Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning

Christian Szegedy
Google Inc.
1600 Amphitheatre Pkwy, Mountain View, CA
szegedy@google.com

Sergey Ioffe
sioffe@google.com

Vincent Vanhoucke
vanhoucke@google.com

Alex Alemi
alemi@google.com

Abstract

提示：该*PDF*由*SpeedPaper*生成，版权归原文作者所有。翻译内容仅供参考，请仔细鉴别并以原文为准。

查看更多论文翻译与复现代码：*https://github.com/hanknewbird/SpeedPaper*

深度卷积网络近年来已成为图像识别性能取得最重大进展的核心。一个例子是*Inception*架构，已经表现出在相对较低的计算成本下取得非常好的性能。最近，在传统架构加入残差连接的情况下，在*2015*年*ILSVRC*挑战中实现了最先进的性能；其性能类似于最新一代的*Inception-v3*网络。这引发了一个问题，即将*Inception*架构与残差连接相结合是否有任何好处。在这里，我们提供明确的实证证据表明，使用残差连接加速了*Inception*网络的训练。还有一些证据表明，带有残差连接的*Inception*网络在轻微边缘上胜过了同样昂贵的没有残差连接的*Inception*网络。我们还提出了几种新的简化架构，适用于带有残差和不带残差的*Inception*网络。这些变体显着提高了*ILSVRC 2012*分类任务中的单帧识别性能。我们进一步展示了如何通过正确的激活缩放稳定训练非常宽的残差*Inception*网络。通过三个残差和一个*Inception-v4*的集合，我们在*ImageNet*分类挑战的测试集上实现了*3.08%*的前*5*错误率。

1. Introduction

自2012年ImageNet竞赛[11]由Krizhevsky等人[8]获胜以来，他们的网络“AlexNet”已成功应用于更多计算机视觉任务，例如目标检测[4]、分割[10]、人体姿势估计[17]、视频分类[7]、物体跟踪[18]和超分辨率[3]等。这些例子只是深度卷积网络至今应用非常成功的众多应用中的一部分。

在本研究中，我们研究了两个最近的思想的结合：He等人在[5]中引入的残差连接和最新修订的Inception架构版本[15]。在[5]中，提出残差连接对训练非常深层次架构至关重要。由于Inception网络往往非常深，将Inception架构的滤波器连接阶段替换为残差连接是很自然的。这样可以使Inception获得残差方法的所有好处，同时保留其计算效率。

除了简单整合外，我们还研究了通过使其更深更广使Inception本身更加高效的可能性。为此，我们设计了一个名为Inception-v4的新版本，它具有更统一简化的架构和更多的Inception模块比Inception-v3。历史上，Inception-v3继承了很多早期版本的包袱。技术约束主要来自于需要通过DistBelief[2]对模型进行分区以进行分布式训练。现在，在将我们的训练设置迁移到TensorFlow[1]后，这些约束已被解除，从而使我们能够显著简化架构。这个简化架构的细节在第3节中描述。

在本报告中，我们将比较两种单纯的Inception变

体Inception-v3和v4，以及类似昂贵的混合Inception-ResNet版本。诚然，这些模型是以一种准备而选取的方式选择的，主要约束是模型的参数和计算复杂性应该与非残差模型的成本相似。事实上，我们测试了更大更宽的Inception-ResNet变体，它们在ImageNet分类挑战[11]数据集上表现非常相似。

这里报告的最后一个实验是对所有在这里呈现的最佳性能模型的集成评估。由于明显地Inception-v4和Inception-ResNet-v2表现得同样出色，超越了ImageNet验证数据集上单帧性能的最新水平，我们想看看这两者的组合如何推动该数据集上的最新技术水平。令人惊讶的是，我们发现在单帧性能上的提高并没有转化为集成性能上同样大的提升。尽管如此，它仍然使我们可以报告在验证集上使用四个模型集成的 top-5 错误率为3.1%，创造了一个新的技术水平，据我们所知。

在最后一节中，我们研究了一些分类失败，并得出结论，集成模型仍未达到该数据集标注的标签噪声水平，对预测仍有改善空间。

2. Related Work

卷积网络在Krizhevsky等人的工作 [8]之后已经在大规模图像识别任务中变得流行起来。接下来一些重要的里程碑是由Lin等人提出的Network-in-network [9]，Simonyan等人的VGGNet [12]，以及Szegedy等人的GoogLeNet（Inception-v1） [14]。

He等人在 [5]中引入了残差连接，在其中他们为使用信号的加性合并在图像识别和特别是物体检测中的优势提供了令人信服的理论和实证证据。作者认为残差连接在训练非常深的卷积模型时是必需的。我们的研究结果似乎不支持这种观点，至少在图像识别中如此。然而，了解残差连接提供的有益方面的真正规模可能需要更深层次的架构和更多的测量点。

在实验部分中，我们展示了在不使用残差连接的情况下训练竞争性的非常深层网络并不是很困难。然而，残差连接的使用似乎极大地提高了训练速度，这本身就是对它们使用的一个很好的论点。 Inception深度卷积架构首次出现在 [14] 中，并在本文中被称为GoogLeNet或Inception-v1。随后，该Inception架构通过多种方式进行了改进，首先是通过引入批量归一化 [6]（Inception-v2）由Ioffe等人完成。随后，通过在

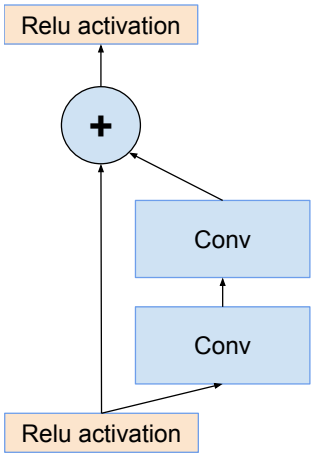

图 1. 在He等人的工作中引入的残差连接[5]。

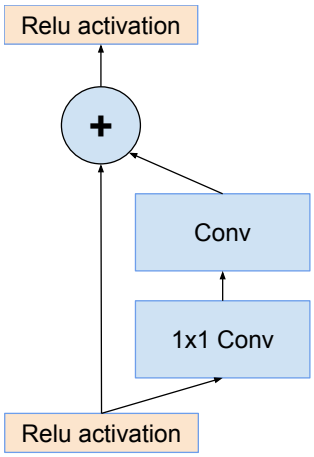

图 2. 通过[5]提出的ResNet连接的优化版本，用于屏蔽计算。

第三次迭代中加入额外的分解思想对架构进行了改进[15]，本报告中将称其为Inception-v3。

3. Architectural Choices

3.1. Pure Inception blocks

我们过去的Inception模型通常是以分区的方式训练的，其中每个副本被分成多个子网络，以便能够将整个模型放入内存中。然而，Inception架构是高度可调的，意味着在各层的滤波器数量方面有许多可能的变化，这不会影响完全训练的网络质量。为了优化训练速度，我们过去会仔细调整层大小，以在各个模型子网络之间平衡计算。相比之下，随着TensorFlow的引入，我们最新的模型可以在不分区割复制品的情况下训练。部分原因在于最近通过仔细考虑哪些张量需要

用于梯度计算以及构造计算来减少这些张量数量的反向传播内存优化。在历史上，我们在更改架构选择方面相对保守，并将我们的实验限制在改变孤立网络组件的同时保持网络的其余部分稳定。不简化先前的选择导致网络看起来比需要的更复杂。在我们的新实验中，针对Inception-v4，我们决定摆脱这些不必要的负担，并为每个网格大小的Inception块做出统一选择。请参考图 9查看Inception-v4网络的大规模结构，以及图3，4，5，6，7和8查看其组件的详细结构。图中未标有“V”的所有卷积都是相同填充的，意味着它们的输出网格与其输入的大小相匹配。标有“V”的卷积是有效填充的，意味着每个单元的输入块完全包含在上一层中，激活图的输出网格大小相应减小。

3.2. Residual Inception Blocks

对于Inception网络的剩余版本，我们使用比原始Inception更便宜的Inception块。每个Inception块后面跟着一个过滤器扩展层（1×1卷积没有激活函数），用于增加滤波器组的维度，以便在加法之前匹配输入的深度。这是为了补偿Inception块导致的降维效应。

我们尝试了几个剩余版本的Inception。其中只有两个在这里详细介绍。第一个是”Inception-ResNet-v1”，大致上计算成本与Inception-v3相当，而”Inception-ResNet-v2”匹配了新引入的Inception-v4网络的原始成本。请参见图 15，了解这两个变体的大规模结构。（然而，Inception-v4的步骤时间在实践中明显较慢，可能是由于更多层的存在。）

我们的剩余和非剩余Inception变体之间的另一个小技术差异是，在Inception-ResNet的情况下，我们仅在传统层的顶部使用批量归一化，而不是在求和上面使用。可以合理预期，充分利用批量归一化应该是有益的，但我们希望每个模型副本都可以在单个GPU上进行训练。事实证明，具有大激活尺寸的层的内存占用量占据了不成比例的GPU内存。通过省略那些层顶部的批量归一化，我们能够显著增加Inception块的整体数量。我们希望通过更好地利用计算资源，做出这种权衡将变得不再必要。

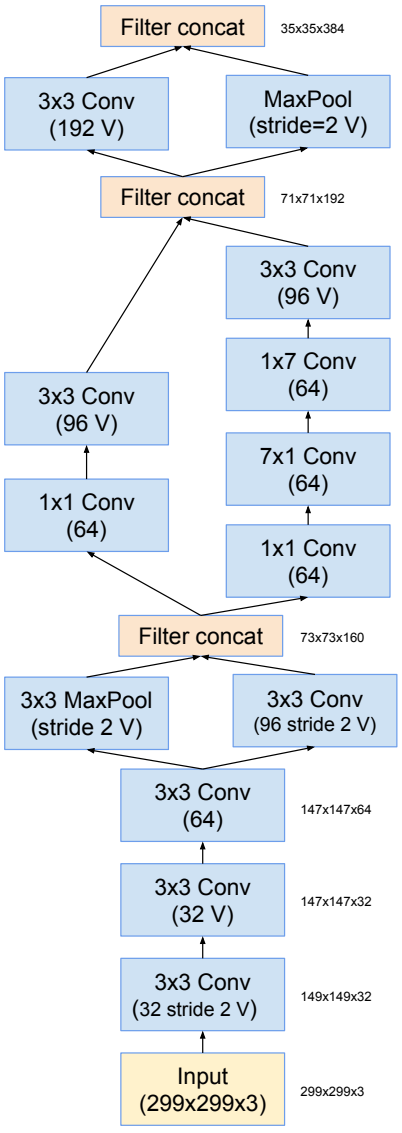

图 3. 纯Inception-v4和Inception-ResNet-v2网络的茎部模式。这是这些网络的输入部分。参见图 9和 15。

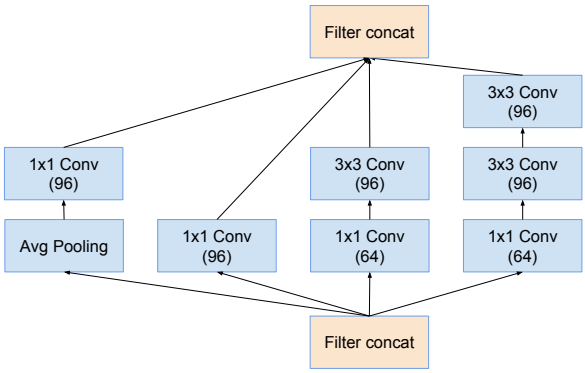

图 4. 这是纯Inception-v4网络35 × 35格模块的架构。这是图9中的Inception-A块。

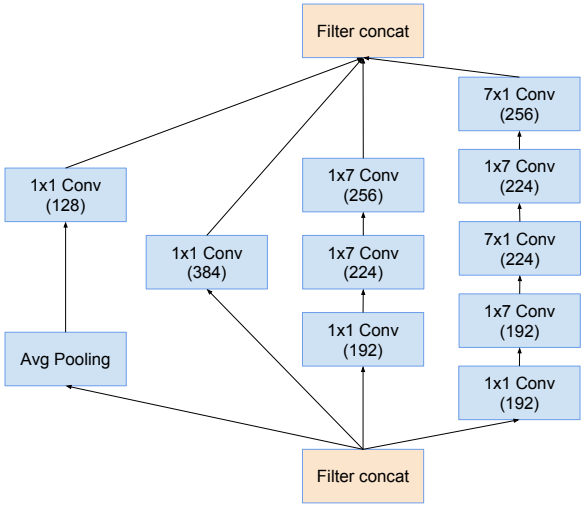

图 5. 纯Inception-v4网络的17 × 17网格模块结构。这是图9中Inception-B块的结构。

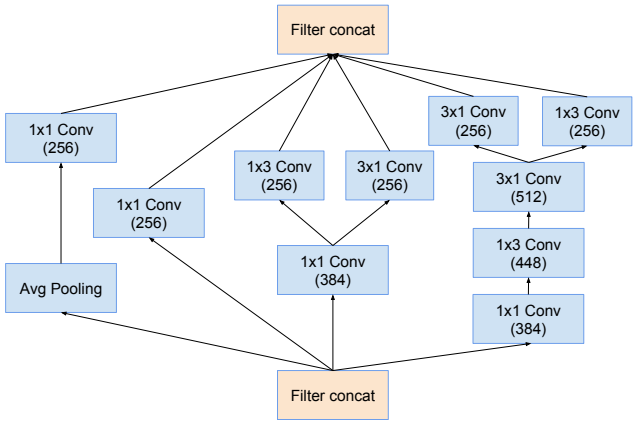

图 6. 纯Inception-v4网络的8 × 8网格模块的架构。这是图 9中的Inception-C块。

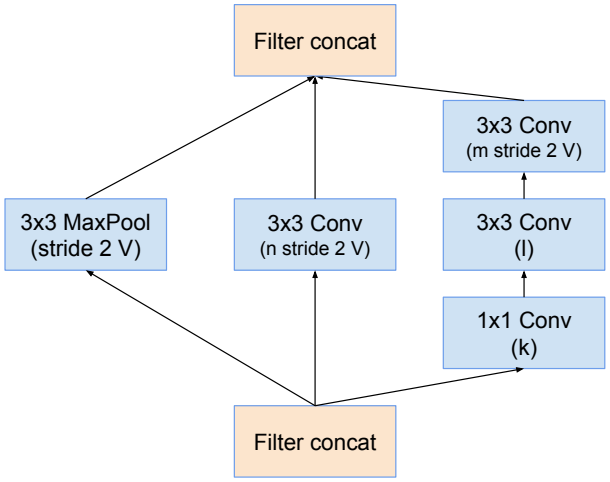

图 7. 35 × 35 到 17 × 17 缩减模块的结构。这些块的不同变体（具有不同数量的滤波器）在本文中呈现的每个新的 Inception（-v4，-ResNet-v1，-ResNet-v2）变体的图中使用，如图 9 和 15。k、l、m、n 数量代表滤波器组的大小，可以在表 1 中查找。

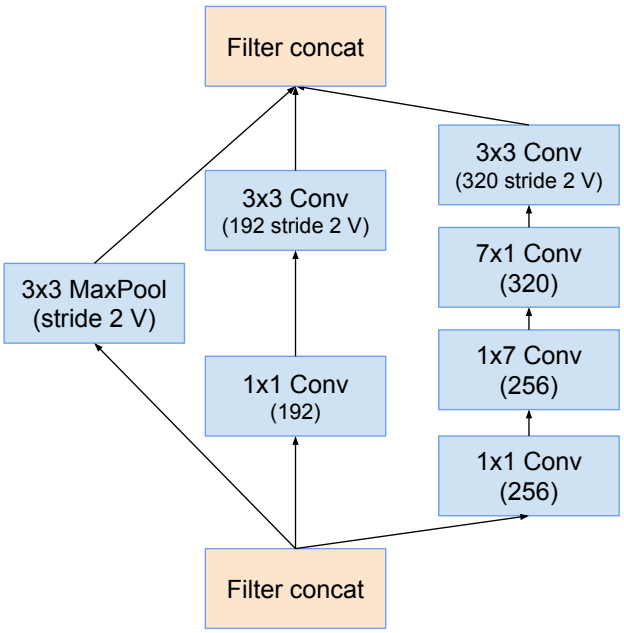

图 8. 用于 17 × 17 到 8 × 8 网格缩减模块的模式。这是纯Inception-v4网络中使用的缩减模块，如图 9所示。

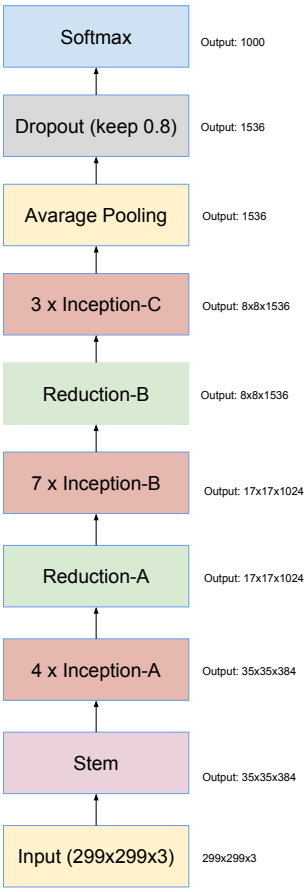

图 9. 整个Inception-v4网络的结构框图。有关详细模块，请参考图3，4，5，6，7和8，了解各个组件的详细结构。

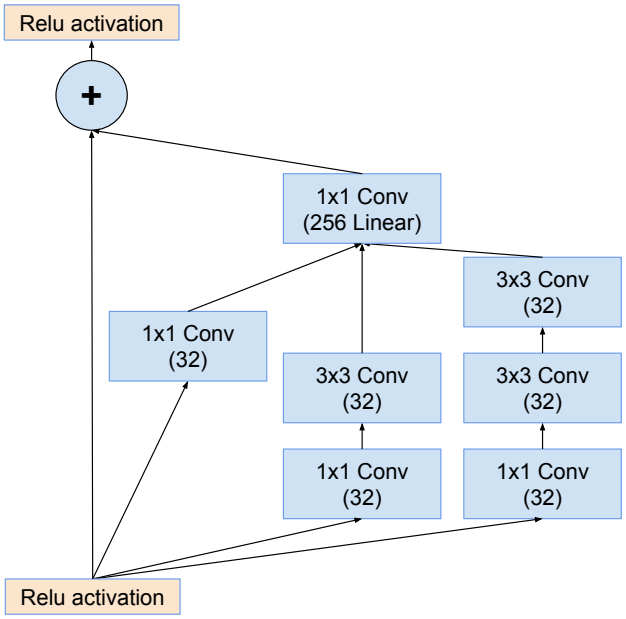

图 10. Inception-ResNet-v1网络的35 × 35网格（Inception-ResNet-A）模块的模式。

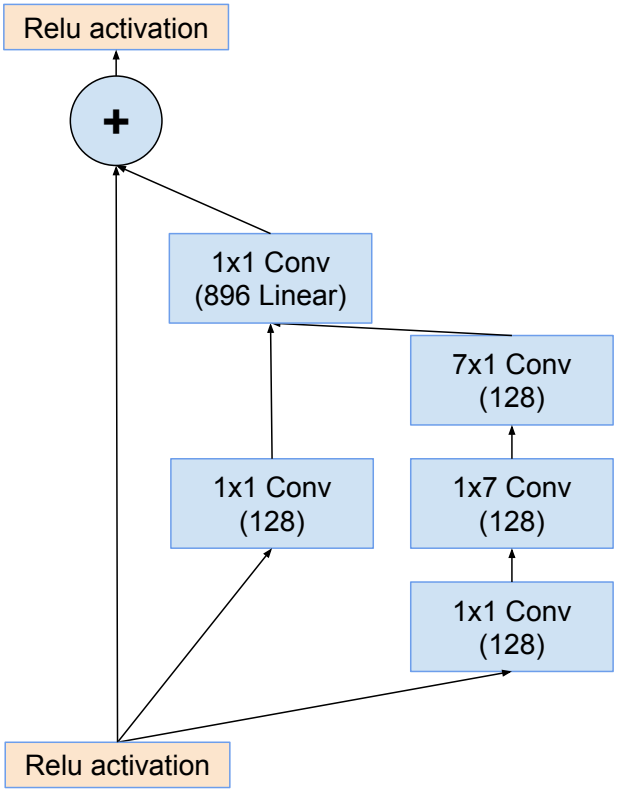

图 11. Inception-ResNet-v1网络中的17 × 17网格（Inception-ResNet-B）模块的架构。

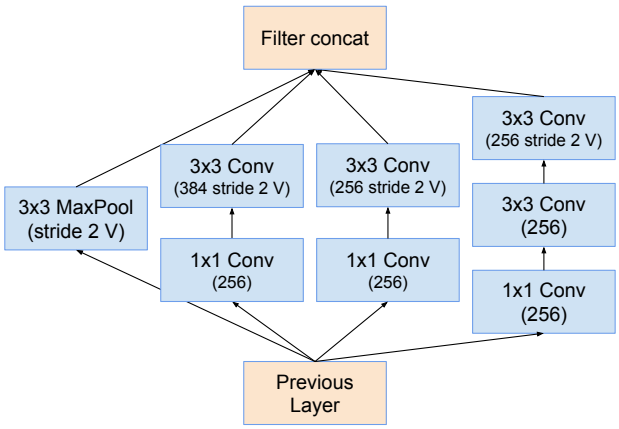

图 12. “Reduction-B” 17 × 17 到 8 × 8 的网格缩减模块。这个模块被用于更小的Inception-ResNet-v1网络，如图 15所示。

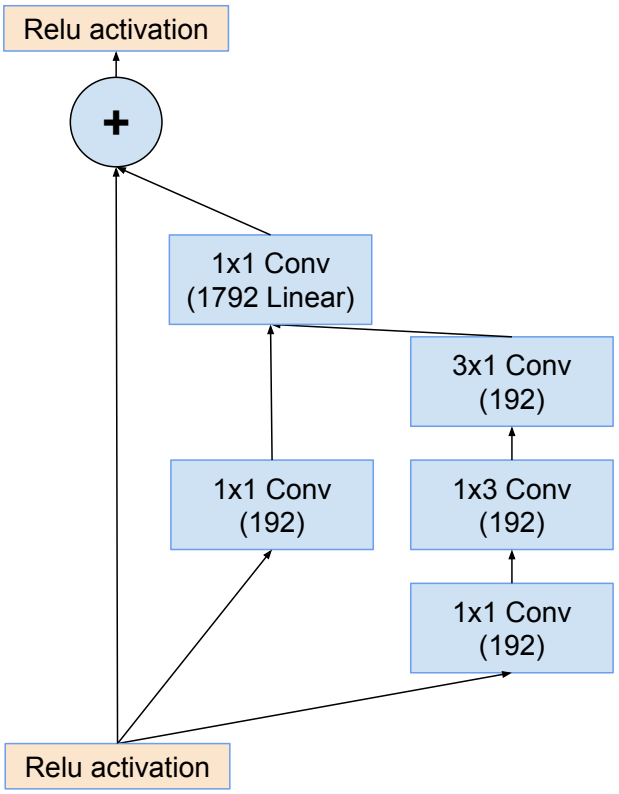

图 13. Inception-ResNet-v1网络中8×8网格（Inception-ResNet-C）模块的架构。

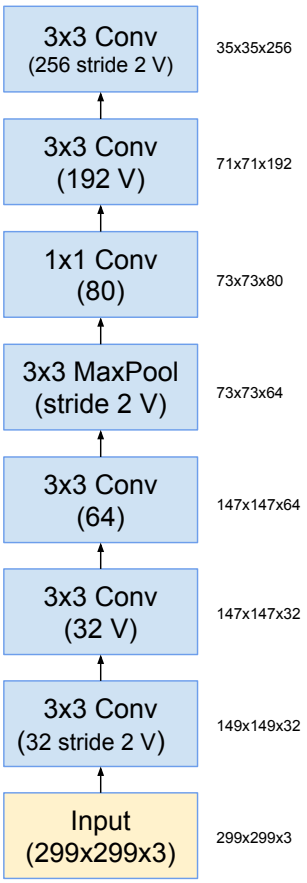

图 14. Inception-ResNet-v1网络的主干。

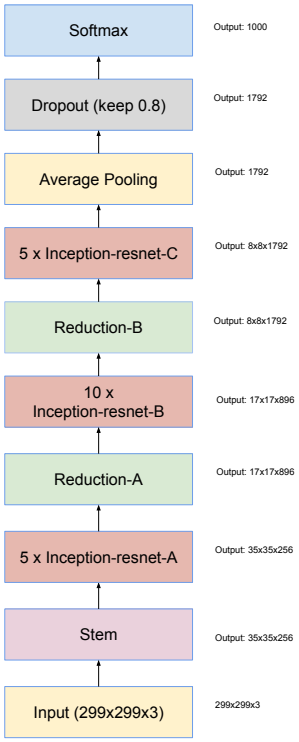

图 15. Inception-ResNet-v1和Inception-ResNet-v2网络的结构图。这个构架适用于两个网络，但基础组件不同。 Inception-ResNet-v1使用如图14，10，7，11，12和13所描述的模块。Inception-ResNet-v2使用如图3，16，7，17，18和19所描述的模块。图中的输出尺寸指的是Inception-ResNet-v1的激活向量张量形状。

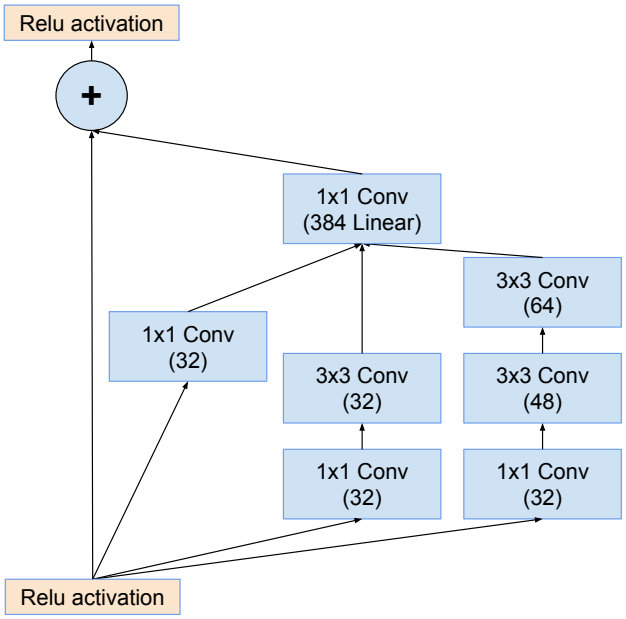

图 16. Inception-ResNet-v2网络的35 × 35网格（Inception-ResNet-A）模块的架构。

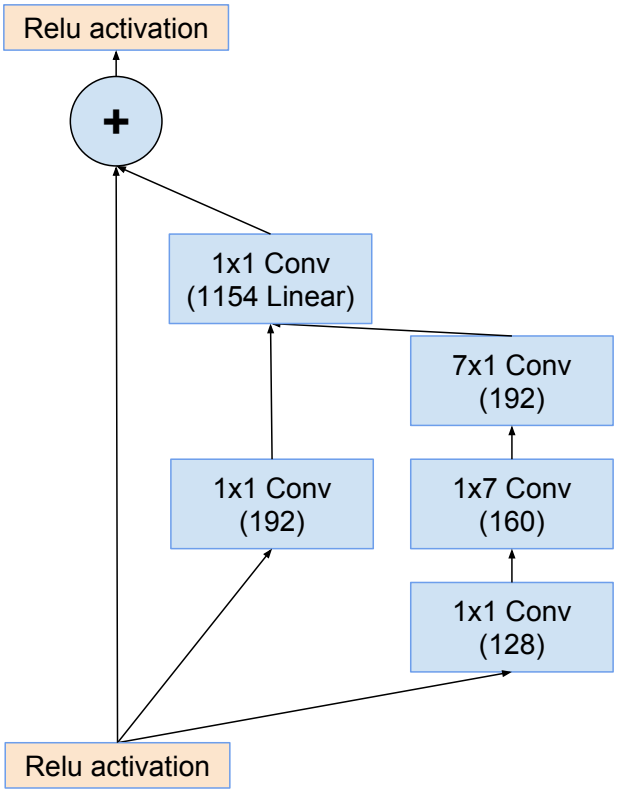

图 17. Inception-ResNet-v2网络的17 × 17网格（Inception-ResNet-B）模块的模式。

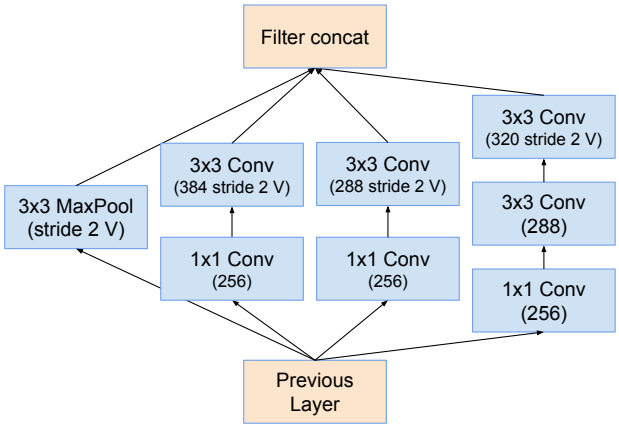

图 18. 用于 17 × 17 到 8 × 8 网格缩减模块的模式。在图 15 中更宽的 Inception-ResNet-v1 网络使用的 Reduction-B 模块。

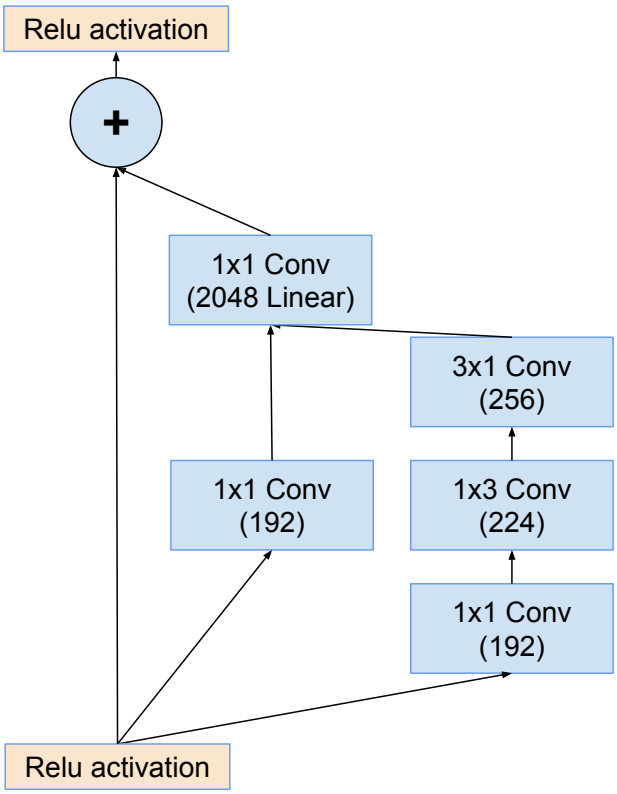

图 19. Inception-ResNet-v2网络的8 × 8网格（Inception-ResNet-C）模块的模式。

3.3. Scaling of the Residuals

我们发现，如果滤波器的数量超过1000，残差变体开始表现出不稳定性，网络在训练初期就已经“死掉”，这意味着在平均池化之前的最后一层在经过数万次迭代后开始只产生零值。无论降低学习率还是在

Network	k	l	m	n
Inception-v4	192	224	256	384
Inception-ResNet-v1	192	192	256	384
Inception-ResNet-v2	256	256	384	384

表 1. 本文提出的三种Inception变体的Reduction-A模块的过滤器数量。文章中列出的四个数字对应于图 7中的四个卷积。

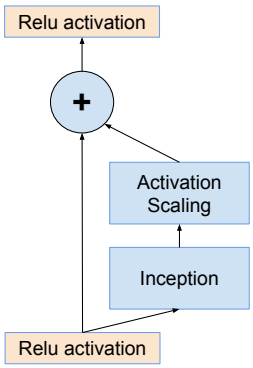

图 20. 将结合Inception-ResNet模块进行缩放的通用模式。我们期望相同的思想在一般的ResNet案例中也是有效的，其中使用任意子网络代替Inception块。缩放块只是通过适当的常数（通常大约是0.1）缩放最后的线性激活。

该层添加额外的批量归一化，都无法防止这种情况发生。

我们发现，在将残差添加到前一层激活之前先将其缩小似乎可以稳定训练。一般来说，我们选择了一些缩放因子在0.1和0.3之间，用于将残差缩小后再添加到累积的层激活中（参见图 20）。

He等人在 [5] 中观察到类似的不稳定性，当残差网络非常深时，他们建议进行两阶段训练，第一阶段是用非常低的学习率进行“热身”，然后是使用高学习率的第二阶段。我们发现，如果滤波器的数量非常高，即使采用非常低（0.00001）的学习率也无法应对不稳定性，而高学习率可能会破坏其效果。我们发现，只需缩放残差要可靠得多。

即使在没有严格必要的情况下，缩放似乎从未损害最终准确度，而是有助于稳定训练。

4. Training Methodology

我们使用 TensorFlow [1] 分布式机器学习系统，在NVidia Kepler GPU 上每个运行20个副本来训练我们的网络。我们较早的实验使用了动量 [13]，衰减率为0.9，

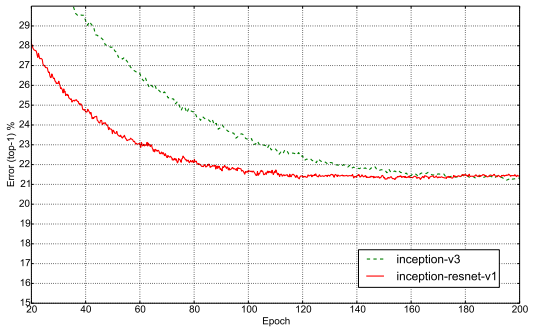

图 21. 在训练纯Inception-v3和一个计算成本类似的残差网络时，Top-1错误率的演变。评估是在ILSVRC-2012验证集的非黑名单图像上进行的单次裁剪。残差模型训练速度更快，但最终准确率略低于传统的Inception-v3。

而我们最佳模型是使用 RMSProp [16]，衰减率为0.9，$\epsilon = 1.0$。我们使用学习率为0.045，每两个周期衰减一次，衰减率为0.94。模型评估是使用随时间计算的参数的运行平均执行的。

5. Experimental Results

首先，我们观察了训练过程中四种变体的top-1和top-5验证错误的演变情况。实验结束后，我们发现我们的持续评估是在一个验证集的子集上进行的，该子集省略了约1700个由于边界框不佳而被列入黑名单的实体。事后发现，这种省略应该仅针对CLSLOC基准进行，但与其他报告（包括我们团队的一些早期报告）相比，结果略显不可比（更为乐观）。在top-1错误方面差异约为0.3%，而在top-5错误方面差异约为0.15%。然而，由于这些差异是一致的，我们认为曲线之间的比较是公平的。

另一方面，我们重新在包含50000张图片的完整验证集上运行了我们的多裁剪和集成结果。最终的集成结果也在测试集上执行，并发送给ILSVRC测试服务器进行验证，以验证我们的调整没有导致过拟合。我们强调，这一最终验证只进行了一次，去年我们仅提交了两次结果：一次是为BN-Inception论文，另一次是在ILSVR-2015 CLSLOC比赛期间提交的，因此我们相信测试集的数据是我们模型泛化能力的真实估计。

最后，我们对Inception的不同版本和Inception-ResNet进行了一些比较。模型Inception-v3和Inception-v4是深度卷积网络，不使用残差连接，而Inception-

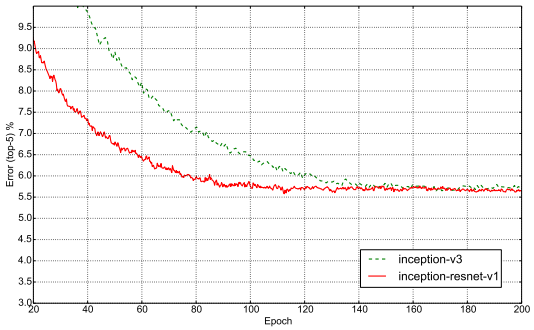

图 22. 在纯Inception-v3与计算成本相似的残差Inception训练过程中的前五个错误演化。评估是在ILSVRC-2012验证集的非黑名单图像上进行的单个裁剪。残差版本训练速度更快，并在验证集上达到略高的最终召回率。

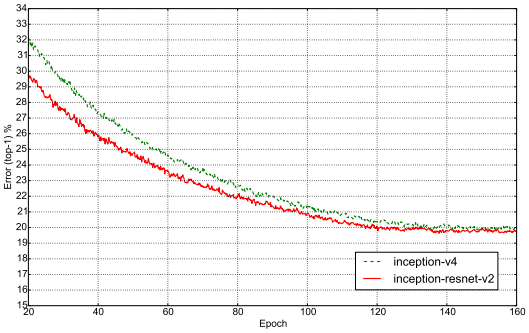

图 23. 在纯Inception-v3和一个计算成本相似的残差Inception训练过程中，Top-1错误率的演变。评估是在ILSVRC-2012验证集的非黑名单图像上的单个裁剪上进行的。残差版本训练速度更快，并且达到了稍微更好的最终准确率，胜过传统的Inception-v4。

Network	Top-1 Error	Top-5 Error
BN-Inception [6]	25.2%	7.8%
Inception-v3 [15]	21.2%	5.6%
Inception-ResNet-v1	21.3%	5.5%
Inception-v4	20.0%	5.0%
Inception-ResNet-v2	19.9%	4.9%

表 2. 单作物 - 单模型实验结果。报道了ILSVRC 2012验证集中未列入黑名单的子集。

ResNet-v1和Inception-ResNet-v2是Inception风格的网络，它们使用残差连接代替滤波器串联。

表 2显示了验证集上各种架构在单模型、单作物的top-1和top-5错误率。

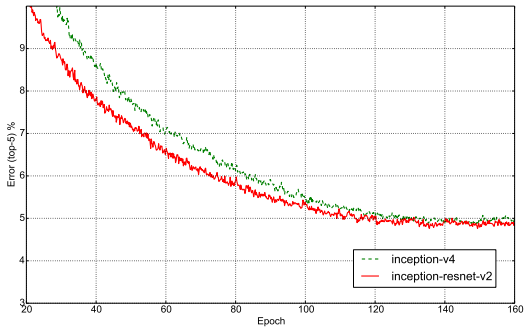

图 24. 在训练纯Inception-v4与类似计算成本的残差Inception过程中的Top-5错误演变。评估是在ILSVRC-2012验证集的非黑名单图像的单个裁剪上进行的。残差版本训练更快，并在验证集上达到稍好的最终召回率。

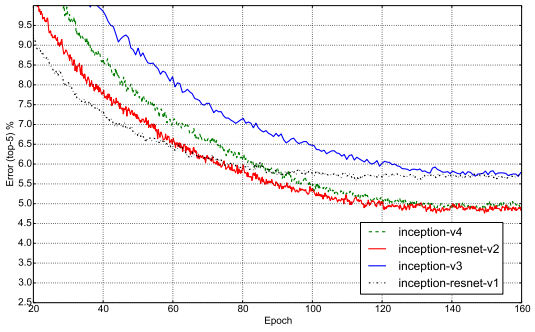

图 25. 所有四种模型（单一模型，单一作物）的前五错误演变。展示由于模型规模较大而带来的改进。尽管残差版本收敛速度更快，但最终准确性似乎主要取决于模型规模。

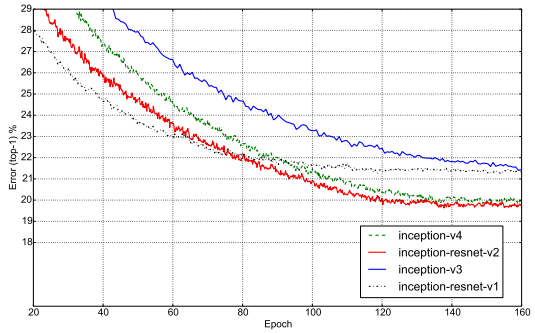

图 26. 所有四个模型（单模型、单裁剪）的Top-1错误率演变。这呈现了与Top-5评估类似的情况。

表 3 显示了各种模型在少量作物（如在[5]中报告的ResNet使用了10个作物，对于Inception变种，我们使用了[14]中描述的12个作物评估）下的性能。

Network	Crops	**Top-1 Error**	**Top-5 Error**
ResNet-151 [5]	10	21.4%	5.7%
Inception-v3 [15]	12	19.8%	4.6%
Inception-ResNet-v1	12	19.8%	4.6%
Inception-v4	12	18.7%	4.2%
Inception-ResNet-v2	12	18.7%	4.1%

表 3. 10/12作物评估 - 单模型实验结果。报告了ILSVRC 2012验证集的全部50000张图像。

Network	Crops	**Top-1 Error**	**Top-5 Error**
ResNet-151 [5]	dense	19.4%	4.5%
Inception-v3 [15]	144	18.9%	4.3%
Inception-ResNet-v1	144	18.8%	4.3%
Inception-v4	144	17.7%	3.8%
Inception-ResNet-v2	144	17.8%	3.7%

表 4. 144种作物评估 - 单模型实验结果。报道了ILSVRC 2012验证集的所有50000张图片。

Network	Models	**Top-1 Error**	**Top-5 Error**
ResNet-151 [5]	6	–	3.6%
Inception-v3 [15]	4	17.3%	3.6%
Inception-v4 + 3× Inception-ResNet-v2	4	16.5%	3.1%

表 5. 在包含144个作物/密集评估的集成结果中。报告了对ILSVRC 2012验证集的全部50000张图像。对于Inception-v4(+Residual)，该集成包括一个纯Inception-v4模型和三个Inception-ResNet-v2模型，并对验证集和测试集进行了评估。测试集表现为3.08%的top-5错误，证明我们没有在验证集上过拟合。

表 4 显示了各种模型单模型性能的评估。对于残差网络，从[5]中报告了密集评估结果。对于Inception网络，使用了[14]中描述的144个作物策略。

表 5 比较了集成结果。对于纯残差网络，从[5]中报告了6个模型的密集评估结果。对于Inception网络，使用了[14]中描述的144个作物策略来集成了4个模型。

6. Conclusions

我们详细介绍了三种新的网络架构：

- Inception-ResNet-v1: 一种混合版本的Inception，其计算成本与文献 [15] 中的Inception-v3 类似。
- Inception-ResNet-v2：一个更昂贵的混合Inception版本，具有显着提高的识别性能。
- Inception-v4：一种纯Inception变体，没有残差连接，具有与Inception-ResNet-v2大致相同的识别性能。

We studied how the introduction of residual connections leads to dramatically improved training speed for the Inception architecture. Also our latest models (with and without residual connections) outperform all our previous networks, just by virtue of the increased model size.

参考文献

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, et al. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems*, pages 1223–1231, 2012.

[3] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *Computer Vision–ECCV 2014*, pages 184–199. Springer, 2014.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[6] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 448–456, 2015.

[7] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with con-

volutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1725–1732. IEEE, 2014.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[9] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[10] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. 2014.

[12] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[13] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 1139–1147. JMLR Workshop and Conference Proceedings, May 2013.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2015.

[16] T. Tieleman and G. Hinton. Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 4, 2012. Accessed: 2015-11-05.

[17] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1653–1660. IEEE, 2014.

[18] N. Wang and D.-Y. Yeung. Learning a deep compact image representation for visual tracking. In *Advances in Neural Information Processing Systems*, pages 809–817, 2013.