简介

VGG 的论文地址为《Very Deep Convolutional Networks for Large-Scale Visual Recognition》。论文探讨了卷积神经网络的深度对于图片识别精确度的影响。此篇论文也是为了解决网络中深度的问题。

下面的文章既有对于原文的理解，也要遇到相关的问题查找的资料，想要了解具体的细节可以查看每一章节的参考部分，或者文末的总参考部分。文中有一些自己的理解，应为接触 DN 不久，免不了会出现错误，阅读时如果有疑问可以留下你的评论，不胜感谢。

VGG 结构

VGG 多种结构示意如下：

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

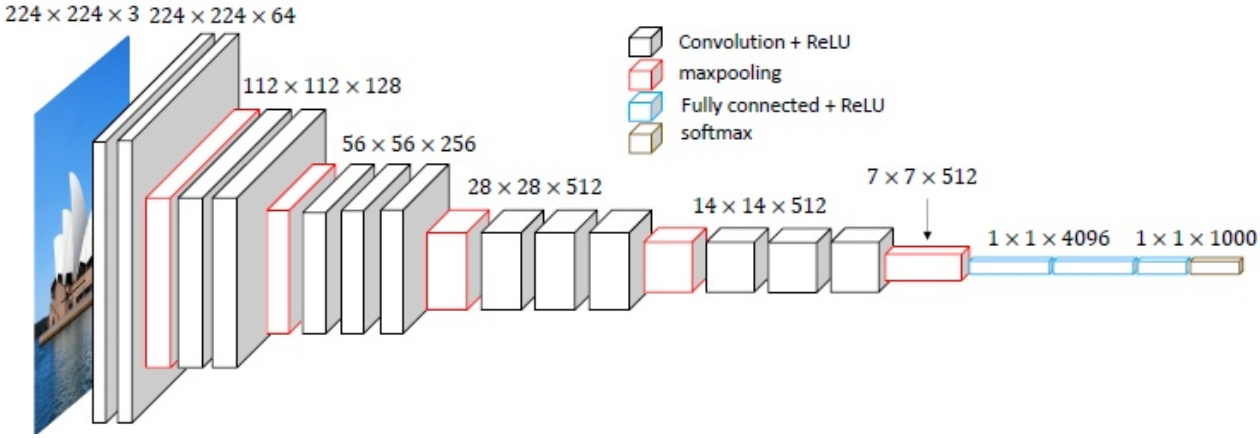
上图列出了从较浅的 VGG11 到 VGG19 不同的结构，其中的 11 或者 19指的是具有权重参数的层，如卷积层（conv layers）和全连接层（FC），不包括池化层，Dropout 和激活函数层（ReLU）。

所有卷积层后面都跟有非线性激活函数层，如 ReLU。每经过一个maxpool 层 filter 的个数就翻倍，如 ConvNet A：input -> conv3-64 -> max pool -> conv3-128(翻倍) -> conv3-256(两个) -> max pool -> conv3-512。

之后再接三个全连接层，其中前两个的结构为 FC -> ReLU -> Dropout ，最后一个只有 FC 。

最后一个是 softmax 用于分类。

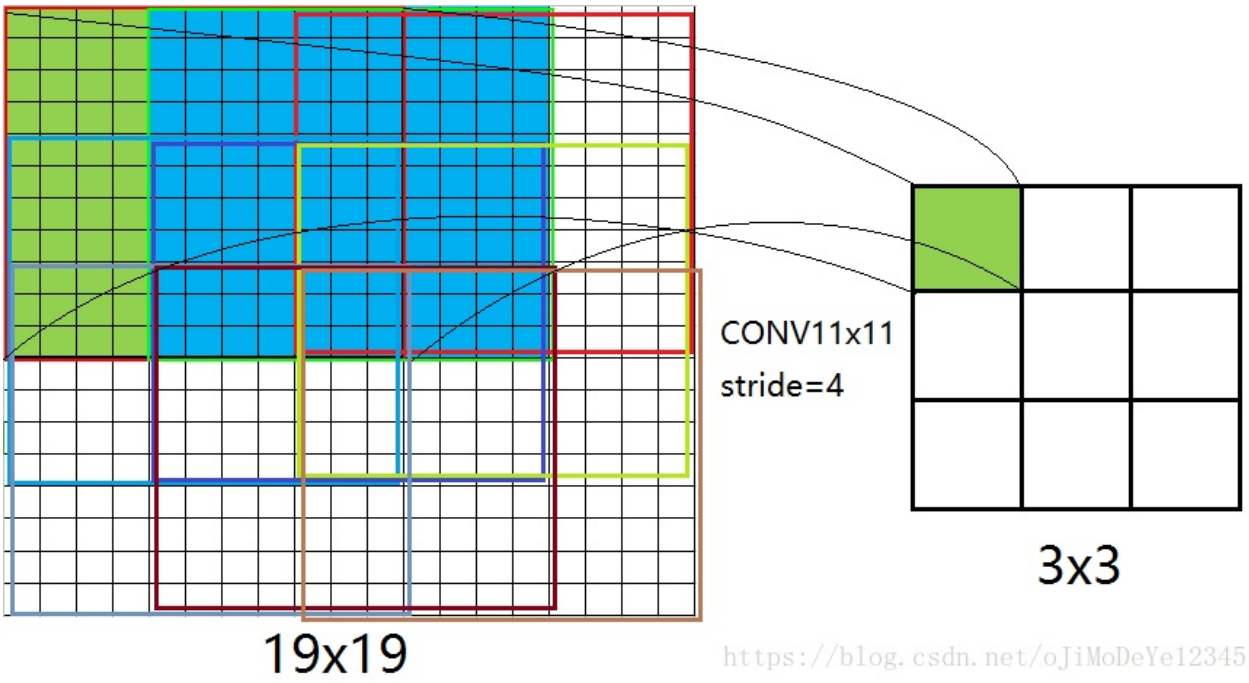
VGG16 的结构如下图：



不像 AlexNet 在第一层卷积使用的卷积核是 11×11+4， ZFNet 与 GooLeNet V1使用的是 7×7+2，VGG 模型从 A-E 都是通过 3×3 的卷积过滤器（ Convolution Filter）增加架构的深度，所有过滤器的 strip 为 1 。之所有使用 3 个 3×3 的卷积核的堆叠来获得 7 ×7 视野（可参考 《CNN：接受视野（Receptive Field）》），是因为这么做有以下好处：

首先：使用三个 ReLU 层来替代一个，使得决策函数更具有判别性（ decision function more discriminative）；
其次：减少了参数，相当于通过3×3的 filter 对7×7的卷积层进行了正则化。

卷积核对于输出的 feature map 的影响，在参考文章《VGGNet 阅读理解 - Very Deep Convolutional Networks for Large-Scale Image Recognition》中有段论述，这里摘录一段：



> conv11x11 这样的大卷积核使用的 stride 为4，见上图是我画的在一张 19×19 的图上做11×11的卷积，其实会发现即使是 stride 为4，对于11×11的kernel size而言，中间有很大的重叠，计算出的3×3区域每个值都会受到周边像素的影响，每个位置卷积的结果会更多考虑周边局部的像素点，原始的特征多少有被平滑掉的感觉。换句话说，局部信息因为过大的重叠，会造成更多细节信息的丢失。

A-LRN 增加了 LRN 层，但在评估的时候可以看到 LRN （ocal Response Normalisation）层并没有起到多大的作用，文章认为 LRN 并没有提升模型在 ILSVRC 数据集上的表现，反而增加了内存消耗和计算时间。

模型 C 和 D 的层数一样，但 C 层使用了 1×1 的卷积核，用于对输入的线性转换，增加非线性决策函数，而不影响卷积层的接受视野。后面的评估阶段也有证明，使用增加的 1×1 卷积核不如添加 3×3 的卷积核。

池化层的核数变小且为偶数，AlexNet 使用的是3×3 stride 为 2，VGG 为2×2 stride 也是 2 。CS231n 课程也提到现在使用 pooling 越来越少了，而是使用 stride 不等于 1 的卷积层来替代。

输入大小为 224×224 RGB 三通道，输入只做了减去 RGB 均值的操作。

训练方法

大部分神经网络的训练都遵循了 AlexNet 的训练方式，除了在输入采样上有所区别。VGG 训练使用了带动量的最小批梯度下降算法（ mini-batch gradient descent with momentum）来优化多项式逻辑回归（ multinomial logistic regression）。参数如下：

- 批次的大小设置为 256，
- 动量设置为 0.9。
- 在前两个全连接层（FC）使用 Dropout，值设置为 0.5。
- 学习速率初始中设置为 1e-2，当验证精度停止提升值，将学习速率衰减10。
- 整个训练过程中学习速率衰减 3 次，在经过 370K 次迭代，即 74 轮。

VGG 训练之所以可以收敛的比 AlexNet 快，是因为：

- a. 通过增加深度和使用小的卷积 filter 隐式的进行了正则化
- b. 预初始化（ pre-initialisation）确定的层

初始化网络的权重很重要，因为在较深的网络中，差的初始化会由于不稳定的梯度而拖延学习。VGG 选择首先选择较浅的网络，如类型 A，他的权重只需要随机初始化即可。然后训练较深的网络，但使用模型 A 的权重来初始化前四层和最后三层 FC的参数，其他中间层使用随机初始化。对于预初始化的层，学习速率不进行减少，而是允许他们在学习期间进行改变（这个啥意思？）。对于随机初始化，通过从均值为 0 方差为 1e-2 的正态分布中采样。偏差 bias 全部初始化为0。

训练图片的尺寸，选取一个固定的最小边 S，然后在 S 上截取大小为 22×224 的区域。S 的选取有两种方式：

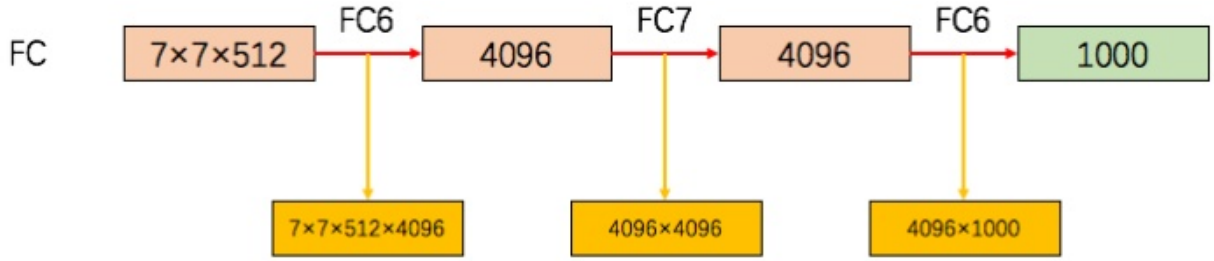
- 第一种：固定 S 的方式，一个是选取一个固定的 S，另一个是选取两个固定的 S，分别为 256（AlexNet、ZFNet有使用） 和 384 。对于一个给定的神经网络配置，首先训练 S=256。为了加速训练 S=384 的网络，会使用预训练的 S=256 网络的权重来初始化参数，然后使用更好的初始化学习速率 1e-3
- 第二种：设置 S 为多尺度，每次训练图片，都通过从一确定的范围[Smin, Smax] （通常值为 Smin=256 、Smax=512）随机采样一个 S ，使用此 S 来缩放图片。因为图片中的物体有不同的尺寸，通过 S 多尺度，这样的情况就被考虑了进去。

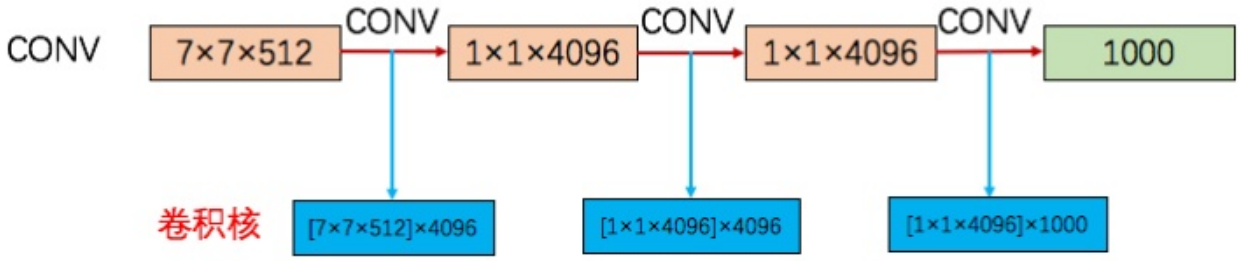
因为速度的原因，论文中训练多尺寸模型时，是通过微调（ fine-tuning）具有相同配置的，固定尺寸 S=384 的预训练模型的所有层。

测试阶段

首先将图片同质化的缩放（ isotropically rescaled）为预定义的最小图片边长，记做 Q。Q 不一定要和训练时的尺寸 S 相等。

作者将三个全连接层在此阶段，转成了1个7×7，和 2 个 1×1 的卷积层。。从图2 VGG16结构图中就可以看到，以第一个全连接层为例，要转卷积层，FC6的输入是 7×7×512，输出是4096（也可以看做 1×1×4096），那么就要对输入在尺寸上（宽高）降维（从7×7 降到 1×1）和深度（channel 或者 depth）升维（从512 升到4096）。把7×7降到1×1，使用大小为 7×7的卷积核就好了，卷积核个数设置为4096，即卷积核为7×7×4096（下图中的[7×7×512]×4096 表示有 4096 个 [7×7×512] 这样的卷积核，7×7×4096 是简写形式忽略了输入的深度），经过对输入卷积就得到了最终的 1×1×4096 大小的 feature map。经过转换的网络就没有了全连接层，这样网络就可以接受任意尺寸的输入，而不是像之前之能输入固定大小的输入。转化如下图：





分类试验

单尺寸评估

设置测试的图片的尺寸为 Q=S， W = 0.5 $*(S_{min} + Smax)$ ，S 的抖动区间为 $[S_{min}, S_{max}]$。

- 论文中提到使用 LRN 的 A 模型的精确度并没有得到提升，所以在 B-E 的模型中就没有使用LRN。
- 误差因为深度的增加而变小，在具有相同深度的 C 和 D 中，使用3×3卷积核的 D 误差小于使用 1×1 卷积核的 C。这说明增加非线性（ non-linearity）是有帮助的，因为 C 比 B 更好。同时使用具有non-trivial 接受视野的卷积核，有利于捕捉空间结构（D 好于 C）。 当模型的深度达到 19 层时，架构的误差率就达到了饱和，即使更深的模型对于大的数据集是有益的。因为 19 层的模型与 16 层的模型误差率基本一致。
- 证明训练时尺寸的抖动（ $S \in [S_{min}, S_{max}]$ ）比固定最小边（S=256 or S=384）有更好的结果。也说明训练时通过尺寸抖动来对训练数据集增强，对于捕捉多尺寸图片统计确实是有帮助的。

多尺寸评估

Table 4: **ConvNet performance at multiple test scales.**

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
B	256	224,256,288	28.2	9.6
C	256	224,256,288	27.7	9.2
	384	352,384,416	27.8	9.2
	[256; 512]	256,384,512	26.3	8.2
D	256	224,256,288	26.6	8.6
	384	352,384,416	26.5	8.6
	[256; 512]	256,384,512	**24.8**	**7.5**
E	256	224,256,288	26.9	8.7
	384	352,384,416	26.7	8.6
	[256; 512]	256,384,512	**24.8**	**7.5**

模型训练时使用固定尺寸的 S，评估时使用 3 个尺寸，这三个尺寸与训练时的尺寸接近 Q={S-32，S，S+32}。同时，在训练时使用多尺寸，那么在测试时可选择的尺寸范围更广。当训练使用 $S \in [S_{min}, S_{max}]$，那么评估时使用的尺寸范围为 $Q = \{S_{min}, 0.5 * (S_{min} + S_{max}), S_{max}\}$。

从上图可以看到，在测试时使用尺寸抖动会有更好的表现（与相同的模型在单尺寸相比）。越深的模型表现越好，尺寸抖动比使用固定最小边 S 训练的模型更好。

多剪裁评估

ConvNet config. (Table 1)	Evaluation method	top-1 val. error (%)	top-5 val. error (%)
D	dense	24.8	7.5
	multi-crop	24.6	7.5
	multi-crop & dense	**24.4**	**7.2**
E	dense	24.8	7.5
	multi-crop	24.6	7.4
	multi-crop & dense	**24.4**	**7.1**

可以看到，使用多裁剪方式表现好于密集评估（ dense evaluation），两者结合起来会更好。

卷积边界条件（convolution boundary conditions）又是怎么回事？可参看 《知乎 - VGG神经网络论文中multi-crop evaluation的结论什么意思？》

神经网络融合

Table 6: **Multiple ConvNet fusion results.**

Combined ConvNet models	Error		
	top-1 val	top-5 val	top-5 test
ILSVRC submission			
(D/256/224,256,288), (D/384/352,384,416), (D/[256;512]/256,384,512) (C/256/224,256,288), (C/384/352,384,416) (E/256/224,256,288), (E/384/352,384,416)	24.7	7.5	7.3
post-submission			
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), dense eval.	24.0	7.1	7.0
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop	23.9	7.2	-
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop & dense eval.	**23.7**	**6.8**	**6.8**

因为模型之间的互补，使得最后的表现有所提升。使用两个表现最好的多尺寸模型，并使用多尺寸评估和密集评估的模型表现最佳。VGGNet 阅读理解 - Very Deep Convolutional Networks for Large-Scale Image Recognition推荐此文章，里面讲解较为详细

问题

怎么增加模型判别性?

论文中提到使 3 个 3×3 的 filter 好于 使用一个 7×7 的 filter，3 个 3×3 的 filter对别提高了模型的判别性（discriminative），那么什么是模型的判别性呢？这个好像和模型的类型有关，模型可以分为生成模型和判别模型。是不是想要了解什么是更具判别性，就需要去了解什么是判别模型。

通过查找判别模型的资料，也没有特别理解，怎么样模型就更具有判别性了。这里引用一些可能对理解判别性有用的资料：

以下是个人的深层思考：网络更深带来了更多变化，更好的特征多样性，就好比是数据增强虽然引来方差是好的，我们想在变化中寻找不变的映射关系。但是，网络更深带来特征更多真的好嘛？我觉得更多的特征和更深的网络，不一定都是有助于、有贡献于正确梯度下降寻找最优或者局部最优的方向，我们真正需要的是可以正确建立映射关系的特征。

反倒是层数越深，特征更多，会有更多局部最优。但为此，我们又在减少因引入特征多样性带来的高方差的影响，不论是在随机梯度下降中引入动量，还是各种正则化的手段，又尝试减少更深网络带来更多特征造成的影响。一方面我们在增大方差，又在减少方差。这样看，似乎这是矛盾的。

网络由于有着本身的更新策略，可以正确建立映射关系。但网络更深却在影响映射关系的建立，把这个建立的过程变得更加曲折，甚至无法建立出好的、正确的、有效的映射关系。

我想了想，感觉这是一个平衡，重要是多了可以筛选：
一方面，我们希望有更多特征，在于我们可以筛选。本身的更新策略可以指导映射关系的建立。但是学习的东西多了，必然会造成学习出现问题，因为要在其中筛选。但是所有的基于部分样本的优化带来的梯度估计，必然会学出的权重都有一点不正确。
另一方面，我们又在减少影响，减少不正确性带来的影响，就有这些正则化来去筛选。有的正则化起到筛选的作用，而有的则是减缓、减小高方差带来错误下降方向的影响的作用。

我觉得重点在于这个特征筛选做的好不好，现在大多情况都不缺数据。其实这样看来，即使是深度学习，又回到了以前的问题，特征工程、特征选择。似乎深度学习带来了更深的网络，表面上看给我们造成没有必要做特征工程的假象，但其实我们做了这个过程，在网络结构设计、模块选择、网络的训练（优化）trick 里。

无论是特征跨 depth 的 cross（resnet），还是跨channel的cross（lrn、shufflenet）。这几年的网络都是在网络更dense的前提（比方mobilenet）下，基于各自的module做特征工程，引入更多特征，一方面特征 diverse 方差加大，一方面又用正则等手段钝化平滑方差。我认为，大家认为深度学习玄学的一个原因可能是不可估量的方差的 tradeoff 造成的。

有小伙伴也提到，玄学也是因为现在的文章中说到的方法其实并不是work的，你按照他说的这么调整就是不对。

附加资料：CSDN - 生成模型与判别模型

什么是密集评估?

文中介绍了 multi-crop 和 dense evaluation 两种评估方法，对其中的dense evaluation 不是特别理解，在网上查了资料也还是没有太明白，好像这个方法和 FCN （Fully Convolutional Networks）有着密切的关系，从参考中的资料中也没有明白两个是怎么密切相关的。

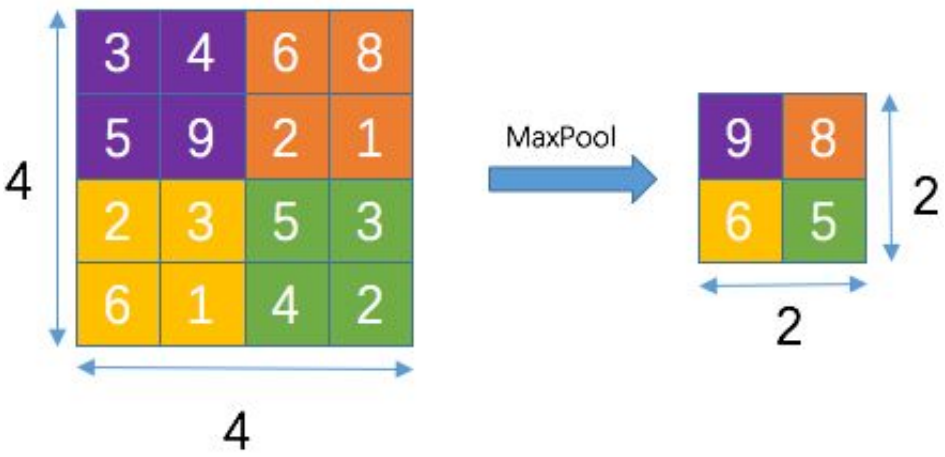
这里也有说下 FCN 和 FC，之前对这两个一直很模糊，总是把两者混在一起。FCN 指的是 Fully Convolutional Networks ，是指一种卷积神经网络，但这个网络中全部都是卷积层。不像传统的卷积神经网络（Convolutional Networks），前面基层是卷积层，最后几层就不是卷积层了，而是全连接层，即 FC（Full Connections）更多的指的是连接的方式。

虽然没有搞懂 dense evaluation 是什么意思，但把参考资料列出来，有懂的看到的话希望能给指点下。

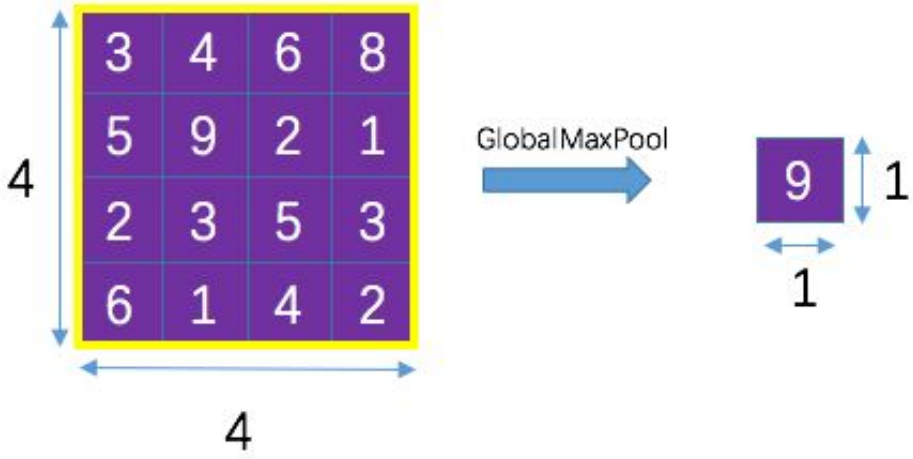
什么是 全局池化（Global Average Pooling）？

论文的附录 B《GENERALISATION OF VERY DEEP FEATURES》提到使用全局平均池化（GAP）方法，那么什么是全局平均池化呢？此概念首先在 NIN（Network In Network） 中提出。

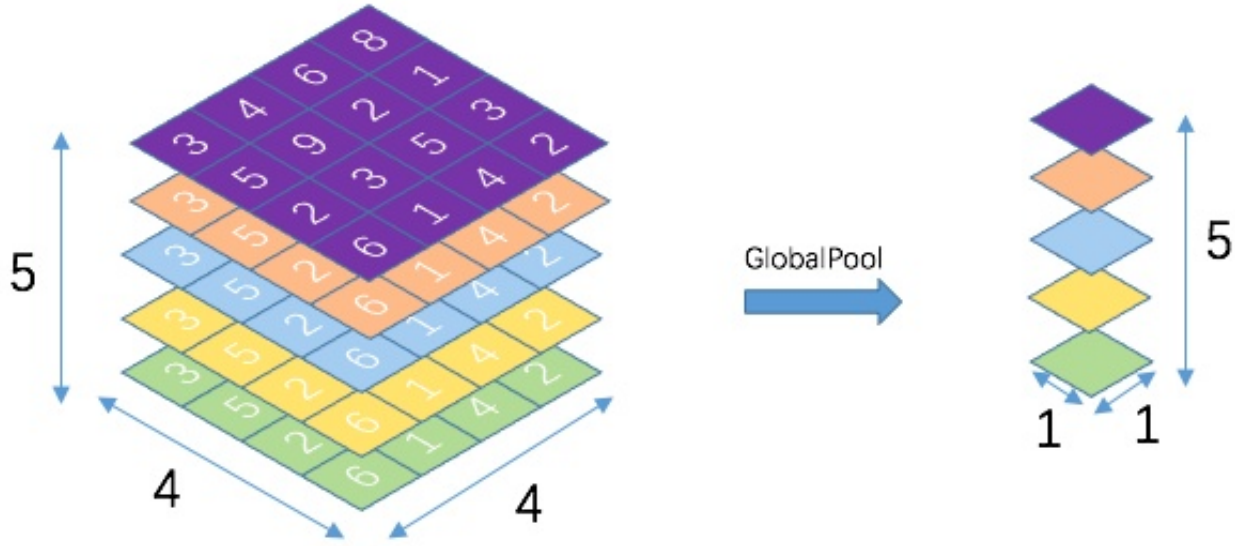
首先，需要知道什么是全局池化（global pooling），它其实指的滑动窗口的大小与整个 feature map 的大小一样，这样一整张feature map 只产生一个值。比如一个 4×4 的 feature map 使用传统的池化方法（2×2 + 2s），那么最终产生的 feature map 大小为 2×2，如下图：



而如果使用全局池化的话（4×4 + 1s，大小与 feature map 相同），一个feature map 只产生一个值，即输出为 1×1，如下图：

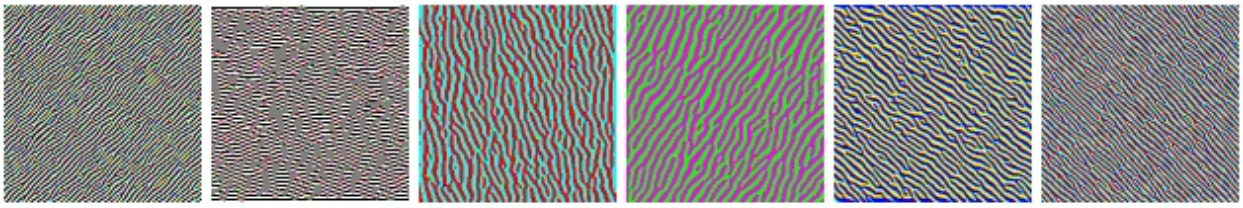


如果前一层有多个feature map 的话，只需要把经过全局池化的结果堆叠起来即可，如下图：

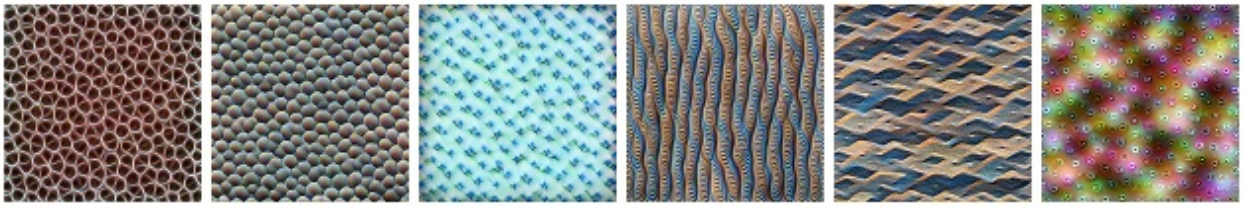


上图，如果使用 Average 池化方法，那么就成为 Global Average Pooling，即 GAP。从而可以总结出，如果输入 feature map 为 W×H×C，那么经过全局池化之后的输出就为 1×1×C。

什么是图像语义?



Edges (layer conv2d0)



Textures (layer mixed3a)

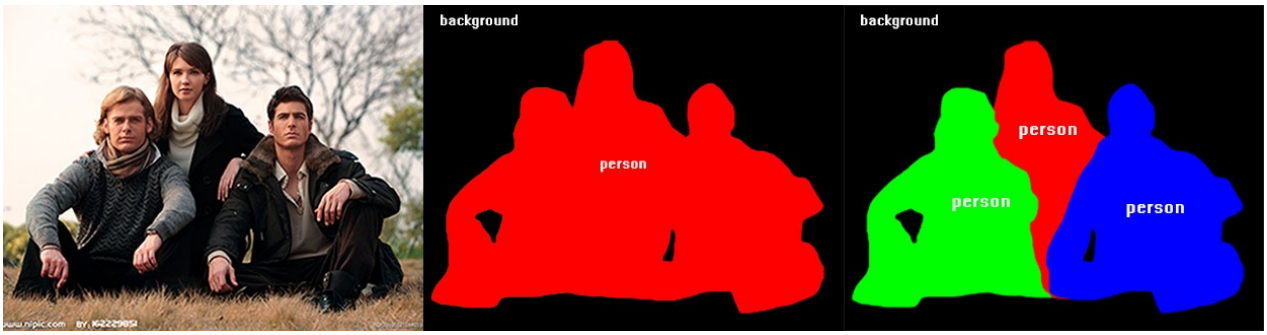
Patterns (layer mixed4a)

Parts (layers mixed4b & mixed4c)

Objects (layers mixed4d & mixed4e)

"浅层学到的是纹理特征，而深层学到的是语义特征" 。从上图可以看到越是低层学到的月粗糙，即学到的都一些边缘（edges）或则纹理（textures），越是高层越偏向于语义特征。那么什么是语义特征呢？语义指的到底是什么呢?

这里的语义主要用于图像分割领域，这里的语义仍主要指分割出来的物体的类别，从分割结果可以清楚的知道分割出来的是什么物体，比如猫、狗等等。即指物体的类别，如猫、狗就是语义。上图，越是高层的就越能展现语义特征。现在还有一种 instance segmentation 方法，可以可以对同一类别的不同物体进行不同的划分，可以清楚地知道分割出来的左边和右边的两个人不是同一个人。如下图：



- `semantic segmentation` - 只标记语义, 也就是说只分割出`人`这个类来
- `instance segmentation` - 标记实例和语义, 不仅要分割出`人`这个类, 而且要分割出`这个人是谁`, 也就是具体的实例

图像中的 L1-normalize 与 L2-normalize

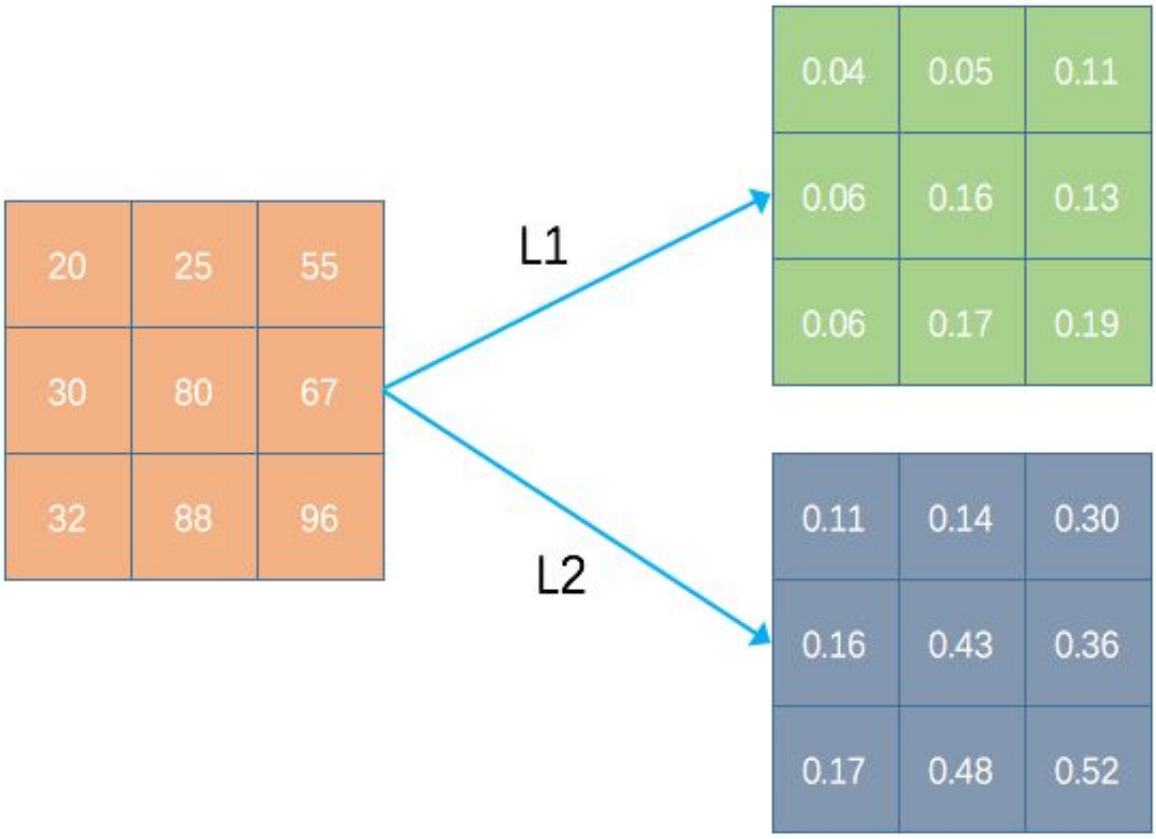
论文的附录部分也提到了图像的 L2-normalize，此 L2 并不是 CNN 中提到的用于解决过拟合的正则化方法，那么图像中的L2-normalize 有指呢?

L1及其 L2的计算公式如下：

$$L1 \rightarrow x_{ij}^{'} = \frac{x_{ij}}{\sum_{i=0}^{H-1}\sum_{i=0}^{W-1} x_{ij}}$$

$$L2 \rightarrow x_{ij}^{'} = \frac{x_{ij}}{\sqrt{\sum_{i=0}^{H-1}\sum_{i=0}^{W-1} x_{ij}^{2}}}$$

其中 $x_{ij}^{'}$ 表示经过 L1或者 L2的值，H 表示图片的高（Height），W 表示宽（Width）， x_{ij} 表示图像第 i行 j 列的像素值。如一个 3×3 的图像，使用 L1与 L2的结果如下图：

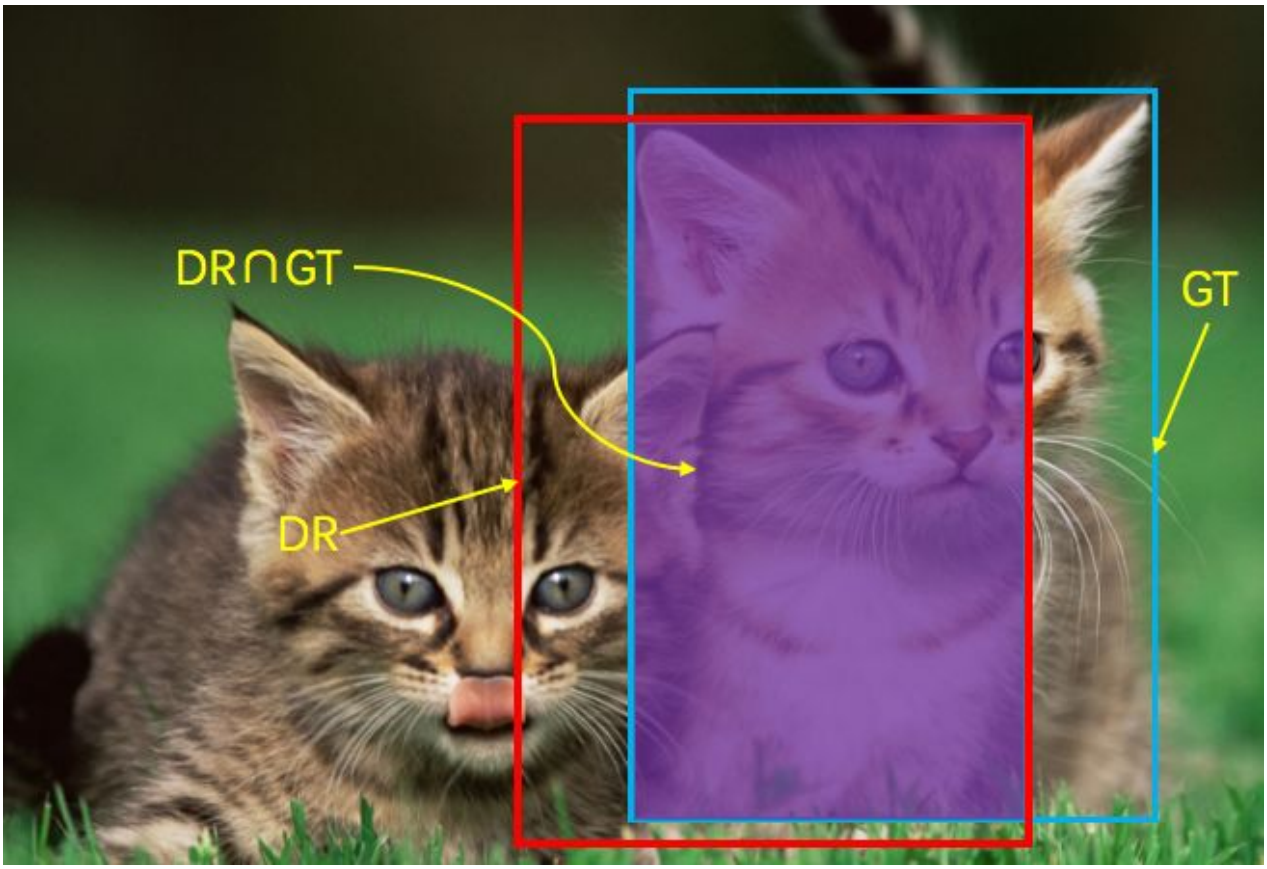


什么是 IoU?

IoU （intersection-over-union）是用于评价目标检测（Object Detection）的评价函数，模型简单来讲就是模型产生的目标窗口和原来标记窗口的交叠率。即检测结果(DetectionResult)与 Ground Truth 的交集比上它们的并集，即为检测的准确率 IoU :

$$IoU = \frac{DR \cap GT}{DR \cup GT}$$

其中DR=Detection Result ， GT = Ground Truth。



或者写成如下的公式：

$$IoU = \frac{\text{交集}}{\text{并集}}$$

可以看到 IoU 的值越大，表明模型的准确度越好，IoU = 1 的时候 DR 与 GT 重合。