

SENet

参考文献:

1. Squeeze-and-Excitation Networks: <https://arxiv.org/pdf/1709.01507.pdf>
代码地址: <https://github.com/hujie-frank/SENet>
2. Momenta详解ImageNet 2017夺冠架构SENet: <https://www.cnblogs.com/bonelee/p/9030092.html>
3. [注意力机制] 经典网络模型1——SENet 详解与复现: https://blog.csdn.net/weixin_45084253/article/details/124234120
4. SENet (Squeeze-and-Excitation Networks) 网络详解: <https://blog.csdn.net/Evan123mg/article/details/80058077>
PyTorch代码地址: <https://github.com/miraclewkf/SENet-PyTorch>



Jie Hu¹, Li Shen², Gang Sun¹

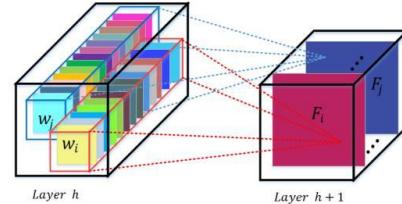
¹ Momenta ² University of Oxford



Convolution

A convolutional filter is expected to be an informative combination

- Fusing **channel-wise** and **spatial** information
- Within local receptive fields

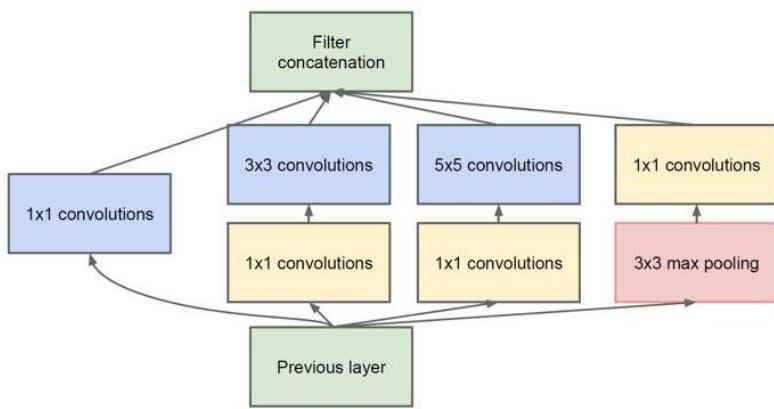


我们从最基本的卷积操作开始说起。

近些年来，卷积神经网络在很多领域上都取得巨大的突破。卷积核作为卷积神经网络的核心，通常被看做是在局部感受野上，将**空间上 (spatial) 的信息**和**特征维度上 (channel-wise) 的信息**进行聚合的信息聚合体。卷积神经网络由一系列卷积层、非线性层和下采样层构成，这样它们能够从全局感受野上去捕获图像的特征来进行图像的描述。

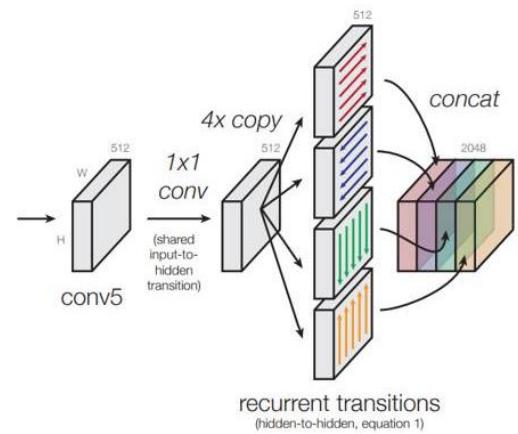
Exploration on Spatial Enhancement

Multi-scale embedding



Inception [9]

Contextual embedding



Inside-outside Network [13]

然而去学到一个性能非常强劲的网络是相当困难的，其难点来自于很多方面。

最近很多工作被提出来从**空间维度层面**来提升网络的性能，

- 如 Inception 结构中嵌入了多尺度信息，聚合多种不同感受野上的特征来获得性能增益；
- 在 Inside-Outside 网络中考虑了**空间中的上下文信息**；
- 还有将**Attention 机制**引入到空间维度上，等等。

这些工作都获得了相当不错的成果。

Squeeze-and-Excitation (SE) Networks

- If a network can be enhanced from the aspect of **channel relationship**?

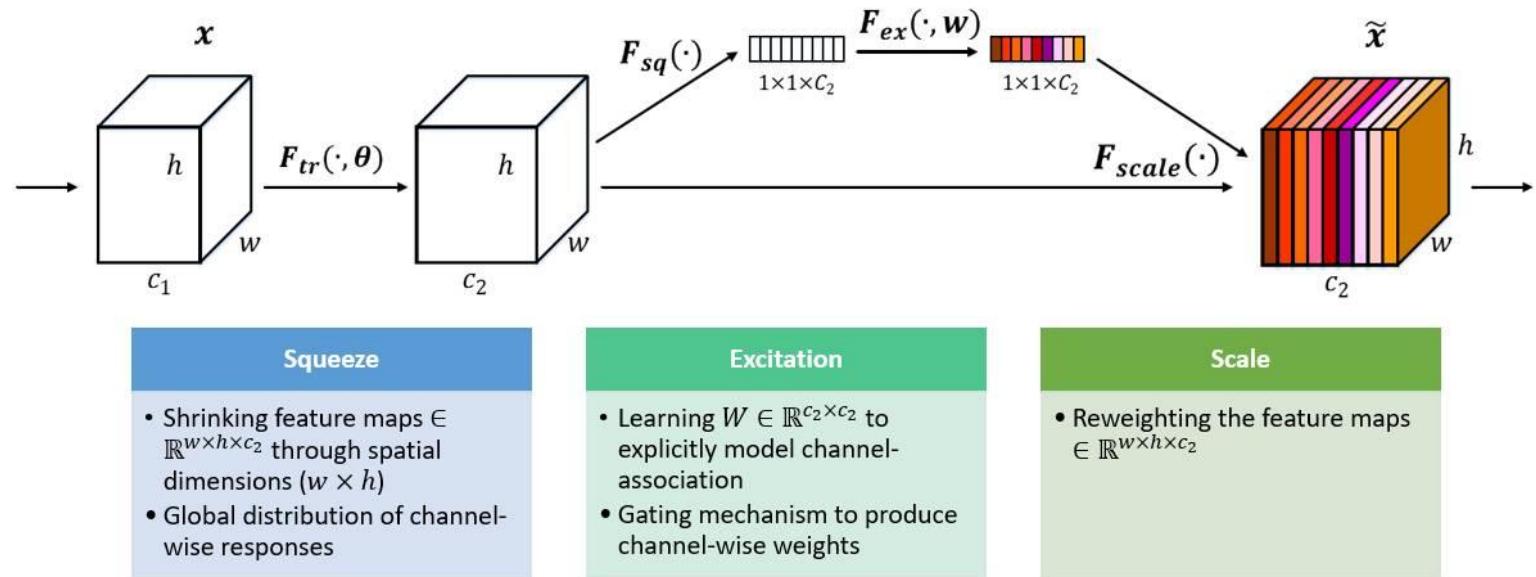
- **Motivation:**

- Explicitly model channel-interdependencies within modules
- Feature recalibration
 - Selectively enhance useful features and suppress less useful ones

我们可以看到，已经有很多工作在空间维度上来提升网络的性能。那么很自然想到，网络是否可以从其他层面来考虑去提升性能，比如考虑**特征通道之间的关系**? 我们的工作就是基于这一点并提出了 **Squeeze-and-Excitation Networks** (简称 **SENet**)。在我们提出的结构中，**Squeeze** 和 **Excitation** 是两个非常关键的操作，所以我们以此来命名。

我们的动机是希望显式地建模特征通道之间的相互依赖关系。另外，我们并不打算引入一个新的空间维度来进行特征通道间的融合，而是采用了一种全新的「特征重标定」策略。具体来说，就是通过学习的方式来自动获取到每个特征通道的重要程度，然后依照这个重要程度去提升有用的特征并抑制对当前任务用处不大的特征。

Squeeze-and-Excitation Module

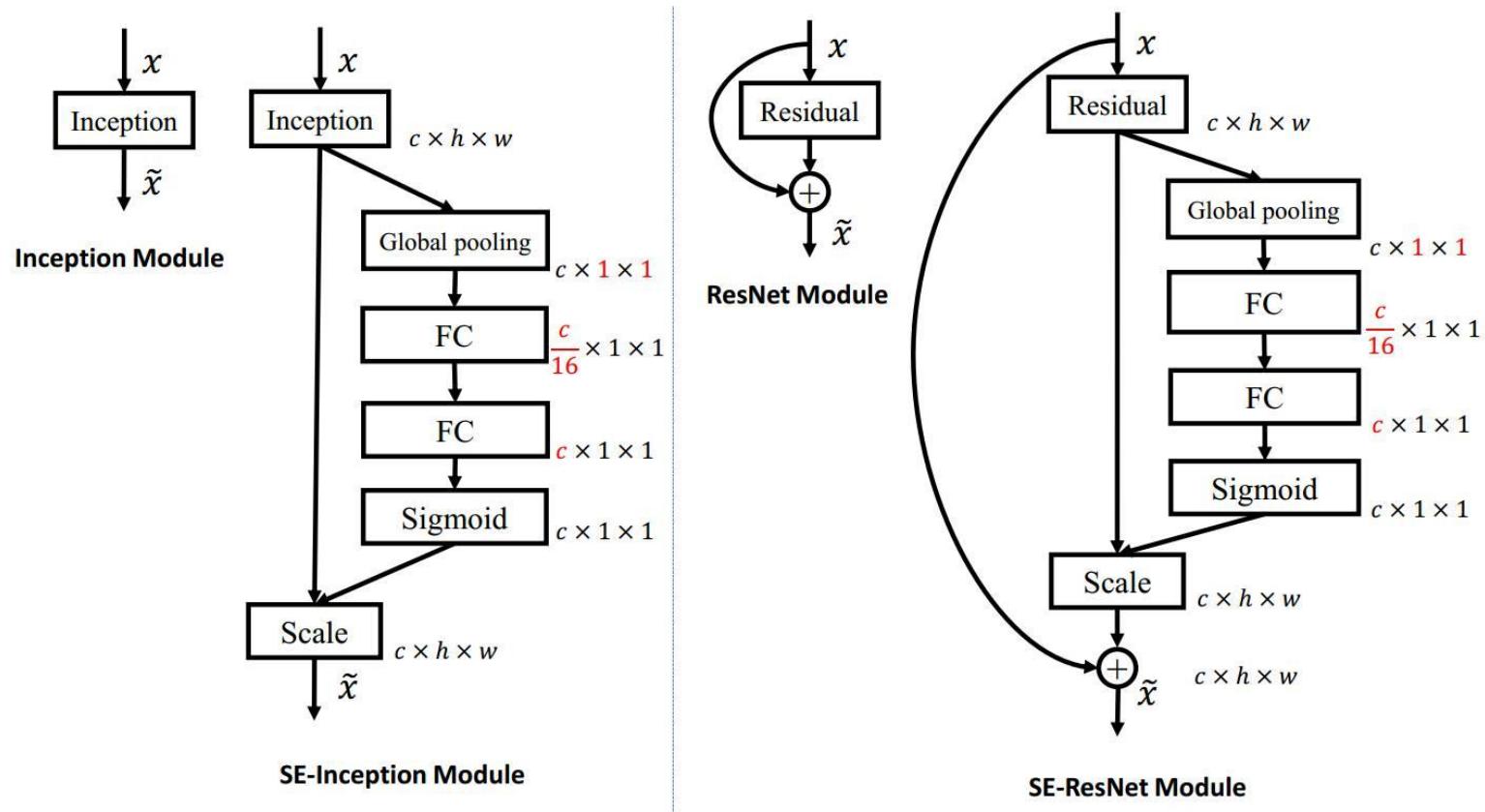


上图是我们提出的 SE 模块的示意图。给定一个输入 x ，其特征通道数为 c_1 ，通过一系列卷积等一般变换后得到一个特征通道数为 c_2 的特征。与传统的 CNN 不一样的是，接下来我们通过三个操作来重标定前面得到的特征。

首先是 **Squeeze 操作**，我们顺着空间维度来进行特征压缩，将每个二维的特征通道变成一个实数，这个实数某种程度上具有全局的感受野，并且输出的维度和输入的特征通道数相匹配。它表征着在特征通道上响应的全局分布，而且使得靠近输入的层也可以获得全局的感受野，这一点在很多任务中都是非常有用的。

其次是 **Excitation 操作**，它是一个类似于循环神经网络中门的机制。通过参数 w 来为每个特征通道生成权重，其中参数 w 被学习用来显式地建模特征通道间的相关性。

最后是一个 **Reweight 的操作**，我们将 Excitation 的输出的权重看做是进过特征选择后的每个特征通道的重要性，然后通过乘法逐通道加权到先前的特征上，完成在通道维度上的对原始特征的重标定。



上左图是将 SE 模块嵌入到 Inception 结构的一个示例。方框旁边的维度信息代表该层的输出。

这里我们使用 global average pooling 作为 Squeeze 操作。紧接着两个 Fully Connected 层组成一个 Bottleneck 结构去建模通道间的相关性，并输出和输入特征同样数目的权重。我们首先将特征维度降低到输入的 $1/16$ ，然后经过 ReLU 激活后再通过一个 Fully Connected 层升回到原来的维度。这样做比直接用一个 Fully Connected 层的好处在于：

1. 具有更多的非线性，可以更好地拟合通道间复杂的相关性；
2. 极大地减少了参数量和计算量。然后通过一个 Sigmoid 的门获得 $0\sim1$ 之间归一化的权重，最后通过一个 Scale 的操作来将归一化后的权重加权到每个通道的特征上。

除此之外，SE 模块还可以嵌入到含有 skip-connections 的模块中。上右图是将 SE 嵌入到 ResNet 模块中的一个例子，操作过程基本和 SE-Inception 一样，只不过是在 Addition 前对分支上 Residual 的特征进行了特征重标定。如果对 Addition 后主支上的特征进行重标定，由于在主干上存在 $0\sim1$ 的 scale 操作，在网络较深 BP 优化时就会在靠近输入层容易出现梯度消散的情况，导致模型难以优化。

目前大多数的主流网络都是基于这两种类似的单元通过 repeat 方式叠加来构造的。由此可见，SE 模块可以嵌入到现在几乎所有的网络结构中。通过在原始网络结构的 building block 单元中嵌入 SE 模块，我们可以获得不同种类的 SENet，如 SE-BN-Inception、SE-ResNet、SE-ReNeXt、SE-Inception-ResNet-v2 等等。

Model and Computational Complexity

SE-ResNet-50 vs. ResNet-50

- Parameters: 2%~10% additional parameters
- Computation cost: <1% additional computation (theoretical)
- GPU inference time: 10% additional time
- CPU inference time: <2% additional time

从上面的介绍中可以发现，**SENet 构造非常简单，而且很容易被部署，不需要引入新的函数或者层。**除此之外，它还在模型和计算复杂度上具有良好的特性。拿 ResNet-50 和 SE-ResNet-50 对比举例来说，SE-ResNet-50 相对于 ResNet-50 有着 10% 模型参数的增长。额外的模型参数都存在于 Bottleneck 设计的两个 Fully Connected 中，由于 ResNet 结构中最后一个 stage 的特征通道数目为 2048，导致模型参数有着较大的增长，实验发现移除掉最后一个 stage 中 3 个 build block 上的 SE 设定，可以将 10% 参数量的增长减少到 2%。此时模型的精度几乎无损失。

另外，由于在现有的 GPU 实现中，都没有对 global pooling 和较小计算量的 Fully Connected 进行优化，这导致了在 GPU 上的运行时间 SE-ResNet-50 相对于 ResNet-50 有着约 10% 的增长。尽管如此，其理论增长的额外计算量仅仅不到 1%，这与其在 CPU 运行时间上的增长相匹配 (~2%)。可以看出，在现有网络架构中嵌入 SE 模块而导致额外的参数和计算量的增长微乎其微。

Training – Momenta ROCS

- Data augmentation
 - ✓ Mirror flip, Random size crop [9], Rotation, Color Jitter
- Mini-batch data sampling
 - ✓ Balance-data strategy [7]
- Training hyper-parameters
 - ✓ 4 or 8 GPU servers (8 NVIDIA Titan X per server)
 - ✓ Batch-size: 1024 / 2048 (32 per GPU)
 - ✓ Initial learning rate : 0.6 (decrease each 30 epochs)
 - ✓ Synchronous SGD with momentum 0.9

在训练中，我们使用了一些常见的数据增强方法和 Li Shen 提出的均衡数据策略。为了提高训练效率，我们使用了我们自己优化的分布式训练系统 ROCS，并采用了更大的 batch-size 和初始学习率。所有的模型都是从头开始训练的。

Benefits against Network Depth

	Original		Our re-implementation		SE-module	
	top-1 err.	top-5 err.	top-1 err.	top-5 err.	top-1 err.	top-5 err.
ResNet-50 [1]	24.7	7.8	24.80	7.48	23.29 (1.51)	6.62 (0.86)
ResNet-101 [1]	23.6	7.1	23.17	6.52	22.38 (0.79)	6.07 (0.45)
ResNet-152 [1]	23.0	6.7	22.42	6.34	21.57 (0.85)	5.73 (0.61)

Table 1. Error rates (%) of single-crop results on the ImageNet-1k validation set.

首先我们来看一下网络的深度对 SE 的影响。上表分别展示了 ResNet-50、ResNet-101、ResNet-152 和嵌入 SE 模型的结果。

- 第一栏 Original 是原作者实现的结果，
- 为了进行公平的比较，我们在 ROCS 上重新进行了实验得到 Our re-implementation 的结果 (ps. 我们重实现的精度往往比原 paper 中要高一些)。
- 最后一栏 SE-module 是指嵌入了 SE 模块的结果，它的训练参数和第二栏 Our re-implementation 一致。括号中的红色数值是指相对于 Our re-implementation 的精度提升的幅值。

从上表可以看出，SE-ResNets 在各种深度上都远远超过了其对应的没有 SE 的结构版本的精度，这说明无论网络的深度如何，SE 模块都能够给网络带来性能上的增益。值得一提的是，SE-ResNet-50 可以达到和 ResNet-101 一样的精度；更甚，SE-ResNet-101 远远地超过了更深的 ResNet-152。

Incorporation with Modern Architectures

	Original		Our re-implementation		SE-module	
	top-1 err.	top-5 err.	top-1 err.	top-5 err.	top-1 err.	top-5 err.
ResNeXt-50 [7]	22.2	-	22.11	5.90	21.10 (1.01)	5.49 (0.41)
ResNeXt-101 [7]	21.2	5.6	21.18	5.57	20.70 (0.48)	5.01 (0.56)
BN-Inception [4]	25.2	7.82	25.38	7.89	24.23 (1.15)	7.14 (0.75)
Inception-ResNet-v2 [5]	19.9 [†]	4.9 [†]	20.37	5.21	19.80 (0.57)	4.79 (0.42)

Table 2. Error rates (%) of single-crop results on the ImageNet-1k validation set. Error rate followed by † means that the image size for center crop is not clear and it evaluates on the non-blacklisted subset of validation set [5], which may lead to slightly better results.

另外，为了验证 SE 模块的泛化能力，我们也在除 ResNet 以外的结构上进行了实验。从上表可以看出，将 SE 模块嵌入到 ResNeXt、BN-Inception、Inception-ResNet-v2 上均获得了不菲的增益效果。由此看出，SE 的增益效果不仅仅局限于某些特殊的网络结构，它具有很强的泛化性。

Comparison with State-of-the-art

	224 × 224		320 × 320 / 299 × 299	
	top-1 err.	top-5 err.	top-1 err.	top-5 err.
ResNet-152 [1]	23.0	6.7	21.3	5.5
ResNet-200 [3]	21.7	5.8	20.1	4.8
Inception-v3 [10]	-	-	21.2	5.6
Inception-v4 [8]	-	-	20.0	5.0
Inception-ResNet-v2 [8]	-	-	19.9	4.9
ResNeXt-101 [11] (64 × 4d)	20.4	5.3	19.1	4.4
DenseNet-161 [4] (k = 48)	22.2	-	-	-
Very Deep PolyNet [12]	-	-	18.71	4.25
SENet	18.68	4.47	17.28	3.79

Table 4. Single-crop error of state-of-the-art CNNs on ImageNet-1k validation set. The shorter edge is resized to 256 and 320 (299 for Inception models) respectively and then conduct center-crop testing as in [11]. The **SENet** is our well-structured model whose error rates are remarkably lower than previous models.

SENet is a SE-ResNeXt-152 (64 × 4d)

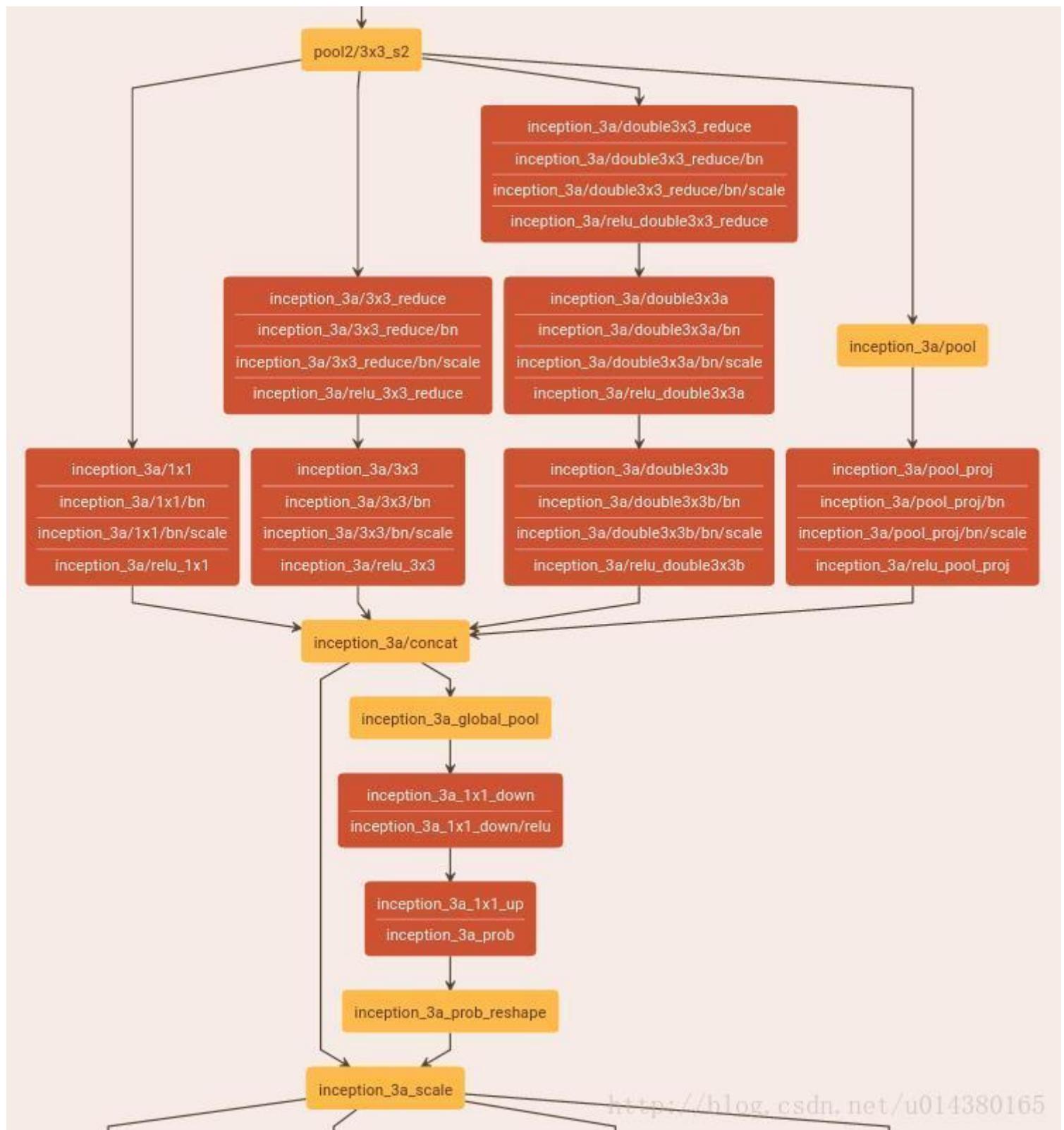
在上表中我们列出了一些最新的在 ImageNet 分类上的网络的结果。其中我们的 SENet 实质上是一个 SE-ResNeXt-152 (64x4d)，在 ResNeXt-152 上嵌入 SE 模块，并做了一些其他修改和训练优化上的小技巧，这些我们会在后续公开的论文中进行详细介绍。可以看出 SENet 获得了迄今为止在 single-crop 上最好的性能。

References

- [1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2015.
- [2] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.
- [3] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV*, 2016.
- [4] G. Huang, Z. Liu, K. Weinberge, and L. Maaten. Densely connected convolutional networks. In *CVPR*, 2017.
- [5] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.
- [6] 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. In *IJCV*, 2015.
- [7] L. Shen, Z. Lin, and Q. Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *ECCV*, 2016.
- [8] C. Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. In *arXiv preprint arXiv:1602.07261*, 2016.
- [9] 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.
- [10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.
- [11] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2016.
- [12] X. Zhang, Z. Li, C. Chen, and D. Lin. Polynet: A pursuit of structural diversity in very deep networks. In *CVPR*, 2017.
- [13] 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.

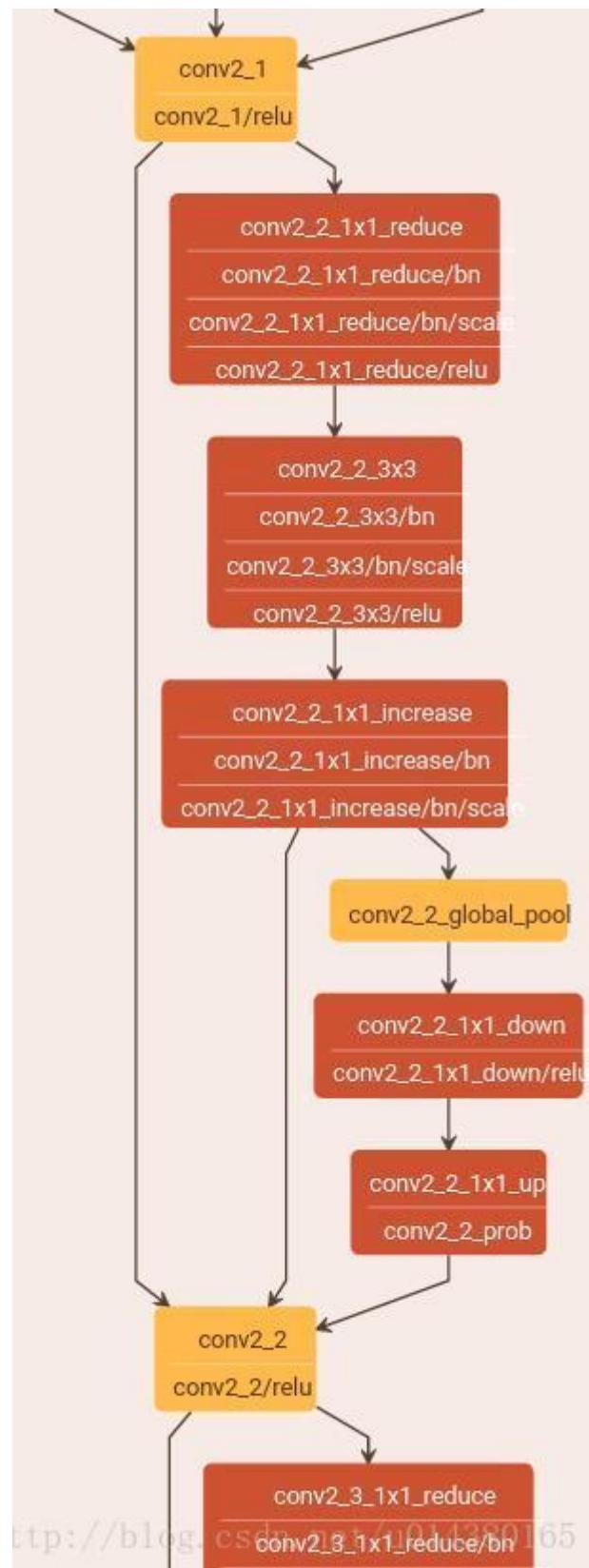
附：看了下 caffe 代码 (.prototxt文件)

和文章的实现还有些不一样。下图是在 Inception 中添加 SENet 的可视化结果：SE-BN-Inception，在 Inception 中是在每个 Inception 的后面连上一个 SENet，下图的上面一半就是一个 Inception，下面一半就是一个 SENet，然后这个 SENet 下面又连着一个新的 Inception。



注意看这个 SENet 的红色部分都是用卷积操作代替文中的全连接层操作实现的，个人理解是为了减少参数（原来一个全连接层是 $C \times C / r$ 个参数，现在变成了 C / r 个参数了），计算量应该是不影响的，都是 $C \times C / r$ 。具体来说，inception_3a_1*1_down 是输出 channel 为 16 的 1×1 卷积，其输入 channel 是 256，这也符合文中说的缩减因子为 16 ($256 / 16 = 16$)；而 inception_3a_1*1_up 是输出 channel 为 256 的 1×1 卷积。其它层都和文中描述一致，比如 inception_3a_global_pool 是 average pooling，inception_3a_prob 是 sigmoid 函数。

SE-ResNet-50 的情况也类似，如下图。在 ResNet 中都是在 Residual block 中嵌入 SENet。下图最左边的长条连线是原来 Residual block 的 skip connection，右下角的 conv2_2_global_pool 到 conv2_2_prob 以及左边那条连线都是 SENet。不过也是用两个 1*1 卷积代替文中的两个全连接层。



参考文献:

- [注意力机制] 经典网络模型1——SENet 详解与复现: https://blog.csdn.net/weixin_45084253/article/details/124234120
- SENet介绍和在推荐中的实践: <https://zhuanlan.zhihu.com/p/483761282>

🚀 Squeeze-and-Excitation Networks

Squeeze : 挤压 Excitation : 激励 ;

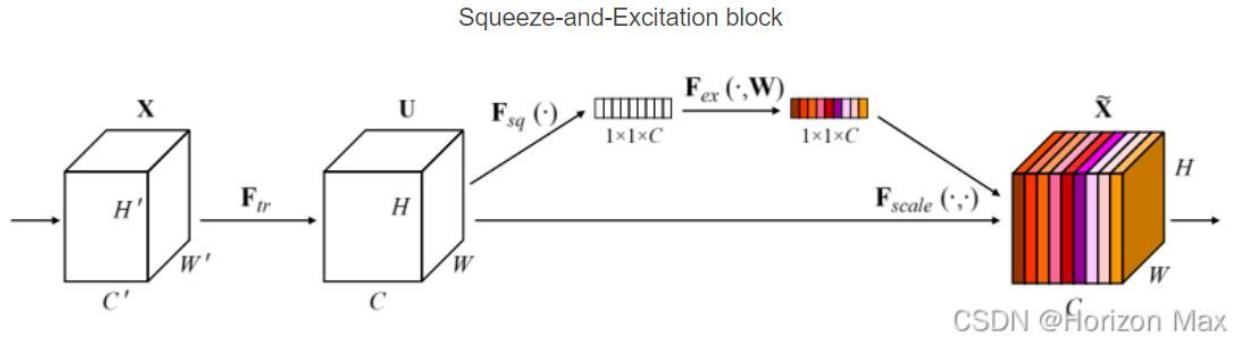
Squeeze-and-Excitation Networks 简称 SENet , 由 Momenta 和牛津大学的Jie Hu等人提出的一种新的网络结构;

目标是通过建模 卷积特征通道之间的相互依赖关系 来提高网络的表示能力;

在2017年最后一届 ImageNet 挑战赛(ILSVRC) classification 任务中获得 冠军, 将错误率降低到 2.251% ;

🚀 SENet 详解

🎨 Squeeze-and-Excitation block



对于任意给定的变换: $F_{tr} : X \rightarrow U$, 其中 $X \in \mathbb{R}^{H' \times W' \times C'}$, $U \in \mathbb{R}^{H \times W \times C}$, F_{tr} 用作一个卷积算子;

▶ Squeeze: Global Information Embedding

挤压: 全局信息嵌入

(1) Squeeze : 特征U通过 squeeze 压缩操作, 将跨空间维度 $H \times W$ 的特征映射进行聚合, 生成一个通道描述符, $H \times W \times C \rightarrow 1 \times 1 \times C$; 将全局空间信息 压缩到上述 通道描述符 中, 使来这些 通道描述符 可以被 其输入的层 利用, 这里采用的是 global average pooling ;

$$z_c = \mathbf{F}_{sq}(\mathbf{u}_c) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W u_c(i, j)$$

▶ Excitation: Adaptive Recalibration

激励: 自适应调整

(2) Excitation : 每个通道通过一个 基于通道依赖 的自选门机制 来学习特定样本的激活, 使其学会使用全局信息, 有选择地强调信息特征, 并抑制不太有用的特征, 这里采用的是 sigmoid , 并在中间嵌入了 ReLU 函数用于限制模型的复杂性和帮助训练;

通过 两个全连接层(FC) 构成的瓶颈来参数化门控机制, 即 W_1 用于降低维度, W_2 用于维度递增;

$$\mathbf{s} = \mathbf{F}_{ex}(\mathbf{z}, \mathbf{W}) = \sigma(g(\mathbf{z}, \mathbf{W})) = \sigma(\mathbf{W}_2 \delta(\mathbf{W}_1 \mathbf{z}))$$

(3) Reweighting : 将 Excitation 输出的权重通过乘法逐通道加权到输入特征上;

$$\tilde{\mathbf{x}}_c = \mathbf{F}_{scale}(\mathbf{u}_c, s_c) = s_c \mathbf{u}_c \quad \text{where } \tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_C]$$

总的来说 SE Block 就是在 Layer 的输入和输出之间添加结构: global average pooling - FC - ReLU - FC - sigmoid ;

SE block 的灵活性意味着它可以直接应用于标准卷积以外的转换, 通过将 SE block 集成到任何复杂模型当中来开发SENet;



方知存心 2022.06.15



有个问题 既然是global average pooling - FC - ReLU - FC - sigmoid 那在实现的时候为什么用二维卷积代替FC呢



方知存心 回复 Horizon Max 2022.06.15



明白了😊



Horizon Max 作者 回复 方知存心 2022.06.15



在前面已经通过 `nn.AdaptiveAvgPool2d((1, 1))` 实现降维，同样的一个网络结构可以通过不同的方式去实现，VGG等模型也是通过卷积操作代替全连接，这样既可以减小参数量又能更有效的捕获特征图中的有效信息

下图引用自：SENet介绍和在推荐中的实践 <https://zhuanlan.zhihu.com/p/483761282>



张hyan

请问，如果针对一维数据，要如何进行se？比如输入为[batch, 1, 500]这种，输入通道为1的话，在进行se前是否需要先升维？变为[batch, n, 500]，输入通道为n这种？但是本身模型原始输入就仅是一维数据，在模型中进行升维有意义么？🤔 (我用的pytorch)

01-09

● 回复



赞



Biububiubiu ▶ 热心市民

升维可以理解为把输入映射到高维空间，原本在一维无法得到的特征，在高维空间可能有好的表现

06-16

● 回复



1



热心市民 ▶ 张hyan

你好，请问提升通道数的意义是什么，仅仅是升维再压缩得到特征权重吗？

03-13

● 回复



赞



Lemon tree

你好，这个问题你有想明白吗？我最近和你遇到了一样的问题 我的输入和你是一模一样的，目前也在想如何针对一维数据利用这个网络🤔

01-18

● 回复



赞



张hyan ▶ Lemon tree

升维，提升它的通道数，然后再投入到SE中，我目前是这么做的。

01-18

● 回复



赞

下图引用自：【论文解读】SENet网络 <https://zhuanlan.zhihu.com/p/80123284>



王火箭

我觉得很多东西的作用都是乱说的，说的高端，但是没有真凭实据。扪心自问一下，可能自己都不知道在说什么。

2021-08-21

● 回复



3



Onedroid

故事只能这么说，说白了这东西就是用很小的计算代价，又保证足够的额外参数和网络深度，这肯定可以提升模型。

2022-02-21

● 回复



2



李慕清

这和通道注意力机制是一个东西吗

2019-11-26

● 回复



2



林小轩

单独看Excitation就是通道注意力机制。但大的收益是看上去不起眼的Squeeze模块带来的，论文里有实验。简单等价通道注意力机制不是很准确。

2020-09-28

● 回复



1