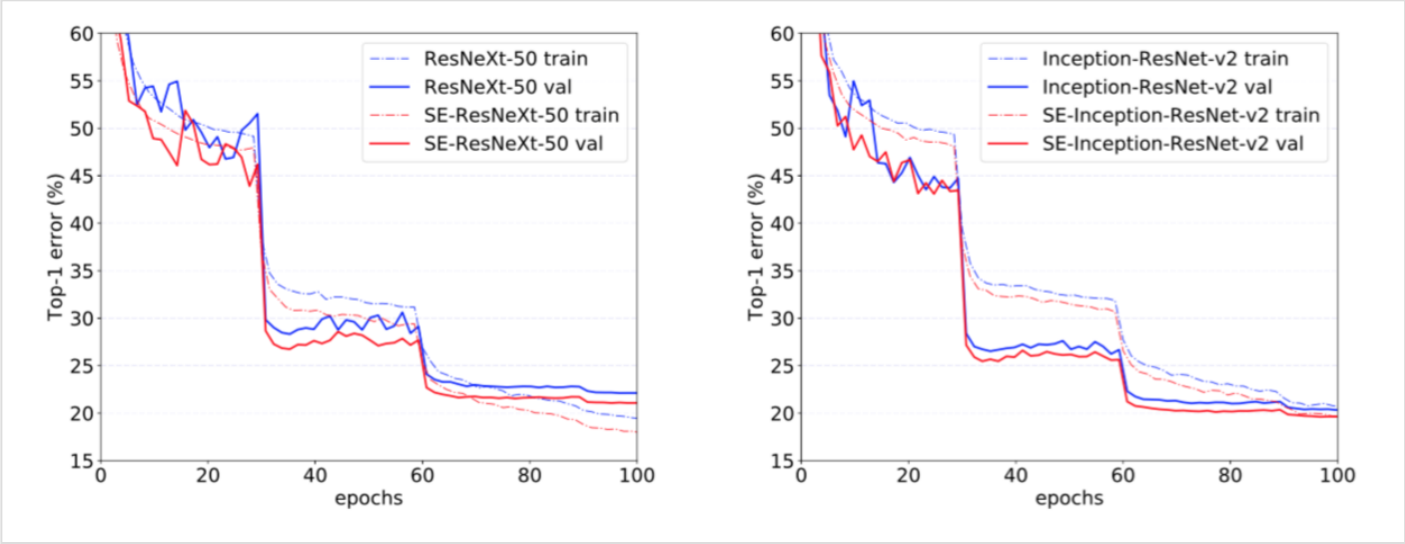

图5。ImageNet的训练曲线。(左): ResNeXt-50和SE-ResNeXt-50；(右)：Inception-ResNet-v2和SE-Inception-ResNet-v2。

最后，我们通过对BN-Inception架构[14]进行实验来评估SE块在非残差网络上的效果，该架构在较低的模型复杂度下提供了良好的性能。比较结果如表2所示，训练曲线如图6所示，表现出的现象与残差架构中出现的现象一样。尤其是与BN-Inception 7.89%7.89%的错误率相比，SE-BN-Inception获得了更低7.14%7.14%的 top-5 错误。这些实验表明SE块引起的改进可以与多种架构结合使用。而且，这个结果适用于残差和非残差基础。

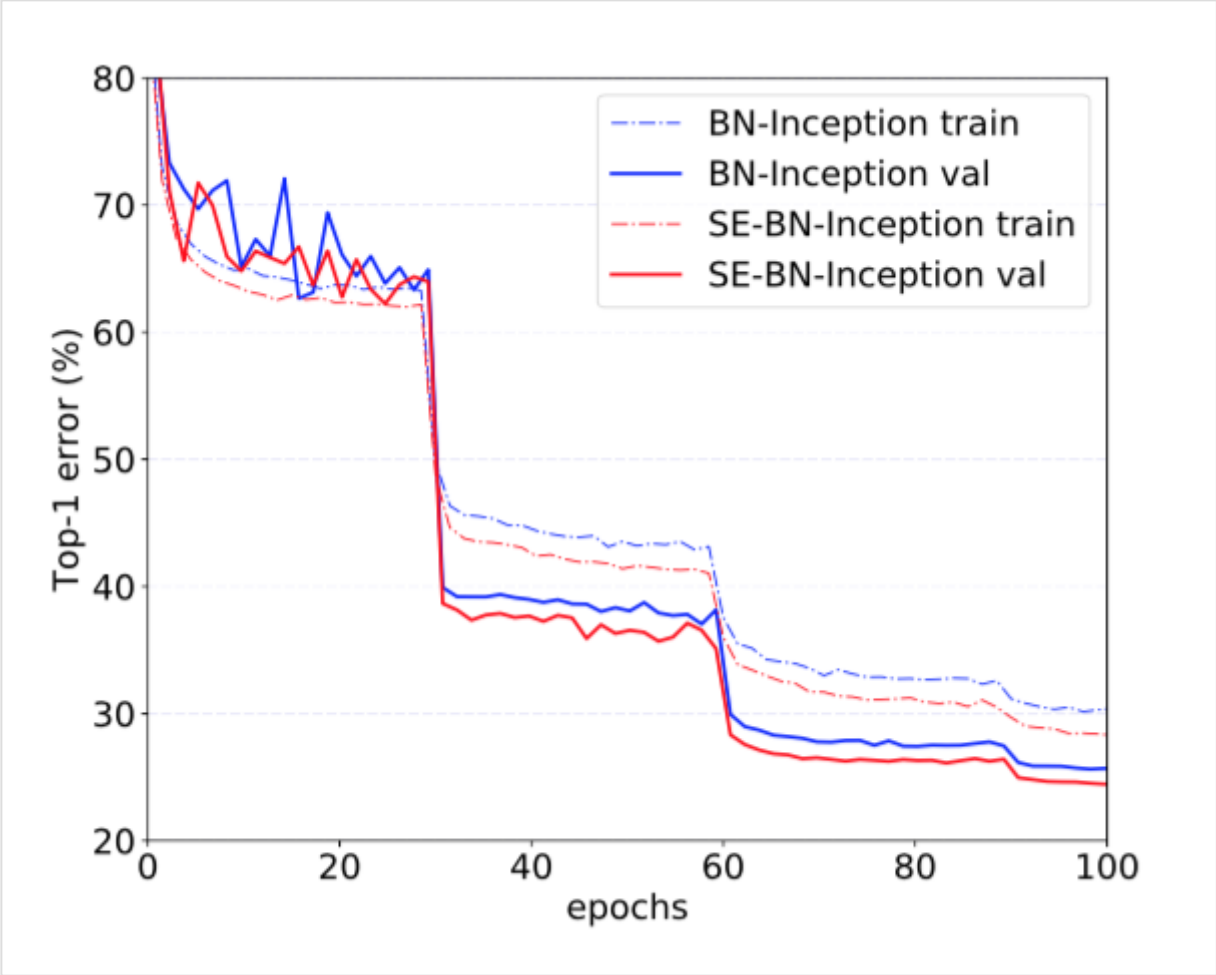

图6。BN-Inception和SE-BN-Inception在ImageNet上的训练曲线。

ILSVRC 2017分类竞赛的结果。ILSVRC[30]是一个年度计算机视觉竞赛，被证明是图像分类模型发展的沃土。ILSVRC 2017分类任务的训练和验证数据来自ImageNet 2012数据集，而测试集包含额外的未标记的10万张图像。为了竞争的目的，使用 `top-5` 错误率度量来对输入条目进行排序。

SENets是我们在挑战中赢得第一名的基础。我们的获胜输入由一小群SENets组成，它们采用标准的多尺度和多裁剪图像融合策略，在测试集上获得了2.251%2.251%的 `top-5` 错误率。这个结果表示在2016年获胜输入（2.99%2.99%的 `top-5` 错误率）的基础上相对改进了~ 25%~25%。我们的高性能网络之一是将SE块与修改后的ResNeXt[43]集成在一起构建的（附录A提供了这些修改的细节）。在表3中我们将提出的架构与最新的模型在ImageNet验证集上进行了比较。我们的模型在每一张图像使用224 × 224224×224中间裁剪评估（短边首先归一化到256）取得了18.68%18.68%的 `top-1` 错误率和4.47%4.47%的 `top-5` 错误率。为了与以前的模型进行公平的比较，我们也提供了320 × 320320×320的中心裁剪图像评估，在 `top-1` (17.28%17.28%)和 `top-5` (3.79%3.79%)的错误率度量中获得了最低的错误率。

	224 × 224		320 × 320 / 299 × 299	
	top-1 err.	top-5 err.	top-1 err.	top-5 err.
ResNet-152 [9]	23.0	6.7	21.3	5.5
ResNet-200 [10]	21.7	5.8	20.1	4.8
Inception-v3 [40]	-	-	21.2	5.6
Inception-v4 [38]	-	-	20.0	5.0
Inception-ResNet-v2 [38]	-	-	19.9	4.9
ResNeXt-101 (64 × 4d) [43]	20.4	5.3	19.1	4.4
DenseNet-161 (k = 48) [12]	22.2	-	-	-
Very Deep PolyNet [47]	-	-	18.71	4.25
DPN-131 [5]	19.93	5.12	18.55	4.16
SENet	**18.68**	**4.47**	**17.28**	**3.79**

表3。最新的CNNs在ImageNet验证集上单裁剪图像的错误率。测试的裁剪图像大小是224 × 224224×224和[10]中的320 × 320320×320/299 × 299299×299。与前面的工作相比，我们提出的模型SENet表现出了显著的改进。

6.2. 场景分类

ImageNet数据集的大部分由单个对象支配的图像组成。为了在更多不同的场景下评估我们提出的模型，我们还在Places365-Challenge数据集[48]上对场景分类进行评估。该数据集包含800万张训练图像和365个类别的36500张验证图像。相对于分类，场景理解的任务可以更好地评估模型泛化和处理抽象的能力，因为它需要捕获更复杂的数据关联以及对更大程度外观变化的鲁棒性。

我们使用ResNet-152作为强大的基线来评估SE块的有效性，并遵循[33]中的评估协议。表4显示了针对给定任务训练ResNet-152模型和SE-ResNet-152的结果。具体而言，SE-ResNet-152（11.01%11.01%的 top-5 错误率）取得了比ResNet-152（11.61%11.61%的 top-5 错误率）更低的验证错误率，证明了SE块可以在不同的数据集上表现良好。这个SENet超过了先前的最先进的模型Places-365-CNN [33]，它在这个任务上有11.48%11.48%的 top-5 错误率。

	top-1 err.	top-5 err.
Places-365-CNN [33]	41.07	11.48
ResNet-152 (ours)	41.15	11.61
SE-ResNet-152	**40.37**	**11.01**

表4。Places365验证集上的单裁剪图像错误率(%)。

6.3. 分析和讨论

减少比率。公式（5）中引入的减少比率rr是一个重要的超参数，它允许我们改变模型中SE块的容量和计算成本。为了研究这种关系，我们基于SE-ResNet-50架构进行了一系列不同rr值的实验。表5中的比较表明，性能并没有随着容量的增加而单调上升。这可能是使SE块能够过度拟合训练集通道依赖性的结果。尤其是我们发现设置 $r = 16$r=16在精度和复杂度之间取得了很好的平衡，因此我们将这个值用于所有的实验。

Ratio r	top-1 err.	top-5 err.	model size (MB)
4	23.21	6.63	137
8	23.19	6.64	117
16	23.29	6.62	108
32	23.40	6.77	103
original	24.80	7.48	98

表5。 ImageNet验证集上单裁剪图像的错误率(%)和SE-ResNet-50架构在不同减少比率rr下的模型大小。这里 `original` 指的是ResNet-50。

激励的作用。虽然SE块从经验上显示出其可以改善网络性能，但我们也想了解自门激励机制在实践中是如何运作的。为了更清楚地描述SE块的行为，本节我们研究SE-ResNet-50模型的样本激活，并考察它们在不同块不同类别下的分布情况。具体而言，我们从ImageNet数据集中抽取了四个类，这些类表现出语义和外观多样性，即金鱼，哈巴狗，刨和悬崖（图7中显示了这些类别的示例图像）。然后，我们从验证集中为每个类抽取50个样本，并计算每个阶段最后的SE块中50个均匀采样通道的平均激活（紧接在下采样之前），并在图8中绘制它们的分布。作为参考，我们也绘制所有1000个类的平均激活分布。

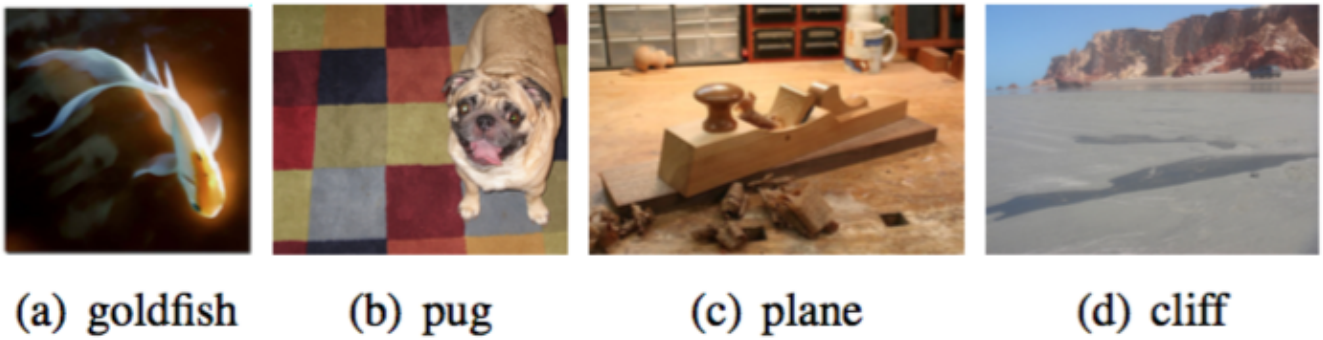

图7。ImageNet中四个类别的示例图像。

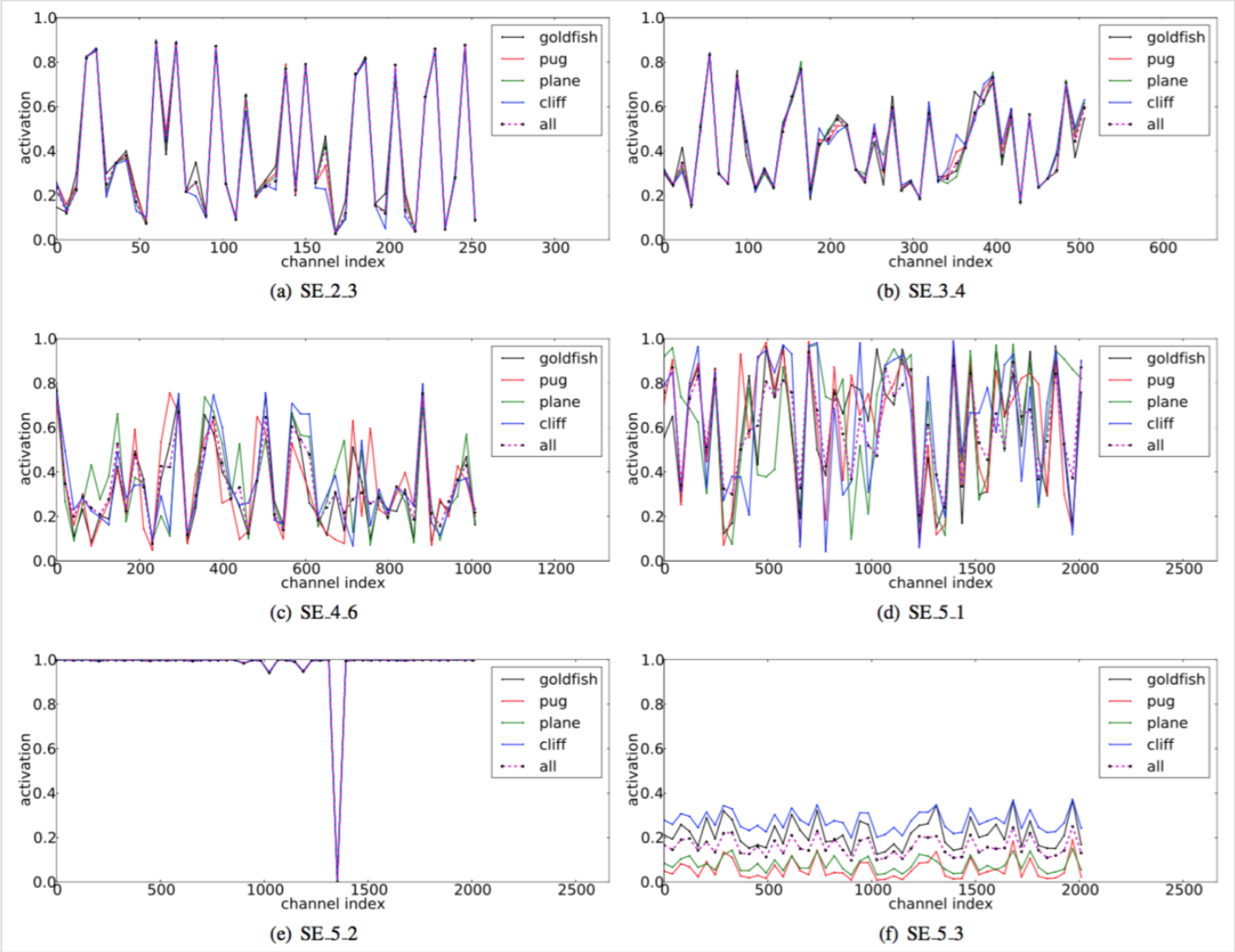

图8。SE-ResNet-50不同模块在ImageNet上由Excitation引起的激活。模块名为“SE stageID blockID”。

我们对SENets中Excitation的作用提出以下三点看法。首先，不同类别的分布在较低层中几乎相同，例如，SE_2_3。这表明在网络的最初阶段特征通道的重要性很可能由不同的类别共享。然而有趣的是，第二个观察结果是在更大的深度，每个通道的值变得更具类别特定性，因为不同类别对特征的判别性值具有不同的偏好。SE_4_6和SE_5_1。这两个观察结果与以前的研究结果一致[21,46]，即低层特征通常更普遍（即分类中不可知的类别），而高层特征具有更高的特异性。因此，表示学习从SE块引起的重新校准中受益，其自适应地促进特征提取和专业化到所需要的程度。最后，我们在网络的最后阶段观察到一个有些不同的现象。SE_5_2呈现出朝向饱和状态的有趣趋势，其中大部分激活接近于1，其余激活接近于0。在所有激活值取1的点处，该块将成为标准残差块。在网络的末端SE_5_3中（在分类器之前紧接着是全局池化），类似的模式出现在不同的类别上，尺度上只有轻微的变化（可以通过分类器来调整）。这表明，SE_5_2和SE_5_3在为网络提供重新校准方面比前面的块更不重要。这一发现与第四节实证研究的结果是一致的，这表明，通过删除最后一个阶段的SE块，总体参数数量可以显著减少，性能只有一点损失（<0.1%0.1%的 `top-1` 错误率）。

7. 结论

在本文中，我们提出了SE块，这是一种新颖的架构单元，旨在通过使网络能够执行动态通道特征重新校准来提高网络的表示能力。大量实验证明了SENets的有效性，其在多个数据集上取得了最先进的性能。此外，它们还提供了一些关于以前架构在建模通道特征依赖性上的局限性的洞察，我们希望可能证明SENets对其它需要强判别性特征的任务是有用的。最后，由SE块引起的特征重要性可能有助于相关领域，例如为了压缩的网络修剪。

致谢。我们要感谢Andrew Zisserman教授的有益评论，并感谢Samuel Albanie的讨论并校订论文。我们要感谢Chao Li在训练系统内存优化方面的贡献。Li Shen由国家情报总监(ODNI)，先期研究计划中心（IARPA）资助，

合同号为2014-14071600010。本文包含的观点和结论属于作者的观点和结论，不应理解为ODNI，IARPA或美国政府明示或暗示的官方政策或认可。尽管有任何版权注释，美国政府有权为政府目的复制和分发重印。