**重新思考计算机视觉的盗梦空间架构**

Christian Szegedy谷歌公司

szegedy@google.com

文森特Vanhoucke vanhoucke@google.com

谢尔盖约飞

sioffe@google.com

乔纳森Shlens shlens@google.com

兹比格涅夫•Wojna

伦敦大学学院zbigniewwojna@gmail.com

arXiv: 1512.00567 v3 [cs。2015年12月11日

**摘要**

*卷积网络是最先进的计算机视觉解决方案的核心，适用于各种各样的任务。自2014年以来，深度卷积网络开始成为主流，在各种基准中产生了可观的收益。虽然增加的模型尺寸和计算成本倾向于转化为大多数任务的即时质量收益(只要提供足够的标记数据用于训练)，计算效率和低参数计数仍然是各种用例(如移动视觉和大数据场景)的启用因素。在这里，我们正在探索扩大网络的方法，目的是通过适当的分解卷积和积极的正则化尽可能有效地利用增加的计算。我们基准方法2012年ILSVRC分类挑战验证设置展示实质性收益超过国家的艺术:排名前21.2%和5.6%的单帧评价前5的错误使用网络的计算成本50亿multiply-adds推理和使用少于2500万参数。在4个模型和多作物评估的集成下，我们报告了3.5%的前5位误差和17.3%的前1位误差。*

**1.介绍**

自2012年ImageNet[16]的竞争赢得Krizhevsky等[9],他们的网络“AlexNet”已成功应用于一个更大的各种各样的计算机视觉任务,例如对象检测[5],分割[12],人体姿态估计[22],[8]视频分类,对象跟踪[23],超限分辨[3]。

这些成功激发了新的研究方向，专注于寻找性能更高的卷积神经网络。从2014年开始，通过利用更深更广的网络，网络架构的质量显著提高。VGGNet[18]和GoogLeNet[20]产生了simi-

在2014年ILSVRC[16]分类挑战中表现优异。一个有趣的观察结果是，在许多应用领域中，分类性能的提高往往会转化为显著的质量提高。这意味着深度卷积体系结构的改进可以用于提高大多数其他计算机视觉任务的性能，这些任务越来越依赖于高质量的、可学习的视觉特征。此外，网络质量的提高为卷积网络带来了新的应用领域，在这种情况下，AlexNet功能无法与手工设计的解决方案竞争，例如在检测[4]中生成提案。

尽管VGGNet[18]具有架构简单性这一引人注目的特性，但代价很高:评估网络需要大量的计算。另一方面，GoogLeNet[20]的Inception架构也被设计成即使在内存和计算预算的严格限制下也能表现良好。例如，GoogleNet只使用了500万个参数，比其前身AlexNet使用了6000万个参数减少了12倍。此外，VGGNet使用的参数比AlexNet多3倍。

Inception的计算成本也远低于VGGNet或其性能更高的后继者[6]。这使得在需要以合理的成本处理大量数据的大数据场景[17]、[13]或内存或计算能力天生有限的场景(例如移动视觉设置)中利用Inception网络变得可行。通过对目标内存使用[2]、[15]应用专门的解决方案，或者通过计算技巧[10]优化某些操作的执行，当然可以减轻这些问题的部分。然而，这些方法增加了额外的复杂性。此外，这些方法也可以用于优化先启体系结构，再次扩大效率差距。

尽管如此，先启体系结构的复杂性造成了

1

对网络进行更改更加困难。如果体系结构被天真地扩展，很大一部分计算收益可能会立即损失。此外，[20]没有对导致googleet架构的各种设计决策的影响因素提供清晰的描述。这使得在保持其效率的同时使其适应新的用例变得更加困难。例如，如果认为有必要增加某些《盗启》风格的模型的容量，将所有滤波器组大小的数量翻倍的简单转换将导致计算成本和参数数量增加4倍。在许多实际情况下，这可能被证明是令人望而却步或不合理的，特别是在相关的收益不大的情况下。在这篇文章中，我们首先描述了一些一般的原则和优化思想，这些被证明对以有效的方式扩大卷积网络是有用的。虽然我们的原则并不局限于Inception类型的网络，但是它们在那个上下文中更容易观察到，因为Inception风格构建块的一般结构足够灵活，可以自然地合并那些约束。这是通过大量使用初始模块的降维和并行结构来实现的，这允许减轻结构变化对附近组件的影响。但是，这样做需要谨慎，因为应该遵循一些指导原则来保持模型的高质量。

**2.总体设计原则**

在这里，我们将描述一些基于使用卷积网络的各种架构选择的大规模实验的设计原则。在这一点上，以下原则的效用是推测性的，未来需要更多的实验证据来评估它们的准确性和有效性领域。尽管如此，对这些原则的严重偏差往往会导致网络质量的恶化，而检测到这些偏差的固定情况会导致总体架构的改进。

1.避免表示瓶颈，特别是在网络的早期。前馈网络可以用一个从输入层到分类器或回归器的无环图来表示。这为信息流定义了一个明确的方向。对于任何分离输入和输出的切割，都可以访问通过切割传递的信息量。我们应该避免极限压缩带来的瓶颈。通常情况下，表示大小应该从输入到输出逐渐减小，直到达到当前任务的最终表示。从理论上讲，不能仅仅通过表征的维度来评价信息内容，因为它忽略了相关结构等重要因素;维-

Ity仅仅提供了对信息内容的粗略估计。

2.高维表示更容易在网络内进行局部处理。在一个卷积网络中增加每个贴图的激活量可以允许更多的解缠特征。由此产生的网络将训练得更快。

3.空间聚合可以在低维嵌入上进行，而不会损失太多或任何表示能力。例如，在执行更广泛的卷积(例如3 × 3)之前，可以在空间聚合之前降低输入表示的维数，而不会产生严重的不良影响。我们假设，其原因是，如果输出用于空间聚合上下文，则相邻单元之间的强相关性导致在降维过程中信息丢失更少。考虑到这些信号应该很容易压缩，降维甚至可以促进更快的学习。

4.平衡网络的宽度和深度。通过平衡每级滤波器的数量和网络的深度，可以达到网络的最佳性能。增加网络的宽度和深度可以促进更高质量的网络。然而，如果同时增加这两种计算量，就可以达到对固定计算量的最优改进。因此，计算预算应在网络的深度和宽度之间平衡分配。

尽管这些原则可能是有意义的，但使用它们来提高开箱即用的网络质量并不简单。其想法是，只有在模棱两可的情况下才明智地使用它们。

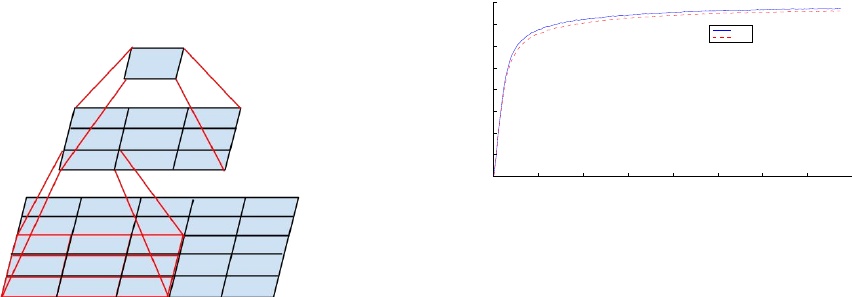
**3.利用大滤波器分解卷积**

**大小**

google网络[20]的大部分原始收益来自于对降维的大量使用。这可以看作是一个特殊的情况，分解卷积在计算上有效的方式。以1 × 1卷积层后面跟着3 × 3卷积层为例。在视觉网络中，期望近端激活的输出是高度相关的。因此，我们可以预期，它们的激活可以在聚集之前被减少，这应该会导致类似的表达性局部表示。

在这里，我们探索在各种设置中分解卷积的其他方法，特别是为了提高解决方案的计算效率。由于初始网络是完全卷积的，每个权重对应

线性与ReLU激活因子分解



0.8

0.7

ReLU线性

0.6

顶级−1准确性

0.5

0.4

0.3

0.2

0.1

0 0

0.5

1

1.5

2迭代

2.5

3.

3.5

4

6

x 10

图2。两个初始模型之间的几个控制实验之一，其中一个使用分解成线性+ ReLU层，另一个使用两个ReLU层。后386万歌剧-

图1所示。迷你网络替代5 × 5卷积。

一次激活一次乘法。因此，计算成本的任何减少都会导致参数数量的减少。这意味着，通过适当的因式分解，我们最终可以得到更多的解缠参数，从而得到更快的训练。此外，我们可以使用节省的计算和内存来增加我们网络的过滤器组大小，同时保持我们在单个计算机上训练每个模型副本的能力。

**3.1。分解成更小的卷积**

使用较大的空间滤波器(例如5 × 5或7 × 7)的卷积在计算方面往往异常昂贵。例如，在一个包含m个滤波器的网格上，对n个滤波器进行5 × 5卷积的计算成本是使用相同数量滤波器的3 × 3卷积的25/9 = 2.78倍。当然，5×5过滤器可以捕获早期层中更远处单元激活之间的信号依赖关系，所以减少过滤器的几何尺寸需要付出很大的表现力代价。然而，我们可以问，一个5 × 5的卷积是否可以被一个具有更少参数的多层网络代替，但具有相同的输入大小和输出深度。如果我们放大5×5卷积的计算图表,我们可以看到,每个输出看起来就像一个小型全连通网络滑动/ 5×5瓦/输入(见图1)。由于我们构造一个视觉网络,似乎自然再次利用平移不变性和取代完全连接组件通过一个两层的卷积archi - tecture:第一层是一个3×3卷积,第二个是一个完全连接层的3×3输出网格的第一层(见图1)。滑动这个小型网络输入激活网格可以归结为替换5×5卷积与两层3×3卷积(比较图4和5)。

这种设置通过在相邻贴图之间共享权重明显地减少了参数计数。分析前

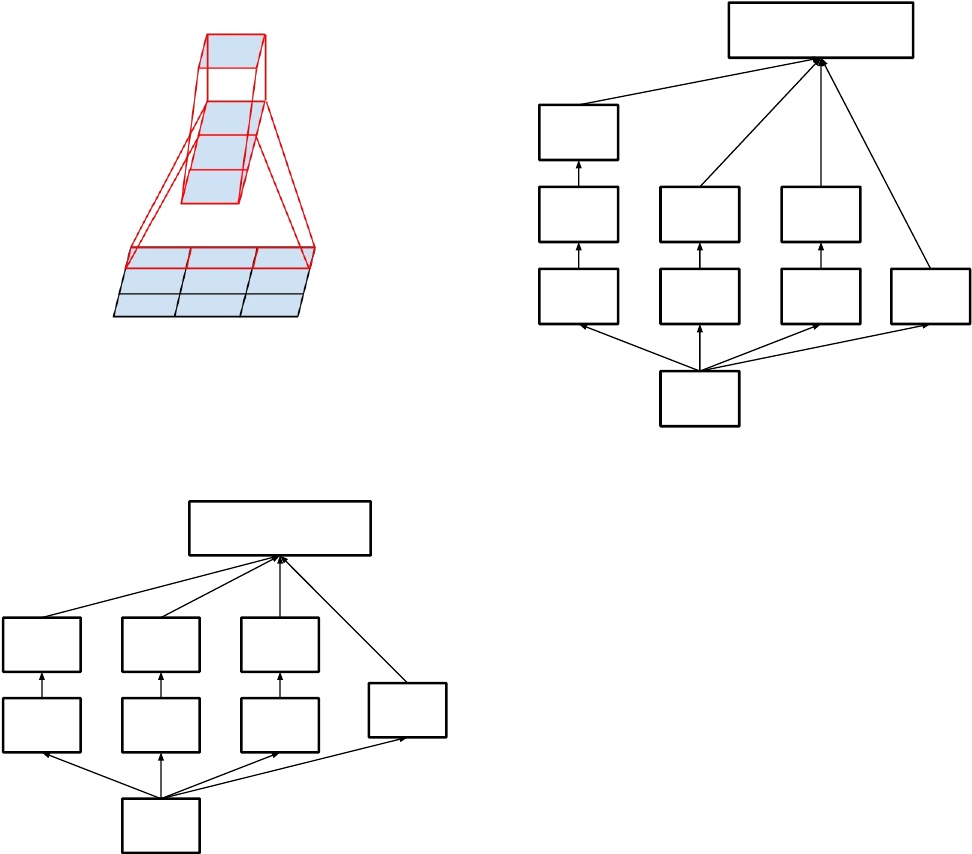
在验证集上，前者达到76.2%，后者达到77.2%的top-1 Accuracy。

我们可以假设n = αm，也就是说，我们想用一个常数α因子来改变活化数/单位。由于5 × 5卷积是聚合的，所以α通常略大于1(在GoogLeNet中约为1.5)。将5 × 5层替换成两层，这样的扩展似乎是合理的

√两步:每步增加α的过滤器数量。为了简化我们的估计，选择α = 1(不展开)，如果我们天真地滑动一个网络，而不重复使用相邻网格块之间的计算，我们将增加计算成本。滑动这个网络可以用两个3 × 3的卷积层来表示，这些卷积层重复使用相邻贴图之间的激活。这样，我们就得到了一张网 9+925 ×计算量的减少，通过这种因式分解得到28%的相对增益。在每个单元的激活计算中，每个参数只使用一次，因此参数计数的保存方法是完全相同的。然而，这种设置提出了两个普遍的问题:这种替换会导致任何表达能力的丧失吗?如果我们的主要目标是因式分解计算的线性部分，难道不建议在第一层保持线性激活吗?我们进行了几个控制实验(例如，见图2)，在分解的所有阶段，使用线性激活总是不如使用校正线性单元。我们将这种增益归因于网络可以学习的变异空间的增强，特别是当我们对输出激活[7]进行批处理规范化时。当对降维组件使用线性激活时，可以看到类似的效果。

**3.2。非对称卷积的空间分解**

以上结果表明，具有较大3 × 3a滤波器的卷积可能通常并不有用，因为它们总是可以被简化为3 × 3卷积序列



[20]引入了辅助分类器的概念来提高深度网络的收敛性。的起源,

过滤器Concat

3 x3

3 x3

3 x3

1 x1

1 x1

1 x1

池

1 x1

图3。迷你网络替代3 × 3卷积。该网络的底层由3 × 1卷积组成，有3个输出单元。

基地

图5。初始模块，将每个5 × 5卷积替换为两个3 × 3卷积，如第2节原则3所示。

过滤器Concat

5 x5 1 x1

3 x3 1 x1

1 x1池

1 x1

后跟一个n×1卷积,计算节省费用大幅增加随着n(参见图6)。在实践中,我们发现使用这对早期层分解并不奏效,但它给很好的结果介质网格大小(m×m特征图,米范围12 - 20)。在这个层面上，使用1 × 7卷积和7 × 1卷积可以获得非常好的结果。

**4.辅助分类器的用途**

基地

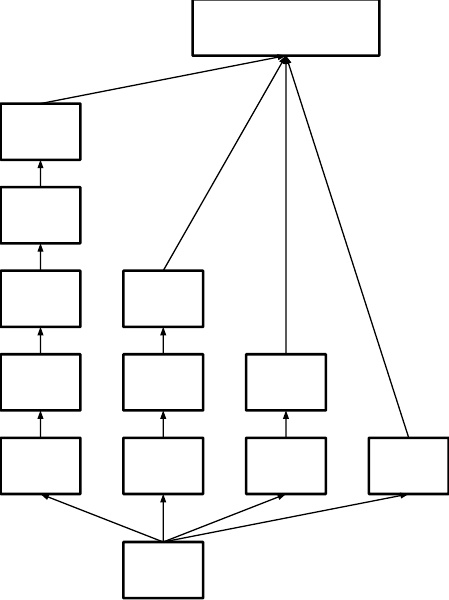
图4。[20]中描述的初始模块。

层。我们仍然可以问是否应该把它们分解成更小的，例如2×2卷积。然而，事实证明，使用非对称卷积(如n × 1)可以得到比2 × 2更好的结果。例如使用一个3×1卷积之后1×3卷积等价于滑动两层网络与相同的接受域3×3卷积(参见图3)。还是两层的解决方案是便宜33%相同数量的输出过滤器,如果输入和输出滤波器的数量是相等的。相比之下，将一个3 × 3卷积分解成两个2 × 2卷积只节省11%的计算量。

从理论上讲，我们可以更进一步说，我们可以用1 × n卷积来代替任何n × n的卷积

最终的动机是将有用的梯度推到较低的层，使它们立即有用，并通过对抗非常深的网络中的消失梯度问题来提高训练过程中的收敛性。Lee等人[11]也认为辅助分类器可以促进更稳定的学习和更好的收敛。有趣的是，我们发现辅助分类器在训练初期并没有提高收敛性:在两种模型达到较高的准确率之前，有侧头和没有侧头的网络训练进程看起来几乎相同。在训练接近尾声时，有辅助分支的网络开始超过没有辅助分支的网络，并达到一个略高的平台。

[20]在网络的不同阶段也使用了两个侧头。下辅助支路的移除对网络的最终质量没有任何不利影响。连同前一段中先前的观察



过滤器Concat

资料片

1 xn



1 xn

1 xn

1 x1



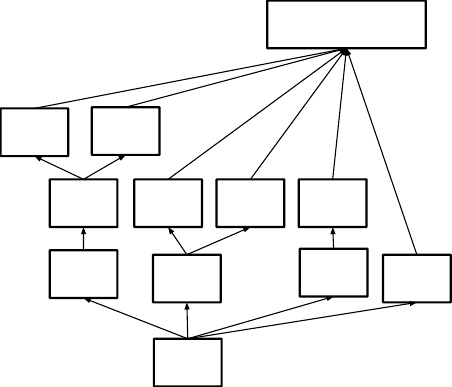
基地

图6。初始模块分解后的n × n个卷积。在我们提议的架构中，我们选择n = 7作为17 × 17的网格。(过滤尺寸采用原则3)

。

图，这意味着最初[20]的假设，即这些分支帮助进化低级特征，很可能是错位的。相反，我们认为辅助分类器作为正则化。如果分支是批标准化[7]或有dropout层，则网络的主分类器性能更好，这一事实支持了这一点。这也为批标准化充当正则化器的猜想提供了一个弱的支持证据。

**5.有效减小网格尺寸**



过滤器Concat



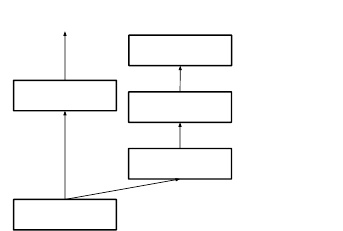
3 x1

1 x1



基地

图7。初始模块与扩展的滤波器组输出。这种架构用于最粗糙的(8 × 8)网格，以促进高维表示，正如第2节的原则2所建议的那样。我们只在最粗糙的网格上使用这个解决方案，因为在这里产生高维稀疏表示是最关键的，因为与空间聚合相比，局部处理(1 × 1卷积)的比例增加了。



…

1 x1x1024

完全连接

8 x8x1280



1 x1卷积

《盗梦空间》

5 x5x768

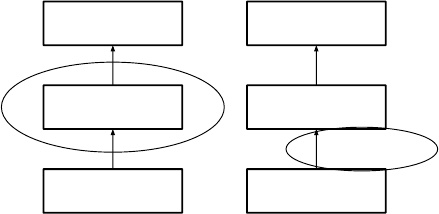
平均池化，步幅3

17 x17x768

图8。辅助分类器上的最后17×17层。在侧头中对层进行批归一化[7]，可以在top-1精度中获得0.4%的绝对增益。下面的轴显示执行的迭代次数，每个批处理大小为32。

传统的卷积网络使用一些池化操作来减小特征地图的网格大小。为了避免表征瓶颈，在应用最大池或平均池之前，对网络过滤器的激活维进行了扩展。例如，从带有k个滤波器的d×d网格开始，如果我们想到达a d2 × d2 带有2k滤波器的网格，我们首先需要用2k滤波器计算一个stride-1卷积，然后应用一个额外的池化步骤。这意味着在使用2d的更大的网格上，总的计算代价主要是昂贵的卷积2k2 操作。一种可能是使用卷积切换到池化，从而产生2( d2 )2k2

将计算成本降低四分之一。然而，当表示的整体维度下降到( d2 )2k会导致表达性较差的网络(见图9)。与此相反，我们建议使用另一种变体来进一步降低计算成本，同时消除表征瓶颈。(见图10)。我们可以使用两个平行的stride 2块:P和c。P是一个池化层(或平均池化或最大池化)的激活，它们都是stride 2，其中的滤波器组连接如图10所示。



17 x17x640

《盗梦空间》17 x17x320

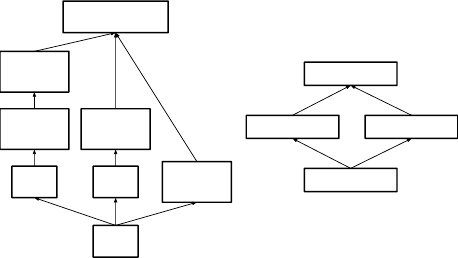
池35 x35x320

17 x17x640

池35 x35x640

《盗梦空间》35 x35x320

图9。减少网格大小的两种替代方法。左边的解决方案违反了原则1，即不引入第2节中的表征瓶颈。右边的版本在计算上要贵3倍。



过滤器Concat

3 x3

步2

3 x3

步1

1 x1

3 x3

步2

1 x1

池步2

17 x17x640 concat

17 x17x320 17 x17x320

conv 池35 x35x320

基地

图10。初始模块，减少网格大小，同时扩大过滤器组。它既便宜又避免了原则1所建议的表征瓶颈。右边的图表示相同的解决方案，但是是从网格大小而不是操作的角度来看的。

**6.Inception-v2**

在这里，我们将以上几点联系起来，并在ILSVRC 2012分类基准上提出一个性能更好的新体系结构。我们的网络布局如表1所示。请注意，我们已经将传统的7 × 7卷积分解为3 × 3卷积，其原理与3.1节中描述的相同。对于网络的初始部分，我们有3个35 × 35的传统初始模块，每个模块有288个过滤器。使用第5节所述的网格缩减技术，将其简化为带有768个滤波器的17 × 17网格。接下来是5个被分解的初始模块实例，如图5所示。用图10所示的网格缩减技术将其简化为8 × 8 × 1280网格。在最粗糙的8 × 8级别上，我们有如图6所示的两个Inception模块，每个贴图的串联输出滤波器组大小为2048。该网络的详细结构，包括初始模块中过滤库的大小，在补充材料中给出，在此提交的tar文件中的model.txt中给出。

|  |  |  |
| --- | --- | --- |
| **类型** | **补丁大小/步幅或备注** | **输入的大小** |
| conv | 3×3/2 | 299×299×3 |
| conv | 3×3/1 | 149×149×32 |
| conv垫 | 3×3/1 | 147×147×32 |
| 池 | 3×3/2 | 147×147×64 |
| conv | 3×3/1 | 73×73×64 |
| conv | 3×3/2 | 71×71×80 |
| conv | 3×3/1 | 35××192 |
| 3×《盗梦空间》 | 如图5所示 | 35××288 |
| 5×《盗梦空间》 | 如图6所示 | 17×17×768 |
| 2×《盗梦空间》 | 如图7所示 | 8×8×1280 |
| 池 | 8×8 | 8 × 8 × 2048 |
| 线性 | 分对数 | 1 × 1 × 2048 |
| softmax | 分类器 | 1 × 1 × 1000 |

表1。拟议的网络架构的概要。每个模块的输出大小是下一个模块的输入大小。我们正在使用图10所示的减少技术的变体来减少初始块之间的网格大小。我们用0-padding标记了卷积，这是用来保持网格大小的。0-padding也用于不减少网格大小的初始模块中。所有其他图层都不使用填充。根据第2节中的原则4来选择不同的滤波器组尺寸。

然而，我们已经观察到，只要遵循第2节的原则，网络的质量相对稳定。虽然我们的网络有42层，但是我们的计算成本只比googleet高2.5左右，它仍然比VGGNet高效得多。

**7.通过标签平滑的模型正则化**

在此，我们提出了一种通过估计训练过程中标签退出的边缘效应来规范分类器层的机制。

对于每个训练例x，我们的模型计算每个标签k∈{1的概率…K}: p (K | x) =

exp (zk )

PKi = 1 exp(子)。这里zi是对数或非标准化对数概率。考虑这个训练示例的标签q(k|x)上的ground-truth分布，归一化后使之 P q (k | x) = 1。为了简短起见，我们省略对k的依赖关系

p和q在例子x上，我们定义P例如，损失为交叉熵:' =−Kk=1 log(p(k))q(k)。最小化这个等价于最大化标签的期望对数似然，其中标签根据其ground-truth分布q(k)被选择。交叉熵损失相对于对数z是可微的k 从而可以用于深度模型的梯度训练。梯度有一个相当简单的形式:∂z∂' = p(k)−q(k)，在−1和1之间。k

考虑只有一个ground-truth label y的情况，因此对于所有的k6 = y, q(y) = 1, q(k) = 0。在这种情况下，

最小化交叉熵等价于最大化正确标签的对数似然。对于一个带有y标签的特殊例子x，当q(k) = δk,y时，对数似然是最大的，其中δk,y是Dirac delta，当k = y时等于1，否则为0。这个最大值对于有限的z是无法达到的k 但是如果z接近y 吗?zk 对于所有的k 6= y，也就是说，如果ground-truth标签对应的logit比其他所有的logit都大。然而，这可能会导致两个问题。首先，它可能导致过拟合:如果模型学习为每个训练实例的ground- truth标签分配全概率，它不能保证泛化。其次，它促使最大logit与所有其他logit之间的差异变大，再加上有界梯度∂z∂'，就降低了abil- k

模型的适应性。直觉上，这是因为模型对其预测过于自信。

我们提出了一种机制，以鼓励模型不那么自信。虽然如果目标是最大限度地提高训练标签的对数可能性，这可能不是理想的，但它确实使模型规范化，使其更具有适应性。方法很简单。考虑标签u(k)上的分布，独立于训练例子x，以及一个平滑参数?。对于具有ground-truth标签y的训练示例，我们替换标签分布q(k|x) = δk,y 与

FORMULA

哪个是原始真实分布q(k|x)和固定分布u(k)的混合物，权重为1−?分别和?。这可以看作是得到的标签k的分布如下:首先，设为ground-真值标签k = y;然后，用概率?，将k替换为从分布u(k)中抽取的样本。我们建议使用标签上的先验分布为u(k)。在我们的实验中，我们使用均匀分布u(k) = 1/ k，所以

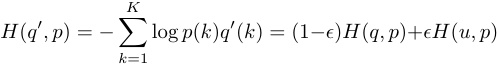
那



我们将ground-truth标签分布中的这种变化称为标签平滑正则化(label-smoothing regularization，简称LSR)。

请注意，LSR实现了防止最大logit比所有其他logit大得多的预期目标。事实上，如果发生这种情况，那么单个q(k)将趋于1，而其他所有q(k)将趋于0。这将导致与q有很大的交叉熵0(k)因为不像q(k) = δk,y,所有问0(k)下界为正。

考虑交叉熵可以得到LSR的另一种解释:



因此，LSR等价于用一对交叉熵损失H(q, p)和H(u, p)替换单个交叉熵损失H(q, p)。

第二个损失惩罚预测的标签分布p与先验的u的偏差，使用相对权重 1−??。注意，由于H(u, p) = D，这个偏差可以被KL散度等价地捕获KL(ukp) + H(u)， H(u)是固定的。当u是均匀分布时，H(u, p)是预测分布p与均匀分布的不同程度的度量，这也可以通过负熵−H(p)来度量(但不是等价的);我们还没有尝试过这种方法。

在ImageNet实验中，K = 1000个类，我们使用u(K) = 1/1000和?= 0.1。对于ILSVRC 2012，我们发现top-1误差和top-5误差都有大约0.2%的绝对改进(参见表3)。

**8.培训方法**

我们使用TensorFlow[1]分布式机器学习系统，利用随机梯度训练我们的网络，使用50个副本，每个副本在NVidia Kepler GPU上运行，批处理大小为32，用于100个纪元。我们早期的实验使用动量[19]衰减0.9，而我们的最佳模型是使用RMSProp[21]衰减0.9和?= 1.0。我们使用0.045的学习速率，每两个历元衰减一次使用指数速率0.94。此外，阈值为2.0的梯度裁剪[14]被发现有助于稳定训练。模型评估是使用随时间计算的参数的运行平均值来执行的。

**9.低分辨率输入的性能**

视觉网络的一个典型用例是用于检测的后分类，例如在Multibox[4]上下文中。这包括对包含特定背景的单个对象的图像的相对小块进行分析。任务是决定补丁的中心部分是否对应于某个对象，如果对应，则确定该对象的类。挑战在于，这些物体往往相对较小且分辨率较低。这就提出了如何正确处理低分辨率输入的问题。

普遍的看法是，采用更高分辨率的接受域的模型往往会显著提高识别性能。然而，区分第一层感受场分辨率的提高和模型容量和计算量的增大的影响是重要的。如果我们只是改变输入的分辨率，而不进一步调整模型，那么我们最终会使用计算成本低得多的模型来解决更困难的任务。当然，这是很自然的，这些解决方案已经松散了，因为减少了计算工作量。为了做出准确的评估，该模型需要分析模糊的暗示，以便能够“幻觉”出细节。这在计算上是昂贵的。因此，问题仍然是:多少

|  |  |
| --- | --- |
| **接受域大小** | **顶级精度(单帧)** |
| 79×79 | 75.2% |
| 151×151 | 76.4% |
| 299×299 | 76.6% |

表2。当接受域的大小不同，但计算代价不变时，识别性能的比较。

如果计算量保持不变，更高的输入分辨率是否有帮助?确保持续努力的一个简单方法是，在分辨率较低的输入情况下，减少前两层的步伐，或者简单地删除网络的第一个池化层。

为此，我们进行了以下三个实验:

1.299 × 299感受野，第一层后步幅为2，最大池化。

2.感受野为151 × 151，步幅为1，第一层后最大池化。

3.79 × 79感受野，步幅为1，第一层后无池化。

所有三个网络的计算成本几乎相同。虽然第三种网络稍微便宜一些，但汇聚层的成本是边际的，并且(在网络总成本的1%以内)。在每种情况下，训练网络直到收敛，并在ImageNet ILSVRC 2012分类基准的验证集上测量其质量。结果见表2。虽然低分辨率的网络需要更长的时间来训练，但最终结果的质量与高分辨率的网络相当接近。

然而，如果根据输入分辨率单纯地减小网络规模，那么网络的性能就会差得多。然而，这将是一个不公平的比较，因为我们将比较一个16倍便宜的模型在一个更困难的任务。

表2的这些结果还表明，可以考虑在R-CNN[5]背景下使用专用的高成本低分辨率网络来处理较小的目标。

**10.实验结果与比较**

表3显示了第6节中描述的我们提出的体系结构(Inception- v2)的识别性能的实验结果。每一行“盗梦空间-v2”都显示了累积变化的结果，包括高亮显示的新修改和所有之前的修改。标签平滑方法参见第7节。7 × 7分解包括将前7 × 7卷积层分解为3 × 3卷积层序列的更改。BN-auxiliary是指其中的版本

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **网络** | **(错误** | **五大错误** | | **Bn运维成本** |
| GoogLeNet [20] | 29% | 9.2% | | 1.5 |
| BN-GoogLeNet | 26.8% |  | - | **1.5** |
| BN-Inception [7] | 25.2% | 7.8 | | 2.0 |
| Inception-v2 | 23.4% |  | - | 3.8 |
| Inception-v2 RMSProp | 23.1% | 6.3 | | 3.8 |
| Inception-v2标签平滑 | 22.8% | 6.1 | | 3.8 |
| 盗梦空间v2因子分解7 × 7 | 21.6% | 5.8 | | 4.8 |
| Inception-v2 BN-auxiliary | **21.2%** | **5.6%** | | 4.8 |

表3。单茬试验结果比较了累积效应对各种影响因素的影响。我们将我们的数字与在al[7]发表的关于Ioffe的最佳单一作物推断进行了比较。对于“盗梦空间-v2”行，更改是累积的，后续的每一行除了先前的更改外，还包括新的更改。最后一行指的是所有的更改，我们在下面将其称为“盗梦空间v3”。不幸的是，他等人[6]只报告了10种作物的评价结果，但没有报告单一作物的结果，报告在下面的表4中。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **网络** | **作物评估** | | **五大错误** | | **(错误** |
| GoogLeNet [20] | 10 | |  | - | 9.15% |
| GoogLeNet [20] | 144 | |  | - | 7.89% |
| VGG [18] |  | - | 24.4% | | 6.8% |
| BN-Inception [7] | 144 | | 22% | | 5.82% |
| PReLU [6] | 10 | | 24.27% | | 7.38% |
| PReLU [6] |  | - | 21.59% | | 5.71% |
| Inception-v3 | 12 | | 19.47% | | 4.48% |
| Inception-v3 | 144 | | **18.77%** | | **4.2%** |

表4。单模型、多作物试验结果比较了各种影响因素的累积效应。我们将我们的数字与在ILSVRC 2012分类基准上发表的最佳单模型推理结果进行了比较。

辅助分类器的全连通层也是批归一化的，而不仅仅是卷积。我们将表3最后一行中的模型称为Inception-v3，并评估其在多作物和集成设置中的性能。

我们所有的评估都是在ILSVRC-2012验证集上的48238个非黑名单示例上完成的，如[16]所建议的。我们也评估了所有50000个例子，结果在前5个错误中大约差0.1%，在前1个错误中大约差0.2%。在本文的下一个版本中，我们将在测试集上验证我们的集成结果，但是在我们最后一次对BN的评估时——在spring的Inception中，[7]表明测试和验证集错误往往非常相关。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **网络** | **模型评估** | **作物评估** | | **(错误** | | **五大错误** |
| VGGNet [18] | 2 |  | - | 23.7% | | 6.8% |
| GoogLeNet [20] | 7 | 144 | |  | - | 6.67% |
| PReLU [6] | - |  | - |  | - | 4.94% |
| BN-Inception [7] | 6 | 144 | | 20.1% | | 4.9% |
| Inception-v3 | 4 | 144 | | **17.2%** | | **3.58%**∗ |

表5所示。比较多模型、多作物报告结果的集成评价结果。并与ILSVRC 2012分类基准上发布的最佳集成推理结果进行了比较。 ∗所有的结果，但报告的前5位集成结果在验证集上。集成在验证集上产生了3.46%的前5位误差。

**11.结论**

我们提供了几个设计原则来扩展卷积网络，并在Inception架构的上下文中对它们进行了研究。这种指导可以导致高性能的视觉网络，与更简单、更单片的架构相比，具有相对适中的计算成本。在ILSVR 2012分类中，我们的最高质量版本的Inception-v3达到了单个作物评估的21.2%、top-1和5.6%的top-5错误，建立了一个新的艺术状态。与Ioffe等人[7]中描述的网络相比，这在计算成本上增加了相对适中的(2.5×)。仍然比最好的出版我们的解决方案使用更少的计算结果基于密集的网络:我们的模型的结果优于他等[6]——削减前5名(排名前)(14%)相对误差25%,分别,而被便宜6倍计算,少使用至少5次参数(估计)。我们对4个Inception-v3模型的集成达到了3.5%，其中多作物评估达到了3.5%的前5位误差，这代表了比最佳发表结果减少了25%以上，几乎是ILSVRC 2014获奖GoogLeNet集成的一半。

我们还证明了在接受场分辨率低至79×79的情况下可以获得高质量的结果。这可能被证明是有用的系统，以检测相对较小的目标。我们已经研究了如何在神经网络内部分解卷积和积极降维可以导致网络具有相对较低的计算成本，同时保持高质量。更低的参数计数和附加的正则化与批归一化辅助分类器和标签平滑的组合允许在相对适中的规模的训练集上训练高质量的网络。

**参考文献**

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghe- mawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia，

R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Man´e, 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. Vi´egas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu，和X. Zheng。张量-流:基于异构系统的大规模机器学习，2015。软件可从tensorflow.org获得。

参考文献:陈永平，陈永平，陈永平，陈永平，陈永平。用哈希技巧压缩神经网络。第32届机器学习国际会议论文集，2015。

[1]董晓东，罗孝华，何凯，唐晓东。学习一种用于图像超分辨率的深度卷积网络。在Computer Vision-ECCV 2014中，第184-199页。施普林格,2014年。

D. Erhan, C. Szegedy, A. Toshev, D. angelov。利用深度神经网络进行可扩展的目标检测。计算机视觉与模式识别(CVPR)IEEE 2014。

R. Girshick, J. Donahue, T. Darrell, J. Malik。丰富的特征层次精确的对象检测和语义分割。计算机视觉与模式识别(CVPR)， 2014。

何凯，张旭东，任士生，孙建平。深入研究整流器:在图像网分类上超越人类水平的性能。*arXiv:1502.01852, 2015。*

[7] S. Ioffe和C. Szegedy。批归一化:通过减少内部协变量偏移来加速深度网络训练。第32届机器学习国际会议论文集，448-456页，2015。

A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei。基于卷积神经网络的大规模视频分类。计算机视觉与模式识别(CVPR)IEEE 2014。

A. Krizhevsky, I. Sutskever和G. E. Hinton。基于深度卷积神经网络的Imagenet分类。《神经信息处理系统进展》，第1097-1105页，2012年。

[10]拉文。卷积神经网络的快速算法。*arXiv:1509.09308, 2015。*

[11]彭译葶。谢淑华，张志明，张志明。深度监督网。*arXiv:1409.5185, 2014。*

J. Long, E. Shelhamer和T. Darrell。完全卷积网络语义分割。《计算机视觉与模式识别学报》，2015年第3期。

M. C. Stumpe, V. Shet, S. Arnoud, L. Yatziv。对街景店面进行细粒度分类的本体监督。《IEEE计算机视觉与模式识别会议论文集》，2015年第1693-1702页。

R. Pascanu, T. Mikolov和Y. Bengio。关于训练循环神经网络的困难。*预印本*

[15] D. C. Psichogios和L. H. Ungar。Svd-net:自动选择网络结构的算法。*IEEE神经网络学报/ IEEE神经网络委员会出版，5(3):513-515,1993。*

J. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein等。Imagenet大规模视觉识别挑战。2014.

F. Schroff, D. Kalenichenko和J. Philbin。Facenet:人脸识别和聚类的统一嵌入。*arXiv:1503.03832, 2015。*

K. Simonyan和A. Zisserman。用于大规模图像识别的深度卷积网络。*arXiv预印本arXiv:1409.1556, 2014。*

I. Sutskever, J. Martens, G. Dahl，和G. Hinton。关于初始化和动量在深度学习中的重要性。第30届机器学习国际会议论文集(ICML-13)，第28卷，1139-1147页。2013年5月，JMLR研讨会论文集。

1 .刘文杰，刘文杰，刘文杰，刘文杰，刘文杰，刘文杰，刘文杰，刘文杰，刘文杰，刘文杰。深入研究卷积。在计算机视觉和模式识别的IEEE会议论文集，第1-9页，2015。

T. Tieleman和G. Hinton。将梯度除以其近期大小的运行平均值。COURSERA:机器学习的神经网络，2012年4月。访问:2015 - 11-05。

A. Toshev and C. Szegedy。深度姿态:通过深度神经网络估计人体姿态。计算机视觉与模式识别(CVPR)IEEE 2014。

[23] N. Wang和d.y。杨。学习用于视觉跟踪的深度压缩图像表示。神经信息处理系统进展，809-817页，2013。