

图像分类经典论文翻译汇总：[\[翻译汇总\]](#)

翻译 pdf 文件下载：[\[下载地址\]](#)

此版为中英文对照版，纯中文版请稳步：[\[SENet 中英文对照版\]](#)

Squeeze-and-Excitation Networks

挤压和激励网络

Jie Hu* Momenta hujie@momenta.ai	Li Shen* University of Oxford lishen@robots.ox.ac.uk	Gang Sun* Momenta sungang@momenta.ai
--	--	--

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表明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）**模块**，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 模块以微小的计算成本为现有最先进的深层架构产生了显著的性能改进。SENet 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%，相对于 2016 年的获胜团队取得了约 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 $F_{tr}: X \rightarrow U$ $F_{tr}: X \rightarrow U$, $X \in \mathbb{R}^{W' \times H' \times C'}$, $U \in \mathbb{R}^{W \times H \times C}$ $X \in \mathbb{R}^{W' \times H' \times C'}$, $U \in \mathbb{R}^{W \times H \times C}$, (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 首先通过 squeeze 操作，该操作跨越空间维度 $W \times H$ 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 **UU** 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。

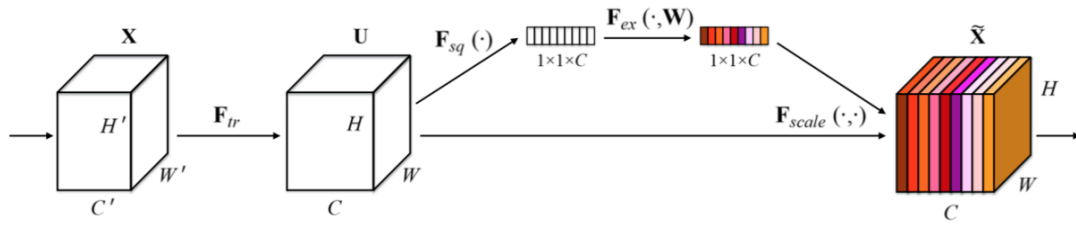


图 1. Squeeze-and-Excitation 块

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 **SE**Nets，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% 的 top-5 错误率。与前一年的获奖者 (2.991% 的 top-5 错误率) 相比, 这表示 ~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 $1 \times 1 \times 1$ 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： $\text{Ftr}: X \rightarrow U, X \in \mathbb{R}^{W' \times H' \times C'}, U \in \mathbb{R}^{W \times H \times C}$
 $\text{Ftr}: X \rightarrow U, X \in \mathbb{R}^{W' \times H' \times C'}, U \in \mathbb{R}^{W \times H \times C}$ 。为了简化说明，在接下来的表示中，我们将 Ftr 看作一个标准的卷积算子。 $V = [v_1, v_2, \dots, v_C]$ 表示学习到的一组滤波器核， v_c 指的是第 c 个滤波器的参数。然后我们可以将 Ftr 的输出写作 $U = [u_1, u_2, \dots, u_C]$ ，其中

$$u_c = v_c * X = \sum_{s=1}^{C'} v_{cs} * x_s. u_c = v_c * X = \sum_{s=1}^{C'} v_{cs} * x_s.$$

这里 $*$ 表示卷积， $v_c = [v_{1c}, v_{2c}, \dots, v_{C'c}]$ ， $X = [x_1, x_2, \dots, x_{C'}]$ （为了简洁表示，忽略偏置项）。这里 v_{cs} 是 2D 空间核，因此表示 v_c 的一个单通道，作用于对应的通道 x_s 。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 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 UU 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。

为了减轻这个问题，我们提出将全局空间信息压缩成一个通道描述符。这是通过使用全局平均池化生成通道统计实现的。形式上，统计 $z \in \mathbb{R}^C$ 是通过在空间维度 $W \times H \times H$ 上收缩 UU 生成的，其中 z 的第 cc 个元素通过下式计算：

$$z_c = \text{Fsq}(u_c) = \frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H u_c(i, j).$$

讨论。转换输出 UU 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：

$$s = \text{Fex}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \delta(W_1 z))$$

，其中 δ 是指 ReLU[26]函数， $W_1 \in \mathbb{R}^{C_r \times C}$ 和 $W_2 \in \mathbb{R}^{C \times C_r}$ 。为了限制模型复杂度和辅助泛化，我们通过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_1 ，降维比例为 r （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_2 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 得到：

$$\tilde{x}^c = \text{Fscale}(u^c, s^c) = s^c \cdot u^c$$

其中 $\tilde{X} = [\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^C]$ 和 $\text{Fscale}(u^c, s^c)$ 指的是特征映射 $u^c \in \mathbb{R}^{W \times H}$ 和标量 s^c 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 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 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。

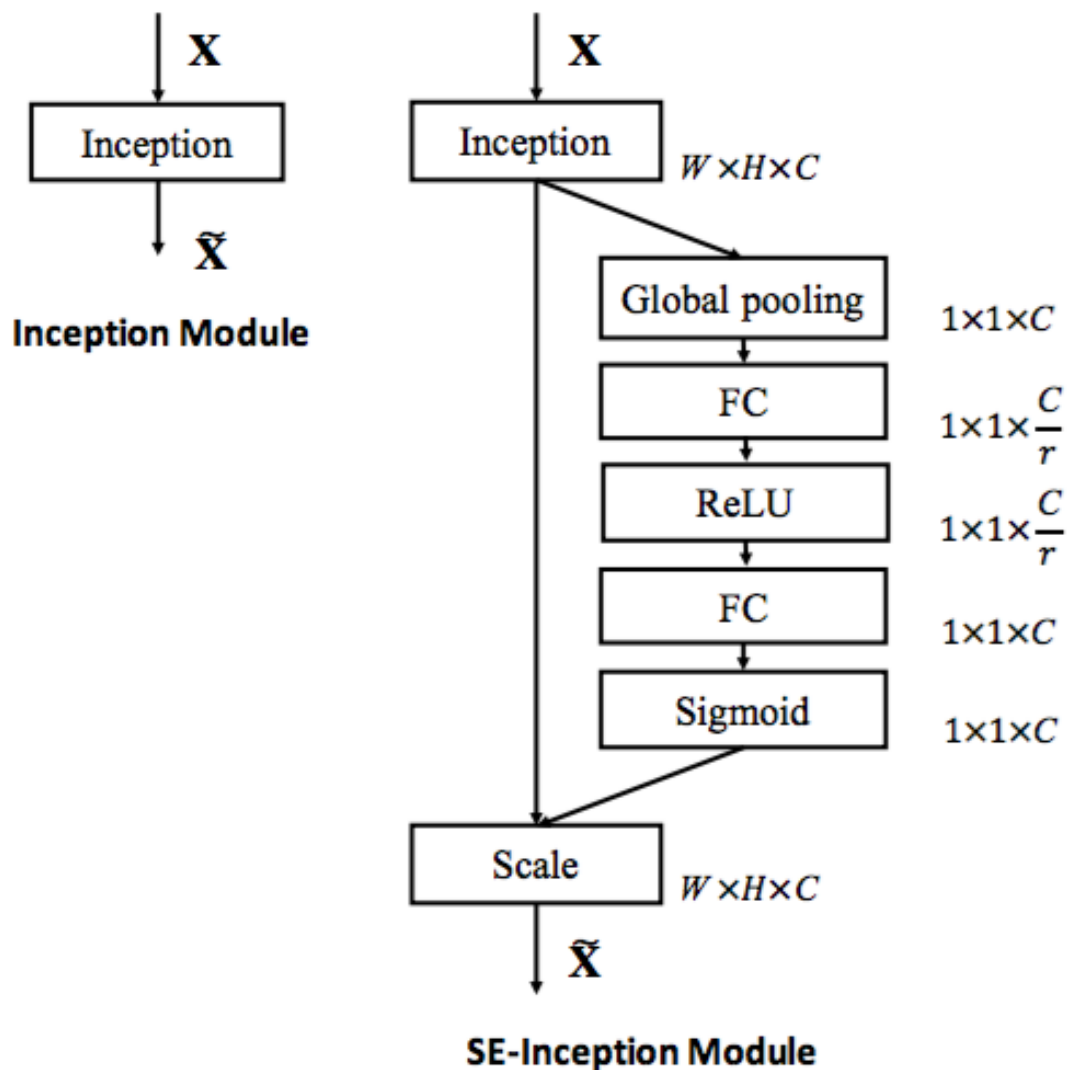


图2.最初的 Inception 模块架构(左)和 SE-Inception 模块架构(右)。

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 $F_{tr}F_{tr}$ 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。

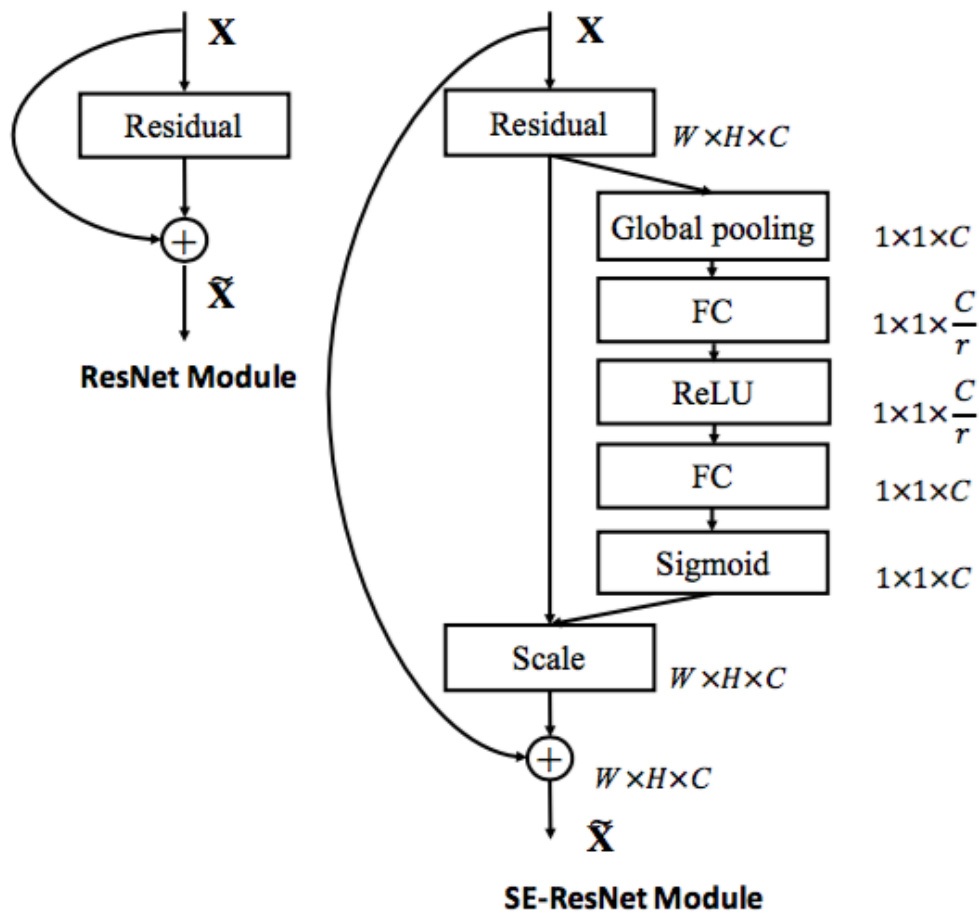


图3。最初的 Residual 模块架构(左)和 SE-ResNet 模块架构(右)。

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	$\text{conv}, 7 \times 7, 64, \text{stride } 2$ $\text{max pool}, 3 \times 3, \text{stride } 2$		
56×56	$\begin{bmatrix} \text{conv}, 1 \times 1, 64 \\ \text{conv}, 3 \times 3, 64 \\ \text{conv}, 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} \text{conv}, 1 \times 1, 64 \\ \text{conv}, 3 \times 3, 64 \\ \text{conv}, 1 \times 1, 256 \\ \text{fc}, [16, 256] \end{bmatrix} \times 3$	$\begin{bmatrix} \text{conv}, 1 \times 1, 128 \\ \text{conv}, 3 \times 3, 128 \\ \text{conv}, 1 \times 1, 256 \\ \text{fc}, [16, 256] \end{bmatrix} \begin{matrix} C = 32 \\ \times 3 \end{matrix}$
28×28	$\begin{bmatrix} \text{conv}, 1 \times 1, 128 \\ \text{conv}, 3 \times 3, 128 \\ \text{conv}, 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} \text{conv}, 1 \times 1, 128 \\ \text{conv}, 3 \times 3, 128 \\ \text{conv}, 1 \times 1, 512 \\ \text{fc}, [32, 512] \end{bmatrix} \times 4$	$\begin{bmatrix} \text{conv}, 1 \times 1, 256 \\ \text{conv}, 3 \times 3, 256 \\ \text{conv}, 1 \times 1, 512 \\ \text{fc}, [32, 512] \end{bmatrix} \begin{matrix} C = 32 \\ \times 4 \end{matrix}$
14×14	$\begin{bmatrix} \text{conv}, 1 \times 1, 256 \\ \text{conv}, 3 \times 3, 256 \\ \text{conv}, 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} \text{conv}, 1 \times 1, 256 \\ \text{conv}, 3 \times 3, 256 \\ \text{conv}, 1 \times 1, 1024 \\ \text{fc}, [64, 1024] \end{bmatrix} \times 6$	$\begin{bmatrix} \text{conv}, 1 \times 1, 512 \\ \text{conv}, 3 \times 3, 512 \\ \text{conv}, 1 \times 1, 1024 \\ \text{fc}, [64, 1024] \end{bmatrix} \begin{matrix} C = 32 \\ \times 6 \end{matrix}$
7×7	$\begin{bmatrix} \text{conv}, 1 \times 1, 512 \\ \text{conv}, 3 \times 3, 512 \\ \text{conv}, 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} \text{conv}, 1 \times 1, 512 \\ \text{conv}, 3 \times 3, 512 \\ \text{conv}, 1 \times 1, 2048 \\ \text{fc}, [128, 2048] \end{bmatrix} \times 3$	$\begin{bmatrix} \text{conv}, 1 \times 1, 1024 \\ \text{conv}, 3 \times 3, 1024 \\ \text{conv}, 1 \times 1, 2048 \\ \text{fc}, [128, 2048] \end{bmatrix} \begin{matrix} C = 32 \\ \times 3 \end{matrix}$
1×1	global average pool, 1000-d fc, softmax		

表 1。(左)ResNet-50, (中)SE-ResNet-50, (右)具有 $32 \times 4d$ 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 $224 \times 224 \times 224$ 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%。

	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 209209ms（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 $224 \times 224 \times 224$ 像素的输入图像，ResNet-50 花费了 164164ms，相比之下，SE-ResNet-50 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：

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

其中 r 表示减少比率（我们在所有的实验中将 r 设置为 1616）， S 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_s 表示阶段 s 的输出通道的维度， N_s 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 $\sim 10\%$ 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 $< 0.1\%$ 的 top-1 错误率）上被移除，将相对参数增加减少到 $\sim 4\%$ ，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 224×224 像素（ 299×299 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 $224 \times 224 \times 224$ 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 $299 \times 299 \times 299$ 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(3.873.87 GFLOPs 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。

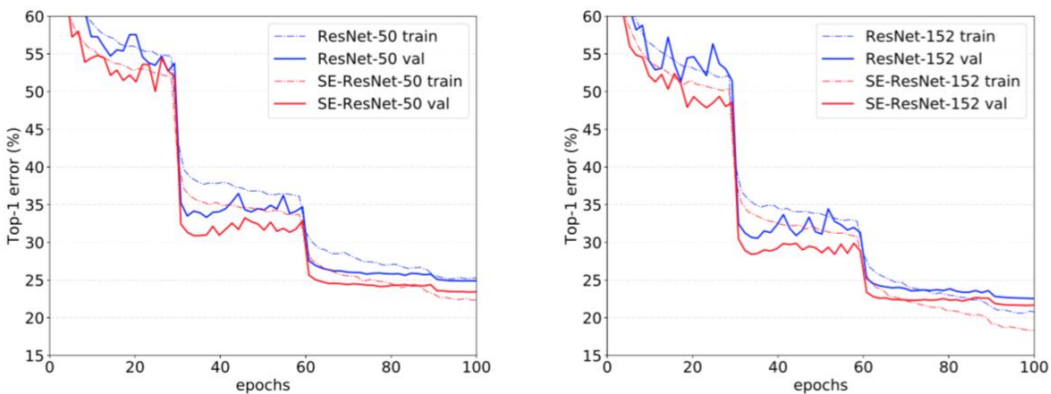


图 4。ImageNet 上的训练曲线。(左): ResNet-50 和 SE-ResNet-50; (右): ResNet-152 和 SE-ResNet-152。

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($32 \times 4d \times 32 \times 4d$)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 $299 \times 299 \times 299$ 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。

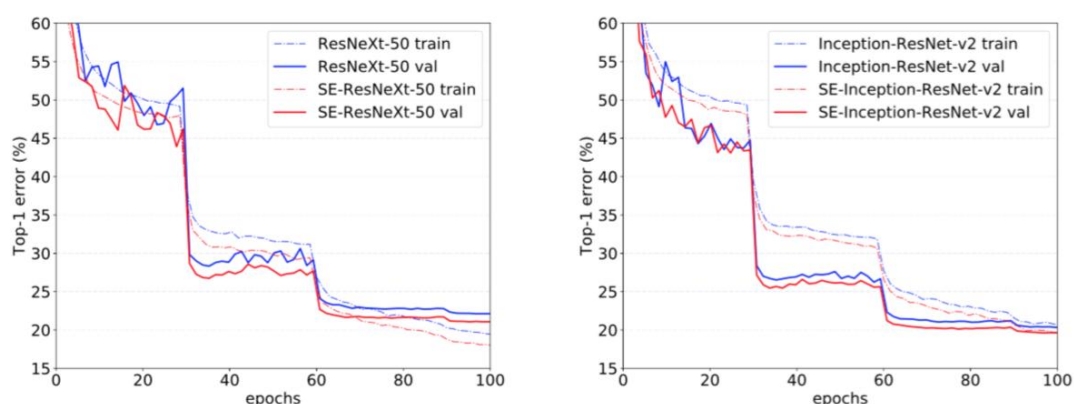


图 5。ImageNet 的训练曲线。(左): ResNeXt-50 和 SE-ResNeXt-50; (右): Inception-ResNet-v2 和 SE-Inception-ResNet-v2。

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。

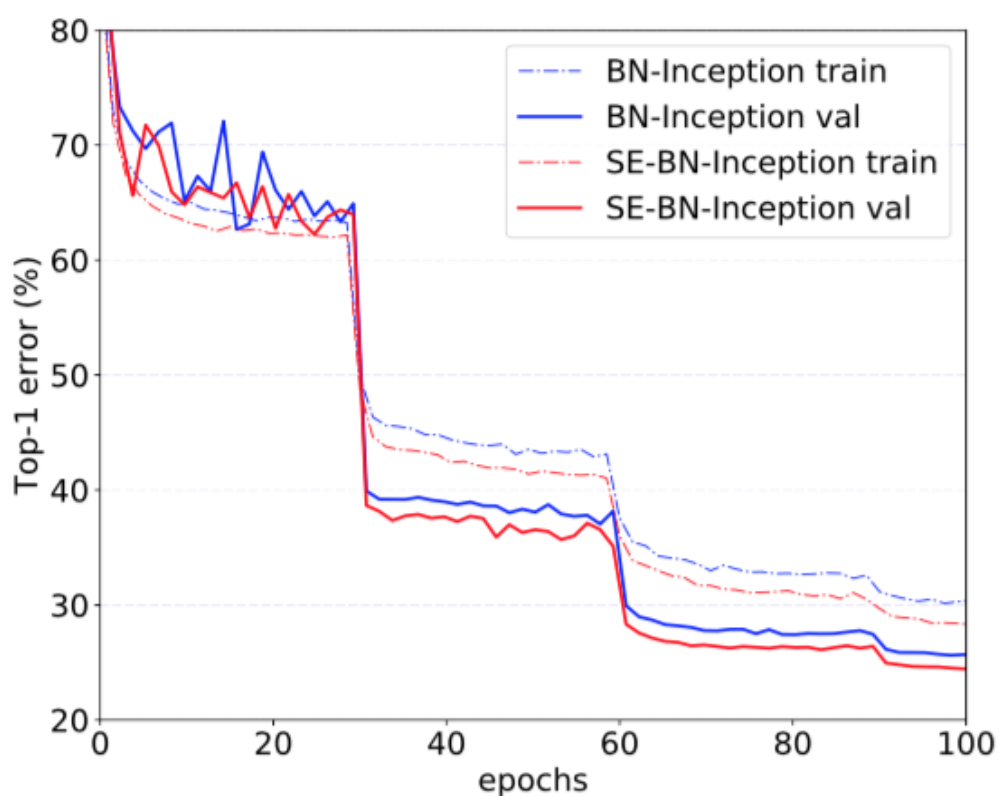


图 6。BN-Inception 和 SE-BN-Inception 在 ImageNet 上的训练曲线。

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。

SE Nets 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 SE Nets 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% 的 top-5 错误率。这个结果表示在 2016 年获胜输入（2.99% 的 top-5 错误率）的基础上相对改进了 ~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 224×224 中间裁剪评估（短边首先归一化到 256）取得了 18.68% 的 top-1 错误率和 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 320×320 的中心裁剪图像评估，在 top-1 (17.28%) 和 top-5 (3.79%) 的错误率度量中获得了最低的错误率。

	224×224		$320 \times 320 / 299 \times 299$	
	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 ($64 \times 4d$) [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 224×224 和 [10] 中的 320×320 和 $320/299 \times 299$ 。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

$r=16$ 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 r 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。

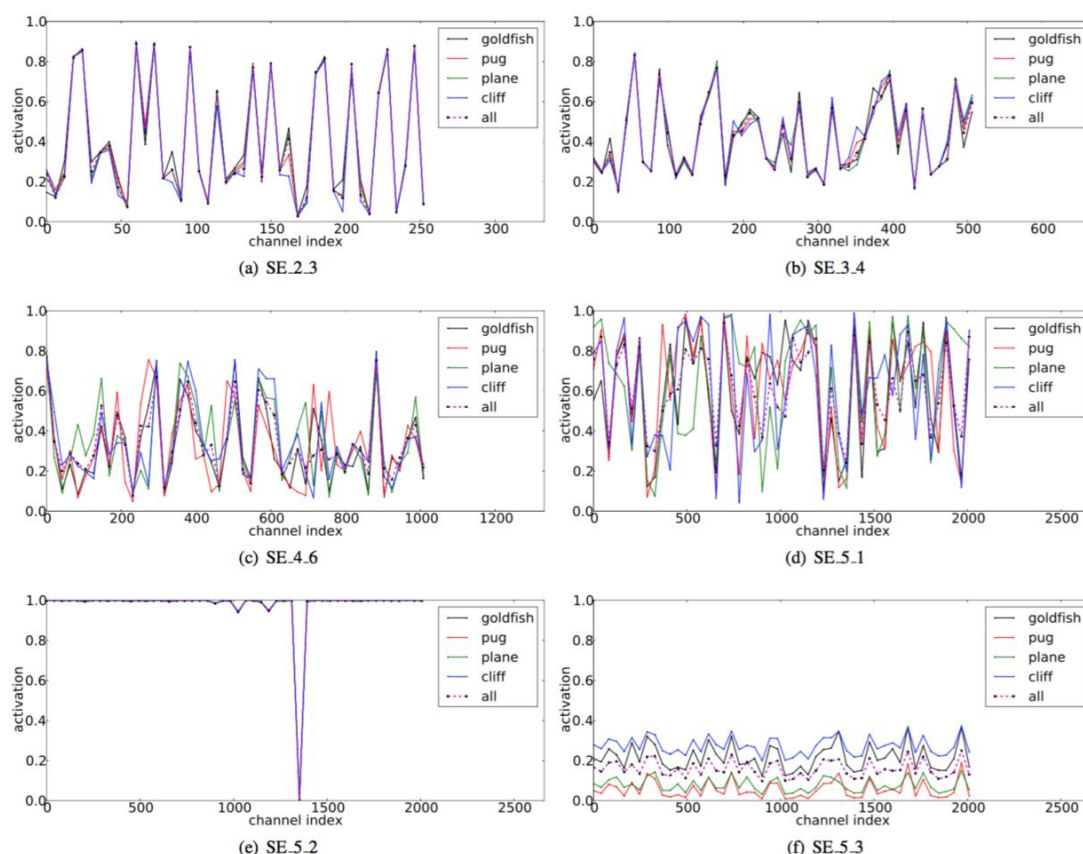


图 8。SE-ResNet-50 不同模块在 ImageNet 上由 Excitation 引起的激活。模块名为“SE stageID blockID”。

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（ $<0.1\%$ 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。

参考文献

- [1] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In CVPR, 2016.
- [2] T. Bluche. Joint line segmentation and transcription for end-to-end handwritten paragraph recognition. In NIPS, 2016.
- [3] C.Cao, X.Liu, Y.Yang, Y.Yu, J.Wang, Z.Wang, Y.Huang, L. Wang, C. Huang, W. Xu, D. Ramanan, and T. S. Huang. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In ICCV, 2015.
- [4] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T. Chua. SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning. In CVPR, 2017.
- [5] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng. Dual path networks. arXiv:1707.01629, 2017.
- [6] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In CVPR, 2017.
- [7] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman. Lip reading sentences in the wild. In CVPR, 2017.
- [8] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In ICCV, 2015.
- [9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016.
- [10] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In ECCV, 2016.
- [11] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, 1997.
- [12] G. Huang, Z. Liu, K. Q. Weinberger, and L. Maaten. Densely connected convolutional networks. In CVPR, 2017.
- [13] Y. Ioannou, D. Robertson, R. Cipolla, and A. Criminisi. Deep roots: Improving CNN efficiency with hierarchical filter groups. In CVPR, 2017.

- [14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 2015.
- [15] L. Itti and C. Koch. Computational modelling of visual attention. *Nature reviews neuroscience*, 2001.
- [16] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE TPAMI*, 1998.
- [17] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In NIPS, 2015.
- [18] M. Jaderberg, A. Vedaldi, and A. Zisserman. Speeding up convolutional neural networks with low rank expansions. In BMVC, 2014.
- [19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In NIPS, 2012.
- [20] H. Larochelle and G. E. Hinton. Learning to combine foveal glimpses with a third-order boltzmann machine. In NIPS, 2010.
- [21] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In ICML, 2009.
- [22] M. Lin, Q. Chen, and S. Yan. Network in network. arXiv:1312.4400, 2013.
- [23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015.
- [24] A. Miech, I. Laptev, and J. Sivic. Learnable pooling with context gating for video classification. arXiv:1706.06905, 2017.
- [25] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu. Recurrent models of visual attention. In NIPS, 2014.
- [26] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In ICML, 2010.
- [27] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In ECCV, 2016.
- [28] B. A. Olshausen, C. H. Anderson, and D. C. V. Essen. A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *Journal of Neuroscience*, 1993.
- [29] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015.
- [30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *IJCV*, 2015.
- [31] J. Sanchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. RR-8209, INRIA, 2013.
- [32] L. Shen, Z. Lin, and Q. Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In ECCV, 2016.
- [33] L. Shen, Z. Lin, G. Sun, and J. Hu. Places401 and places365 models. <https://github.com/lishen-shirley/Places2-CNNs>, 2016.
- [34] L. Shen, G. Sun, Q. Huang, S. Wang, Z. Lin, and E. Wu. Multi-level discriminative dictionary learning with application to large scale image classification. *IEEE TIP*, 2015.

- [35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.
- [36] R. K. Srivastava, K. Greff, and J. Schmidhuber. Training very deep networks. In NIPS, 2015.
- [37] M. F. Stollenga, J. Masci, F. Gomez, and J. Schmidhuber. Deep networks with internal selective attention through feedback connections. In NIPS, 2014.
- [38] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv:1602.07261, 2016.
- [39] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In CVPR, 2015.
- [40] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In CVPR, 2016.
- [41] A. Toshev and C. Szegedy. DeepPose: Human pose estimation via deep neural networks. In CVPR, 2014.
- [42] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. In CVPR, 2017.
- [43] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In CVPR, 2017.
- [44] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In ICML, 2015.
- [45] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In CVPR, 2009.
- [46] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In NIPS, 2014.
- [47] X. Zhang, Z. Li, C. C. Loy, and D. Lin. Polynet: A pursuit of structural diversity in very deep networks. In CVPR, 2017.
- [48] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. IEEE TPAMI, 2017.

A. ILSVRC 2017 分类竞赛输入细节

表 3 中的 SENet 是通过将 SE 块集成到 $64 \times 4d64 \times 4d$ 的 ResNeXt-152 的修改版本中构建的, 通过遵循 ResNet-152[9] 的块堆叠来扩展原始 ResNeXt-101[43]。更多设计和训练差异 (除了 SE 块的使用之外) 如下: (a) 对于每个瓶颈构建块, 首先 $1 \times 1 \times 1$ 卷积通道的数量减半, 以性能下降最小的方式降低网络的计算成本。(b) 第一个 $7 \times 7 \times 7$ 卷积层被三个连续的 $3 \times 3 \times 3$ 卷积层所取代。(c) 步长为 2 的 $1 \times 1 \times 1$ 卷积的下采样投影被替换步长为 2 的 $3 \times 3 \times 3$ 卷积以保留信

息。(d) 在分类器层之前插入一个丢弃层 (丢弃比为 0.2) 以防止过拟合。(e) 训练期间使用标签平滑正则化 (如[40]中所介绍的)。(f) 在最后几个训练迭代周期, 所有 BN 层的参数都被冻结, 以确保训练和测试之间的一致性。(g) 使用 8 个服务器 (64 个 GPU) 并行执行培训, 以实现大批量数据大小 (2048), 初始学习率为 1.0。