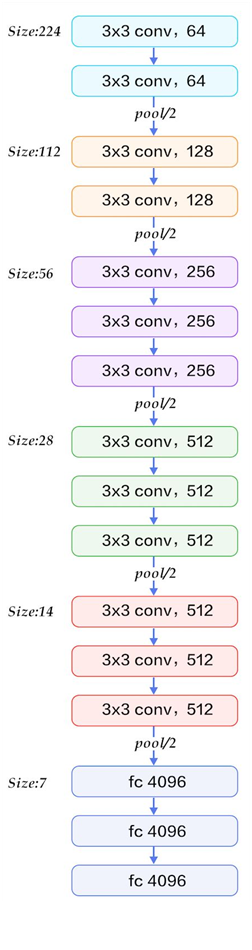
**深度学习知识点总结**

* 梯度方向代表函数值增大的方向，梯度的模长代表函数增大的速率。
* 更新权值信息：是根据梯度信息对Loss值影响进行确定的，当梯度为负值时，则权重方向和Loss值成负相关。
* Sigmoid函数可以将函数值压缩到[0,1],但是在求导导数时，出现梯度弥散现象，长时间权重得不到更新。
* 学习速率：在训练时，前期学习率设置设置较大，随着训练次数的增加，逐渐降低，利用动态学习率。

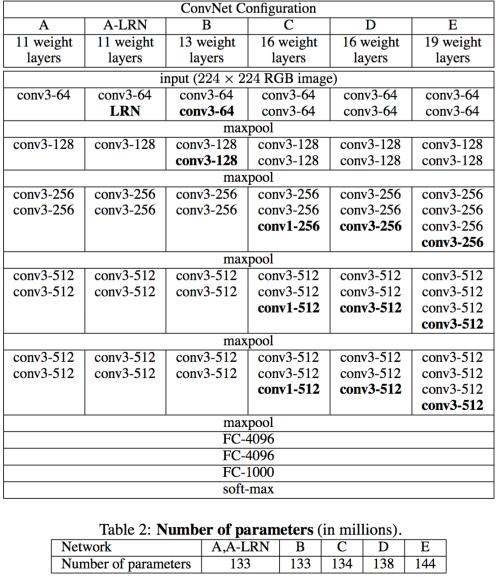
**VGG-Net介绍**

VGG Net由牛津大学的视觉几何组（Visual Geometry Group）和 Google DeepMind公司的研究员一起研发的的深度卷积神经网络。



VGG16-NET结构图

* 输入：224\*224大小的RGB图像；
* 预处理：计算出三个通道的平均值，在每个像素上减去平均值，目的是加快收敛速度；
* 在卷积过程中，采用的卷积和大小为3\*3或者1\*1；
* 5个池化层；
* 3全连接层，前两个全连接层的大小是4096，最后一个是1000；



* 选择采用3\*3的卷积核是因为3\*3是最小的能够捕捉像素8邻域信息的的尺寸
* 使用1\*1的卷积核目的是在不影响输入输出的维度情况下，对输入进行形变，再通过ReLU进行非线性处理，提高决策函数的非线性。
* 2个3\*3卷积堆叠等于1个5\*5卷积，3个3\*3堆叠等于1个7\*7卷积，感受野大小不变，而采用更多层、更小的卷积核可以引入更多非线性（更多的隐藏层，从而带来更多非线性函数），提高决策函数判决力，并且带来更少参数。
* 每个VGG网络都有3个FC层，5个池化层，1个softmax层。
* 在FC层中间采用dropout层，防止过拟合。



**总结：**

**1、通过增加深度能有效地提升性能；**

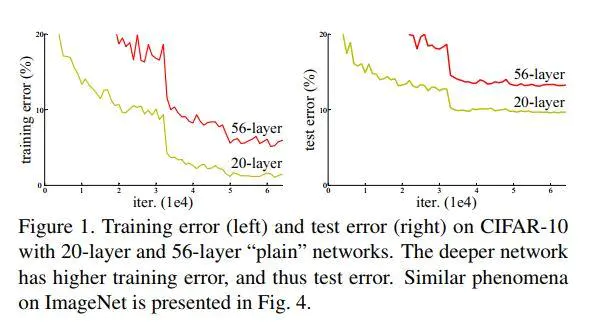
**2、最佳模型：VGG16，从头到尾只有3x3卷积与2x2池化，简洁优美；**

**3、卷积可代替全连接，可适应各种尺寸的图片；**

**4、多个小卷积核比单个大卷积核性能好；**

**ResNet介绍**

深度学习愈是深（复杂，参数多）愈是有着更强的表达能力，可后来发现深度CNN网络达到一定深度后再一味地增加层数并不能带来进一步地分类性能提高，反而会招致网络收敛变得更慢，test dataset的分类准确率也变得更差。



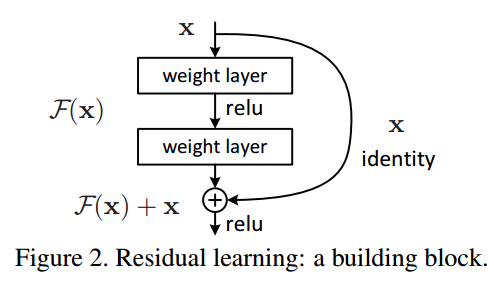
随着网络层数的增加，网络发生了退化（degradation）的现象：随着网络层数的增多，训练集loss逐渐下降，然后趋于饱和，当你再增加网络深度的话，训练集loss反而会增大。注意这并不是过拟合，因为在过拟合中训练loss是一直减小的。当网络退化时，浅层网络能够达到比深层网络更好的训练效果，这时如果我们把低层的特征传到高层，那么效果应该至少不比浅层的网络效果差，从信息论的角度讲，由于DPI（数据处理不等式）的存在，在前向传输的过程中，随着层数的加深，Feature Map包含的图像信息会逐层减少，而ResNet的直接映射的加入，保证了L+1层的网络一定比L层包含更多的图像信息。

* **残差块：Residual Block的设计**

残差网络是由一系列残差块组成的（图1）。一个残差块可以用表示为：



残差块分成两部分直接映射部分和残差部分。



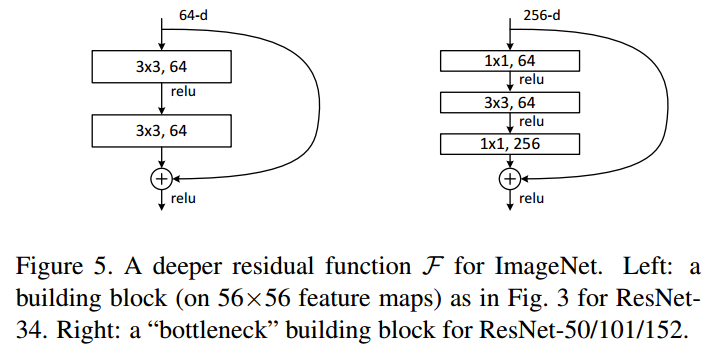
一个残差块有2条路径F(x)和x，F(x)路径拟合残差，不妨称之为残差路径，x路径为identity mapping恒等映射，称之为”shortcut”。图中的⊕为element-wise addition，要求参与运算的F(x)和x的尺寸要相同。所以，随之而来的问题是：

* **残差路径如何设计？**
* **shortcut路径如何设计？**
* **Residual Block之间怎么连接？**

**要求参与运算的F(x)和x的尺寸要相同**

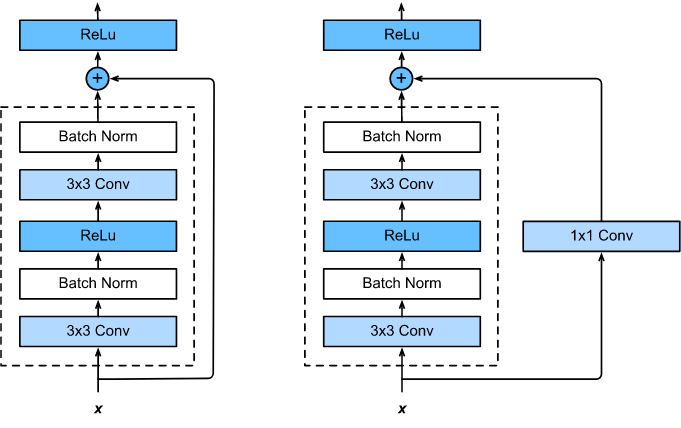
**残差路径可以大致分成2种：**

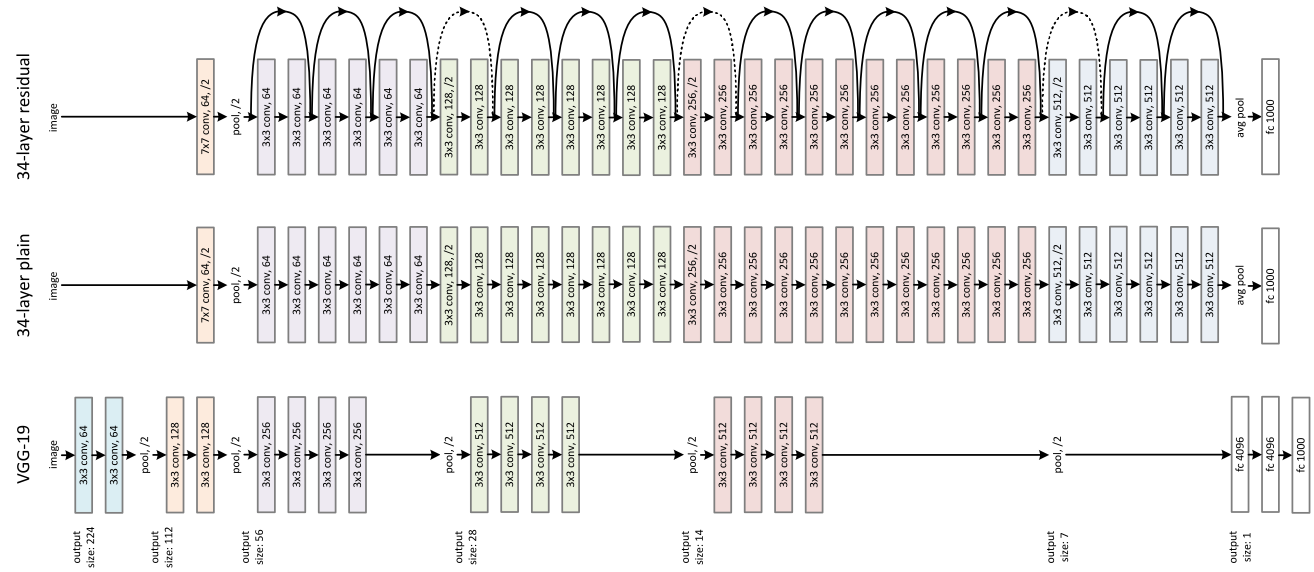
**bottleneck block和basic block**

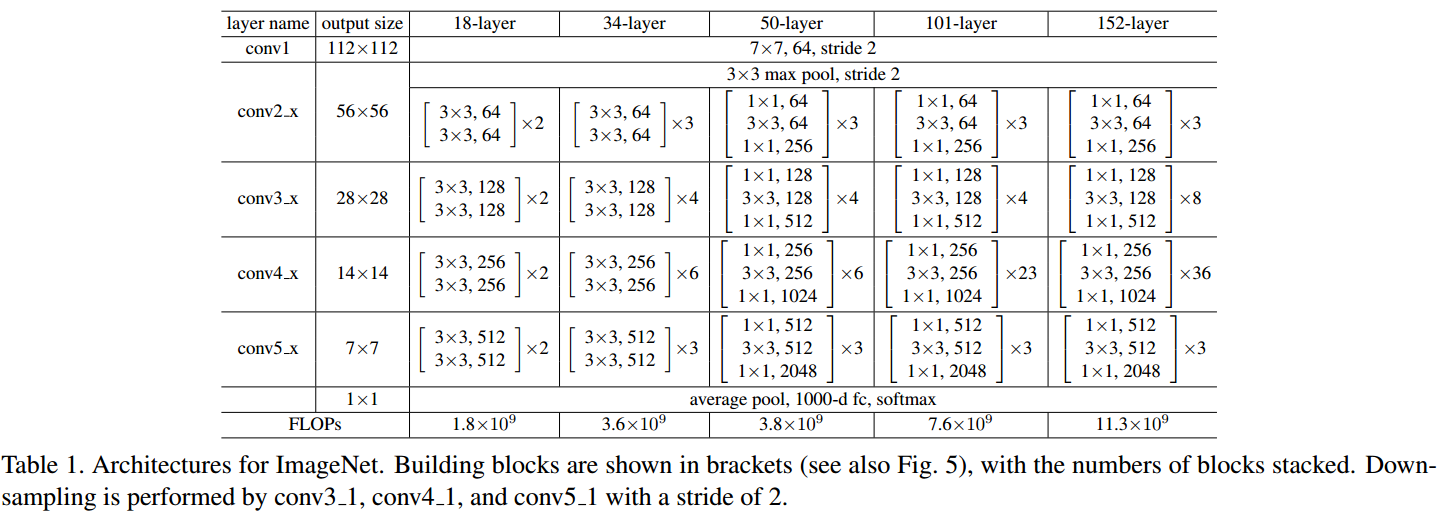


**basic block bottleneck block**

shortcut路径大致也可以分成2种，取决于残差路径是否改变了feature map数量和尺寸，一种是将输入x原封不动地输出，另一种则需要经过1×1卷积来升维 or/and 降采样，主要作用是将输出与F(x)路径的输出保持shape一致，对网络性能的提升并不明显，两种结构如下图所示：

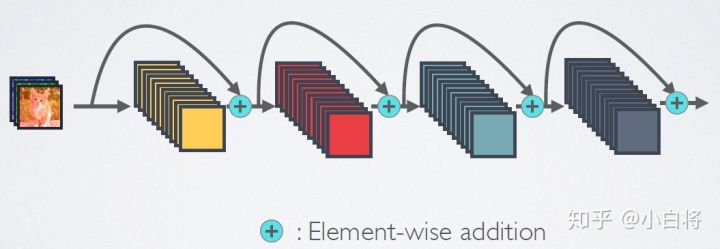


****



**ResNet的设计有如下特点：**

* ResNet中，所有的Residual Block都没有pooling层，降采样是通过conv的stride实现的；
* 分别在conv3\_1、conv4\_1和conv5\_1 Residual Block，降采样1倍，同时feature map数量增加1倍，如图中虚线划定的block；
* 通过Average Pooling得到最终的特征，而不是通过全连接层；
* 每个卷积层之后都紧接着BatchNorm layer



ResNet连接机制

**总结：**

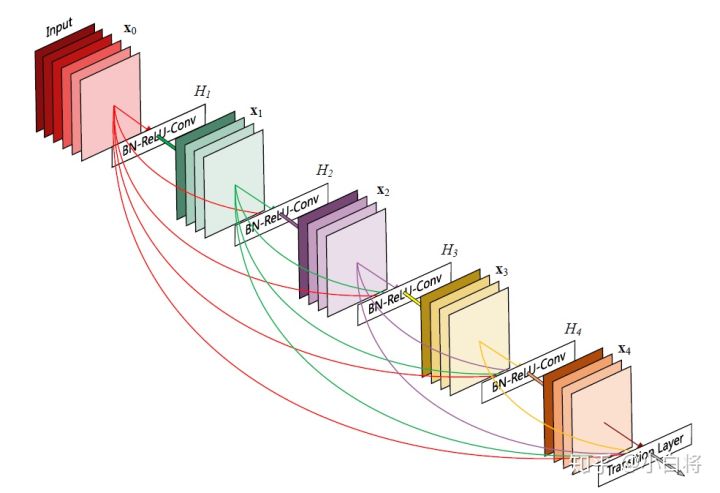
**ResNet结构非常容易修改和扩展，通过调整block内的channel数量以及堆叠的block数量，就可以很容易地调整网络的宽度和深度，来得到不同表达能力的网络，而不用过多地担心网络的“退化”问题，只要训练数据足够，逐步加深网络，就可以获得更好的性能表现。**

**DenseNet介绍**

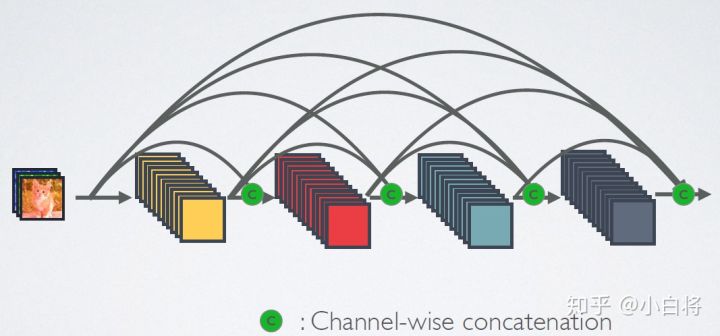
**DenseNet特点：**

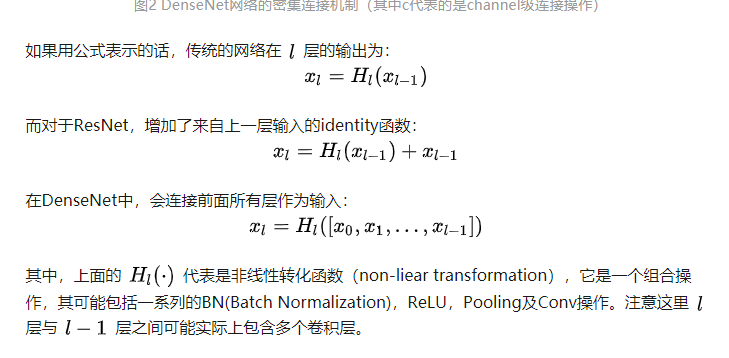
* 建立的是前面所有层与后面层的密集连接；
* 特征在channel上的连接来实现特征重用；
* DenseNet在参数和计算成本更少的情形下实现比ResNet更优的性能。

**网络结构：**

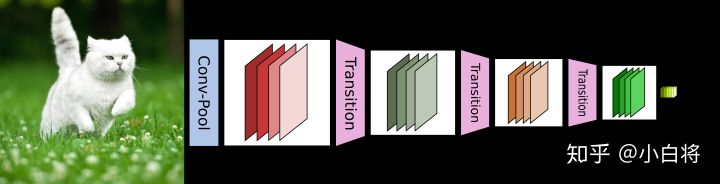


**DenseNet连接机制：**





CNN网络一般要经过Pooling或者stride>1的Conv来降低特征图的大小，而DenseNet的密集连接方式需要特征图大小保持一致。为了解决这个问题，DenseNet网络中使用DenseBlock + Transition的结构，其中DenseBlock是包含很多层的模块，每个层的特征图大小相同，层与层之间采用密集连接方式。而Transition模块是连接两个相邻的DenseBlock，并且通过Pooling使特征图大小降低。图4给出了DenseNet的网路结构，它共包含4个DenseBlock，各个DenseBlock之间通过Transition连接在一起。



**网络结构**

如前所示，DenseNet的网络结构主要由DenseBlock和Transition组成，如图5所示。下面具体介绍网络的具体实现细节。

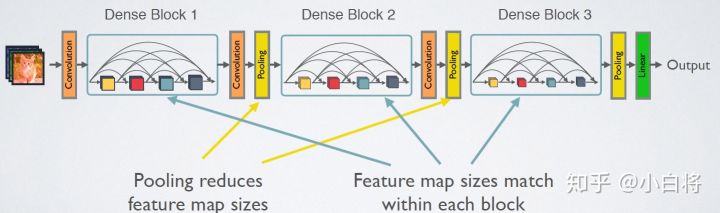
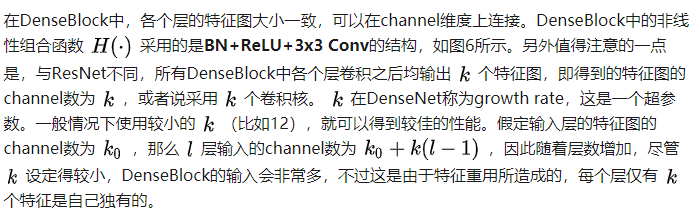
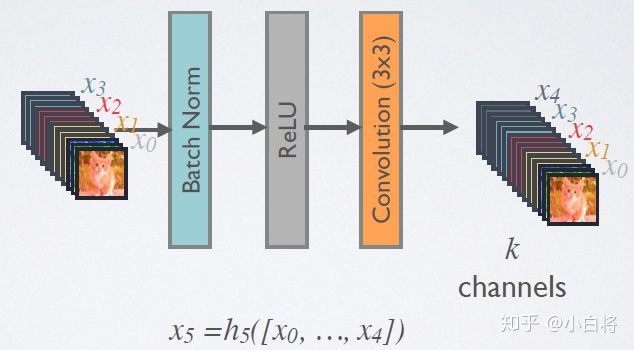


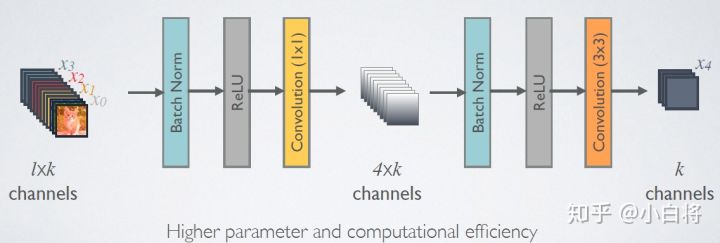
图5



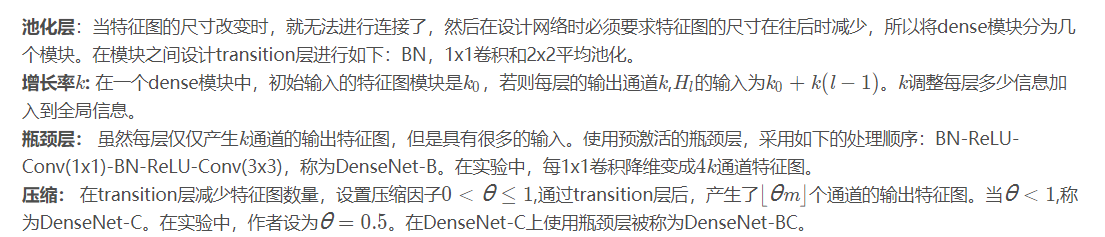


**图6**

由于后面层的输入会非常大，DenseBlock内部可以采用bottleneck层来减少计算量，主要是原有的结构中增加1x1 Conv，如图7所示，即BN+ReLU+1x1 Conv+BN+ReLU+3x3 Conv，称为DenseNet-B结构。其中1x1 Conv得到4\*K个特征图它起到的作用是降低特征数量，从而提升计算效率。



对于Transition层，它主要是连接两个相邻的DenseBlock，并且降低特征图大小。Transition层包括一个1x1的卷积和2x2的AvgPooling，结构为BN+ReLU+1x1 Conv+2x2 AvgPooling。另外，Transition层可以起到压缩模型的作用。

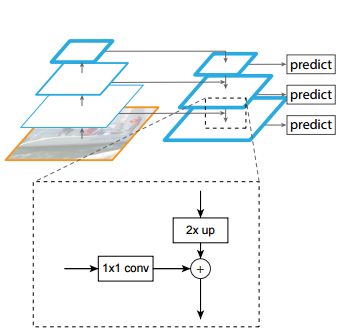


**FPN介绍**

**网页介绍：**

<https://www.cnblogs.com/wangyong/p/8535044.html>

FPN网络，网络结构原理如下图所示:



**FPN能解决什么问题**

* FPN主要解决的是物体检测中的多尺度问题，通过简单的网络连接改变，在基本不增加原有模型计算量的情况下，大幅度提升了小物体检测的性能左边的称为“自底向上”，右边的称为“自上而下”

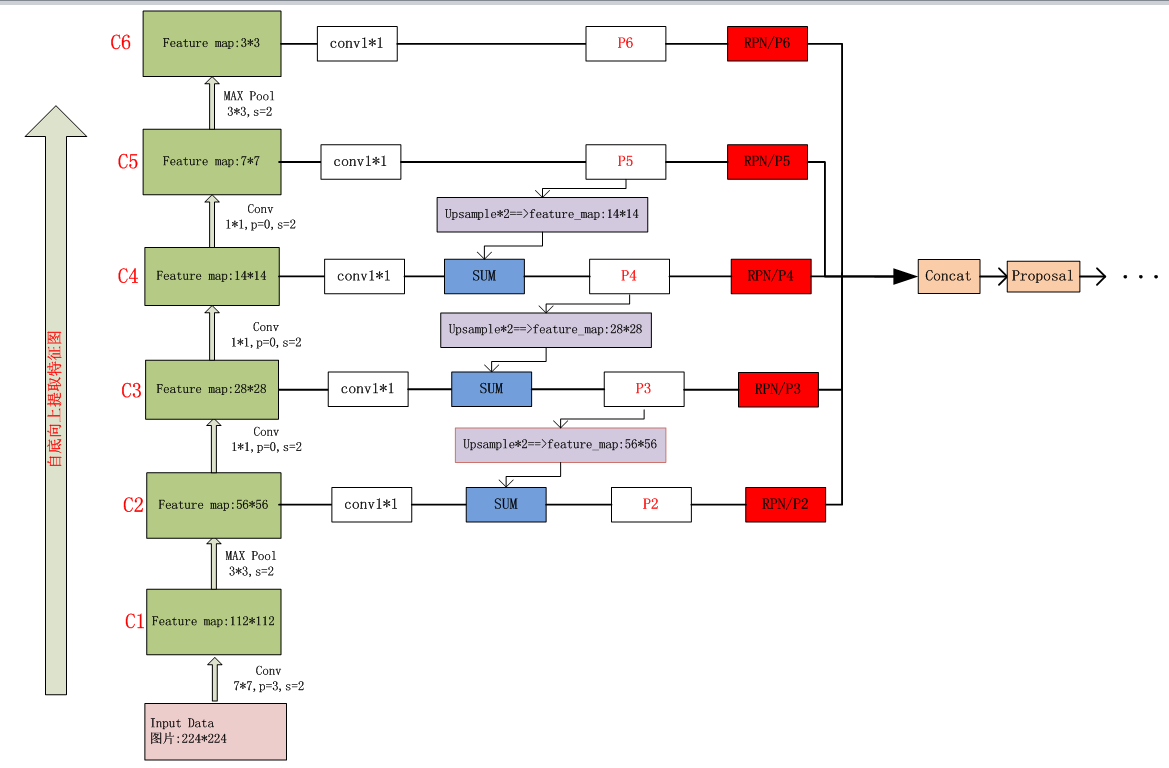
**①自底向上：**

自底向上的过程就是神经网络普通的正向传播过程，特征图经过卷积核计算，通常会越变越小

**②自上而下：**

自上而下的过程是把更抽象、语义更强的高层特征图进行上采样，然后把该特征横向连接至前一层特征，因此，高层特征得到了增强，每一层预测所用的feature map都融合了不同分辨率、不同语义强度的特征，可以完成对应分辨率大小的物体进行检测，保证每一层都有合适的分辨率以及强语义特征。

* 值得注意的是：横向连接的两层特征在空间尺寸上要相同，这样做可以利用底层定位细节信息
* 上面通过文字描述完成了FPN网络的演化历程及其基本原理，下面通过一张图，对FPN的工作原理进行补充描述（基于ResNet50网络）



**YOLOV1**