视觉分类任务

1.1：分类任务简介

CV江湖中
天下武功出**卷积！**

→ 分类任务是必争之地

啥是分类？

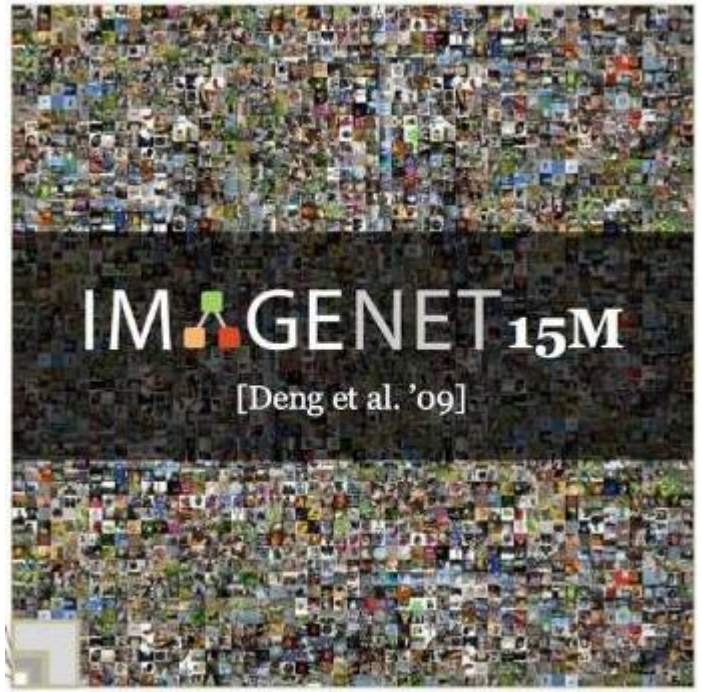
→ 输入：
单一主体图像
输出：label

例如

→

Label：车

想要模型效果好
数据集很重要



1500万张图片
2.2万个类别

←

ImageNet Challenge

IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.

120万张训练图片
1000个类别

1.2：人类分类水平

这个挑战赛难度有多大？

Top1-error: 30%
Top5-error: 5%

人类水平

有点不服？

啥？人类一次就猜对的概率只有70%？

不服的同学对请对右侧图片分类


燕雀


金翅雀


家朱雀


灯芯草雀


蓝鸦


夜莺


松鸦


喜鹊


山雀

给出标签

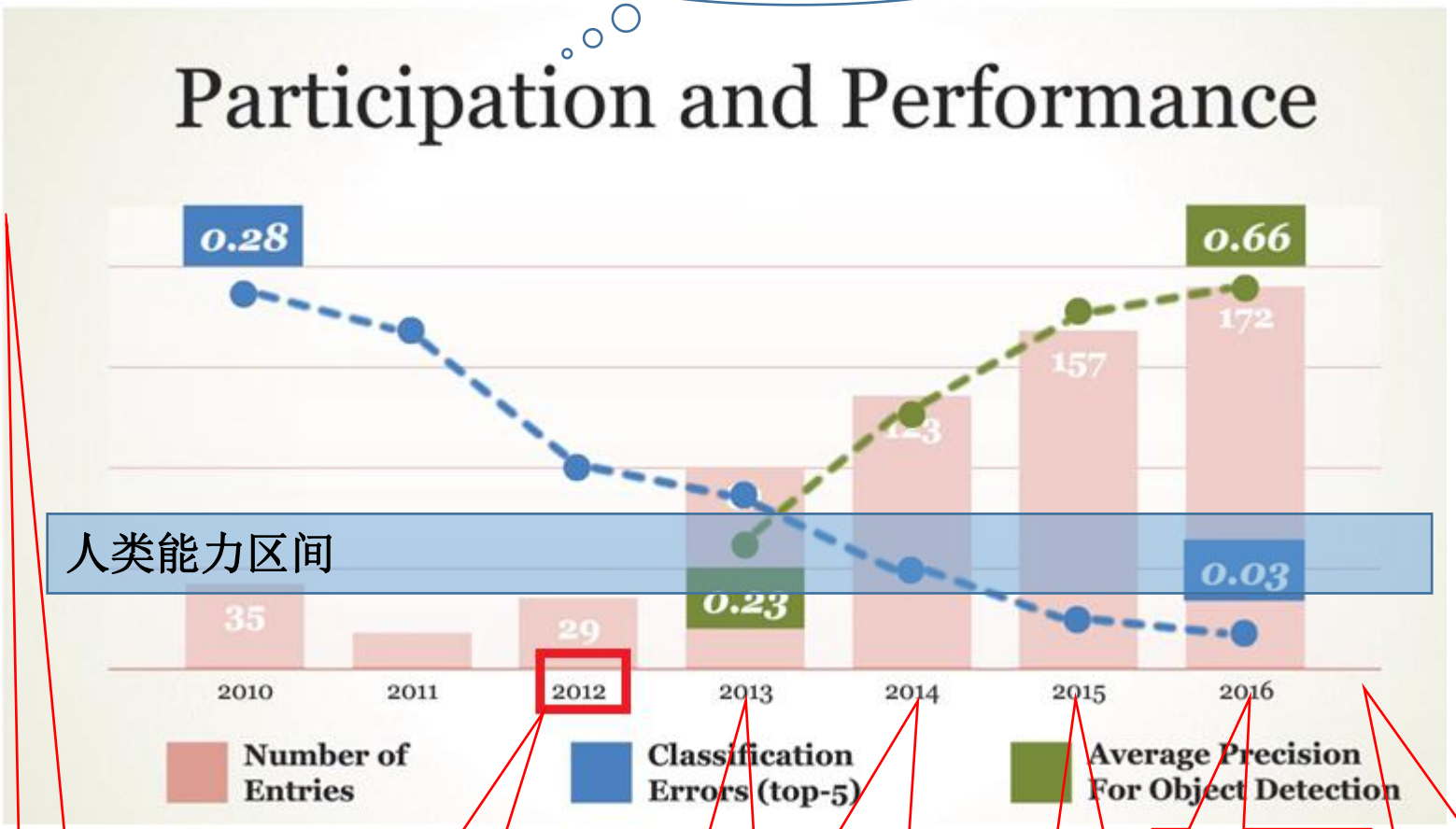
降低难度

正确答案

1.燕雀
2.金翅雀
3.家朱雀
4.灯芯草雀
5.蓝鸦
6.夜莺
7.松鸦
8.喜鹊
9.山雀

1.3：分类算法发展史

别怕，所有这些网络都只在做一件事：
用不同的“姿势”做卷积。来提升分类精度



ImageNet的分类结果（加粗为冠军）

年	网络/队名	val top-1	val top-5	test top-5	备注
2012	AlexNet	38.1%	16.4%	16.42%	5 CNNs
2012	**AlexNet**	36.7%	15.4%	15.32%	7CNNs。用了2011年的数据
2013	OverFeat			14.18%	7 fast models
2013	OverFeat			13.6%	赛后。7 big models
2013	ZFNet			13.51%	ZFNet论文上的结果是14.8
2013	Clarifai			11.74%	
2013	**Clarifai**			11.20%	用了2011年的数据
2014	VGG			7.32%	7 nets, dense eval
2014	VGG（亚军）	23.7%	6.8%	6.8%	赛后。2 nets
2014	**GoogleNet v1**			6.67%	7 nets, 144 crops
	GoogleNet v2	20.1%	4.9%	4.82%	赛后。6 nets, 144 crops
	GoogleNet v3	17.2%	3.58%		赛后。4 nets, 144 crops
	GoogleNet v4	16.5%	3.1%	3.08%	赛后。v4+Inception-Res-v2
2015	**ResNet**			3.57%	6 models
2016	**Trimps-Soushen**			2.99%	公安三所
2016	ResNeXt（亚军）			3.03%	加州大学圣地亚哥分校
2017	**SENet**			2.25%	Momenta 与牛津大学

1998年 LeNet5

AlexNet

ZFNet

GoogleNet V1&VGG

ResNet

ResNeXt

2017冠军：SENet　0.0225

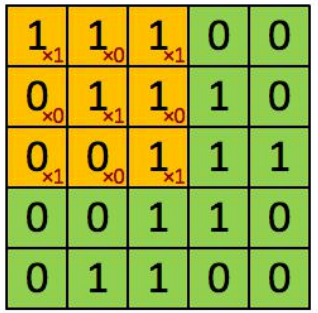
2.1：知识回顾–卷积运算

卷积是一种运算

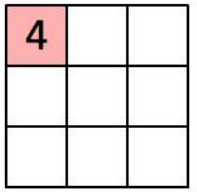
运算的三要素

1、发生关系的两个**变量**是啥
2、运算**规则**是啥
3、运算**结果**是啥

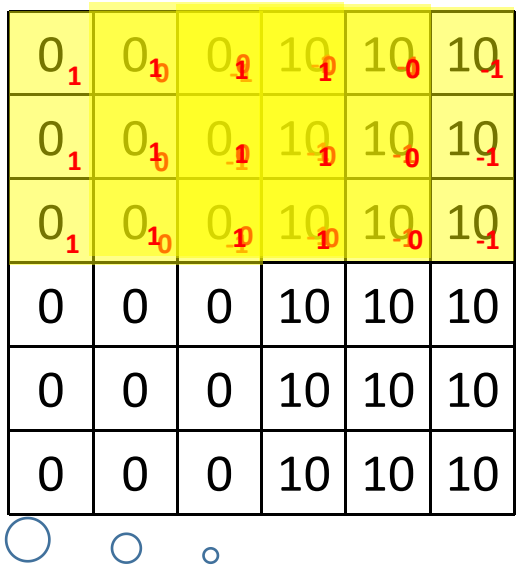
动图展示



Image



Convolved Feature

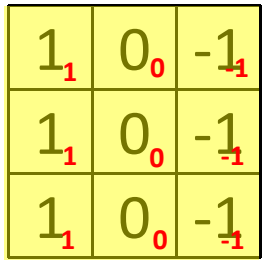


输入图像是特殊的feature map

1.变量：输入图像/FM

2.运算规则：
扫一遍，依次相乘再相加

*



1.变量：卷积核

=

0	-30	-30	0
0	-30	-30	0
0	-30	-30	0
0	-30	-30	0

3.运算结果：新的输出FM

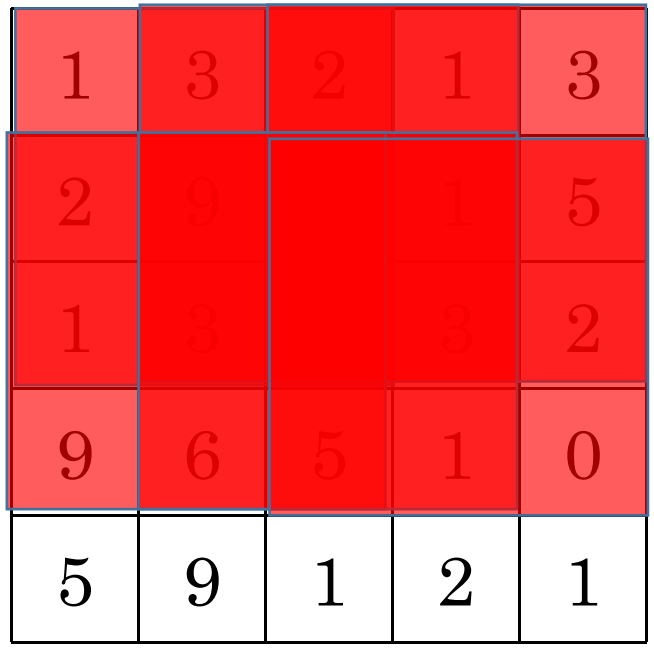
2.2：知识回顾-池化运算

需要选取的超参：
1、卷积/池化核尺寸f
2、卷积/池化核步长s
3、是否需要padding

S越大，运算后得到的FM越小，假设卷积核尺寸为f*f，
输入图像尺寸为n*n，则输出FM的尺寸为：

向下取整符号

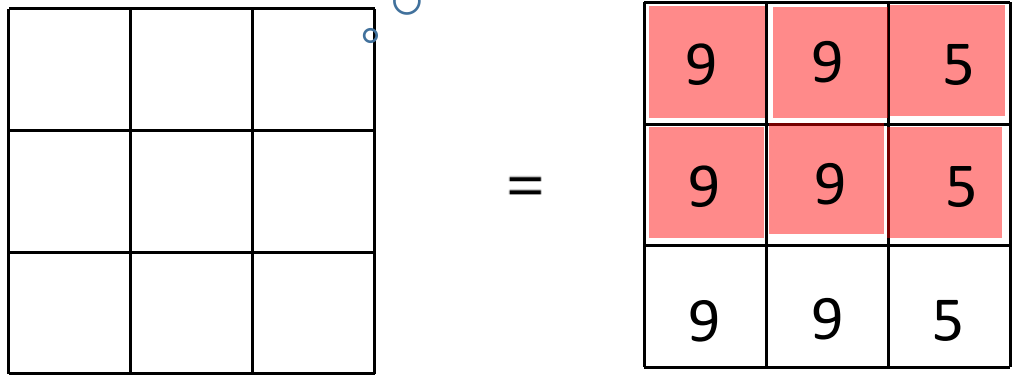
$$\left[\frac{n-f}{s}+1\right] * \left[\frac{n-f}{s}+1\right]$$



2.运算规则：
扫一遍挑出最大的

Max pooling

没有参数需要更新
很满意吧？



1.变量：输入FM

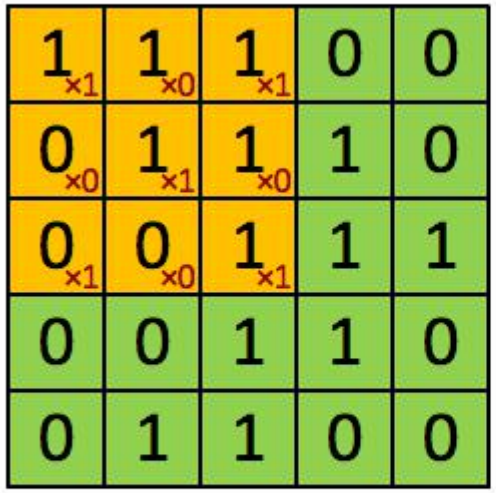
1.变量：池化核

3.运算结果：新的FM

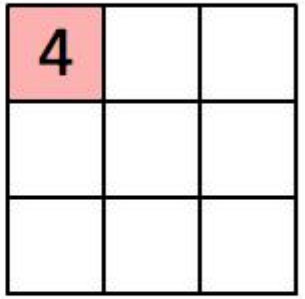
2.3：知识回顾-padding

padding

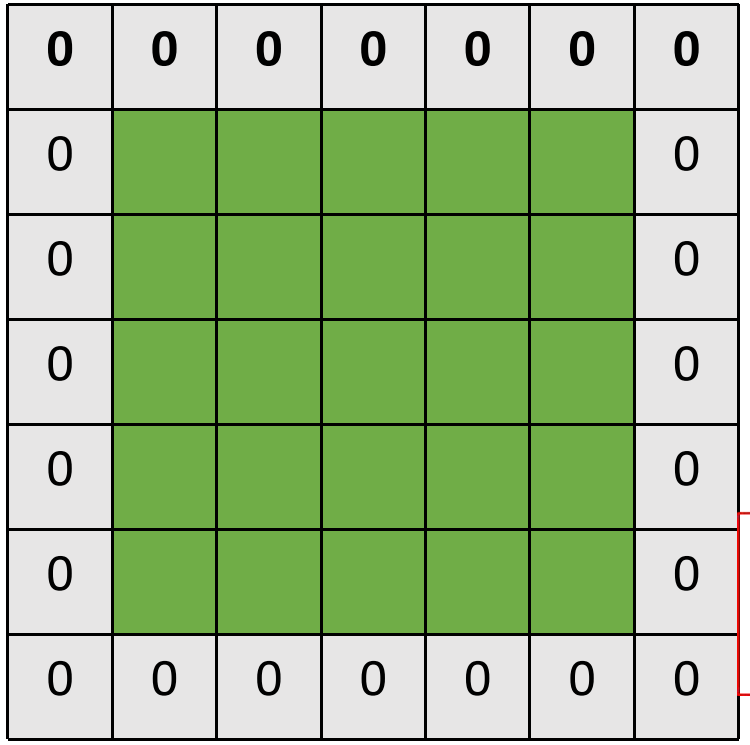
角落里的元素只被扫到一次不太公平？
越靠近边界，被扫到（特征表示）的几率越小



Image



Convolved Feature



由于SAME padding时，$p=\frac{f-1}{2}$，所以f必为奇数

填补的策略有2种：
SAME：保持FM不缩小
VALID ： p=0。

加入padding后，输出FM尺寸变为

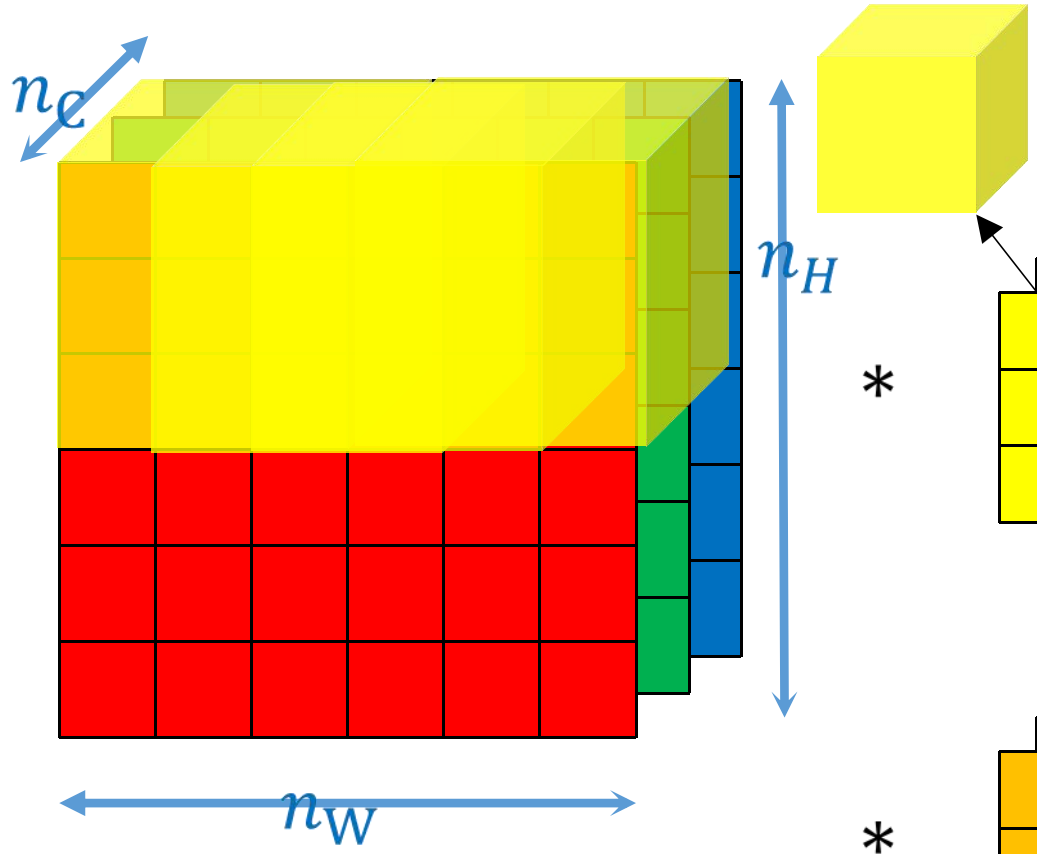
$$\left[\frac{n+2p-f}{s}+1\right] * \left[\frac{n+2p-f}{s}+1\right]$$

在边界处填补（padding）一些像素块
边界向外拓展的像素个数用p表示，此处p=1

2.4：知识回顾-三维卷积(池化)

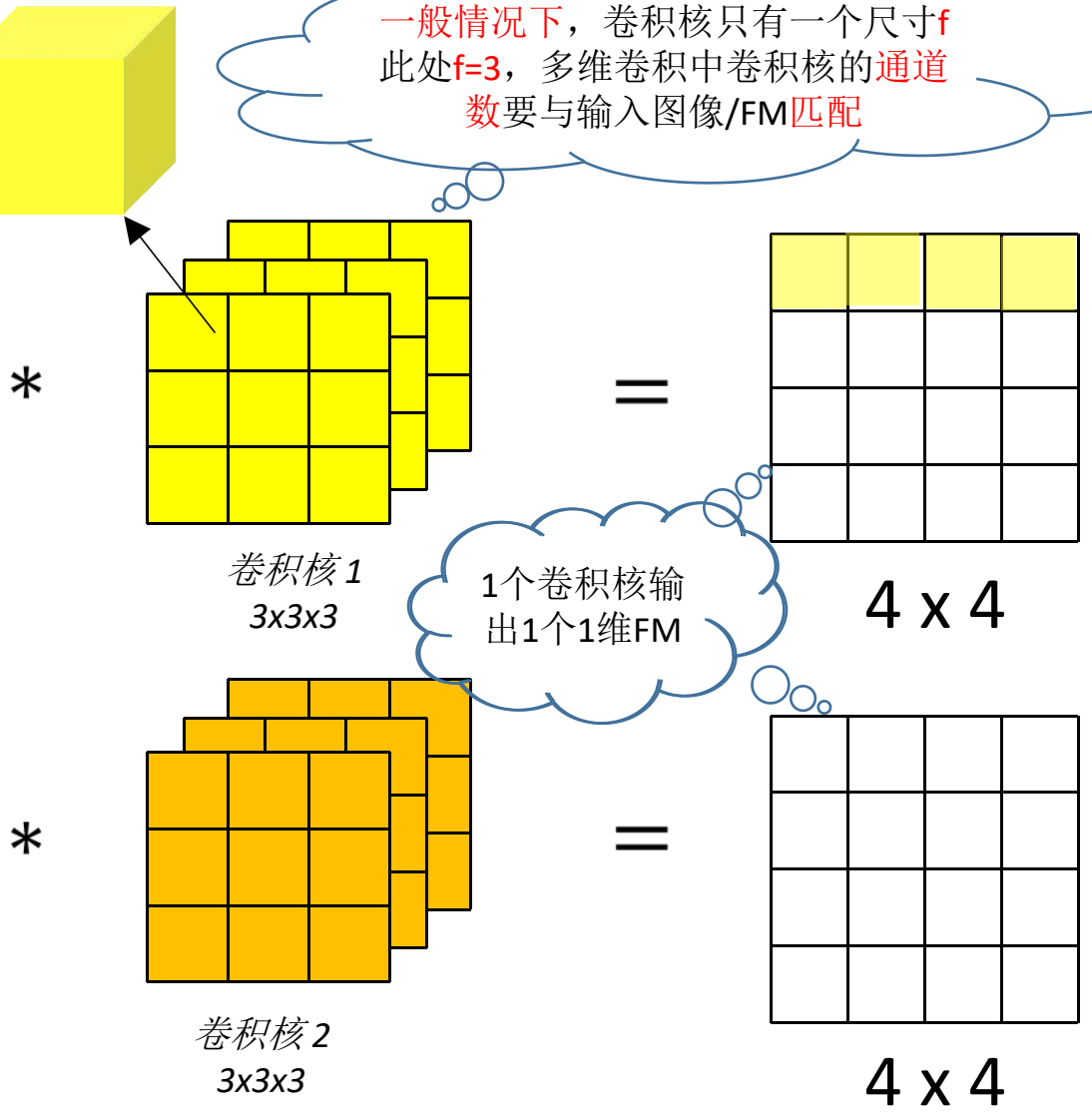
多维卷积&尺寸标记

图像是彩色的
有了三个颜色通道

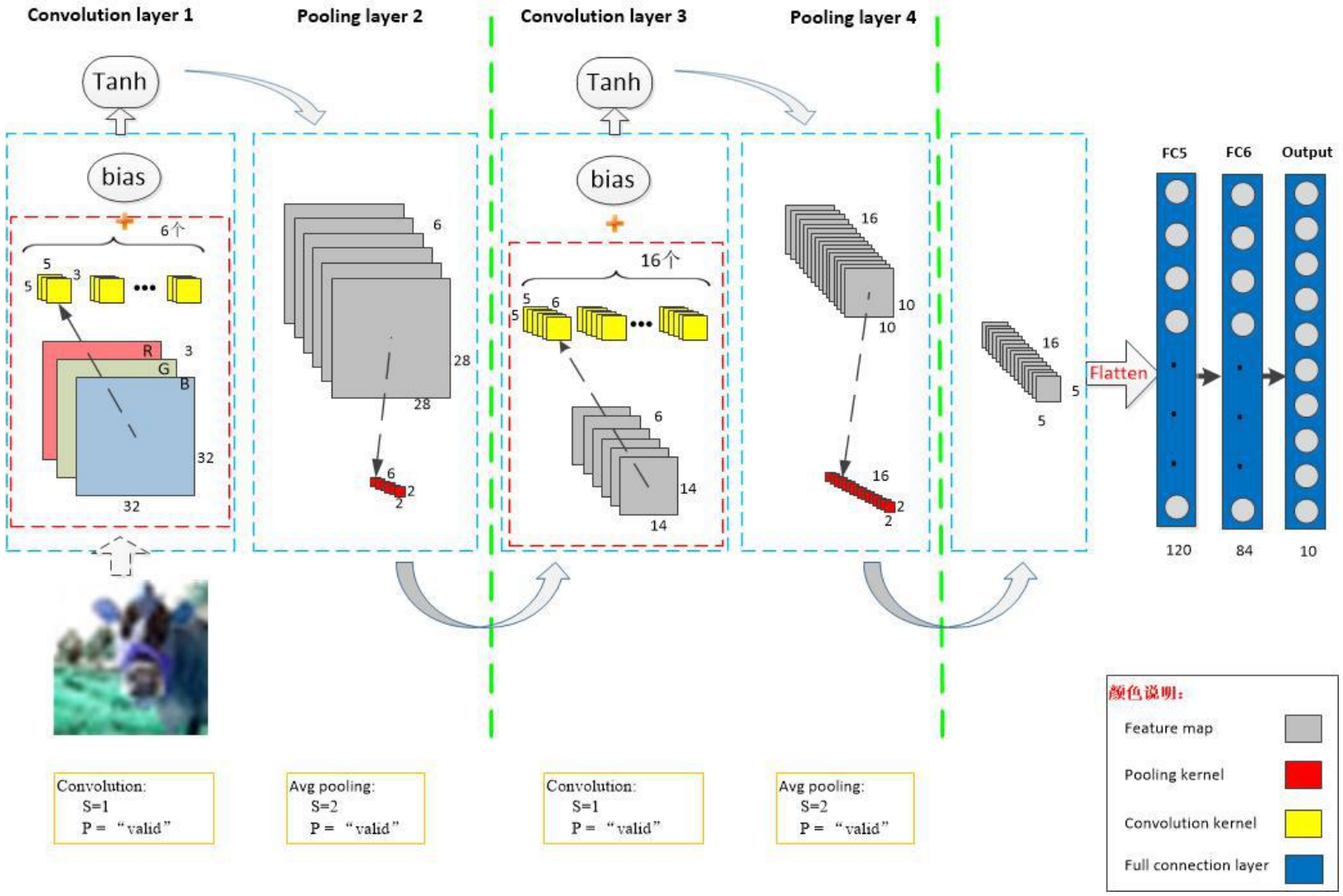


$n_H \times n_W \times n_C = 6 \times 6 \times 3$

一般情况下，卷积核只有一个尺寸f
此处f=3，多维卷积中卷积核的通道数要与输入图像/FM匹配



2.5：LeNet5 --- 一切的原点



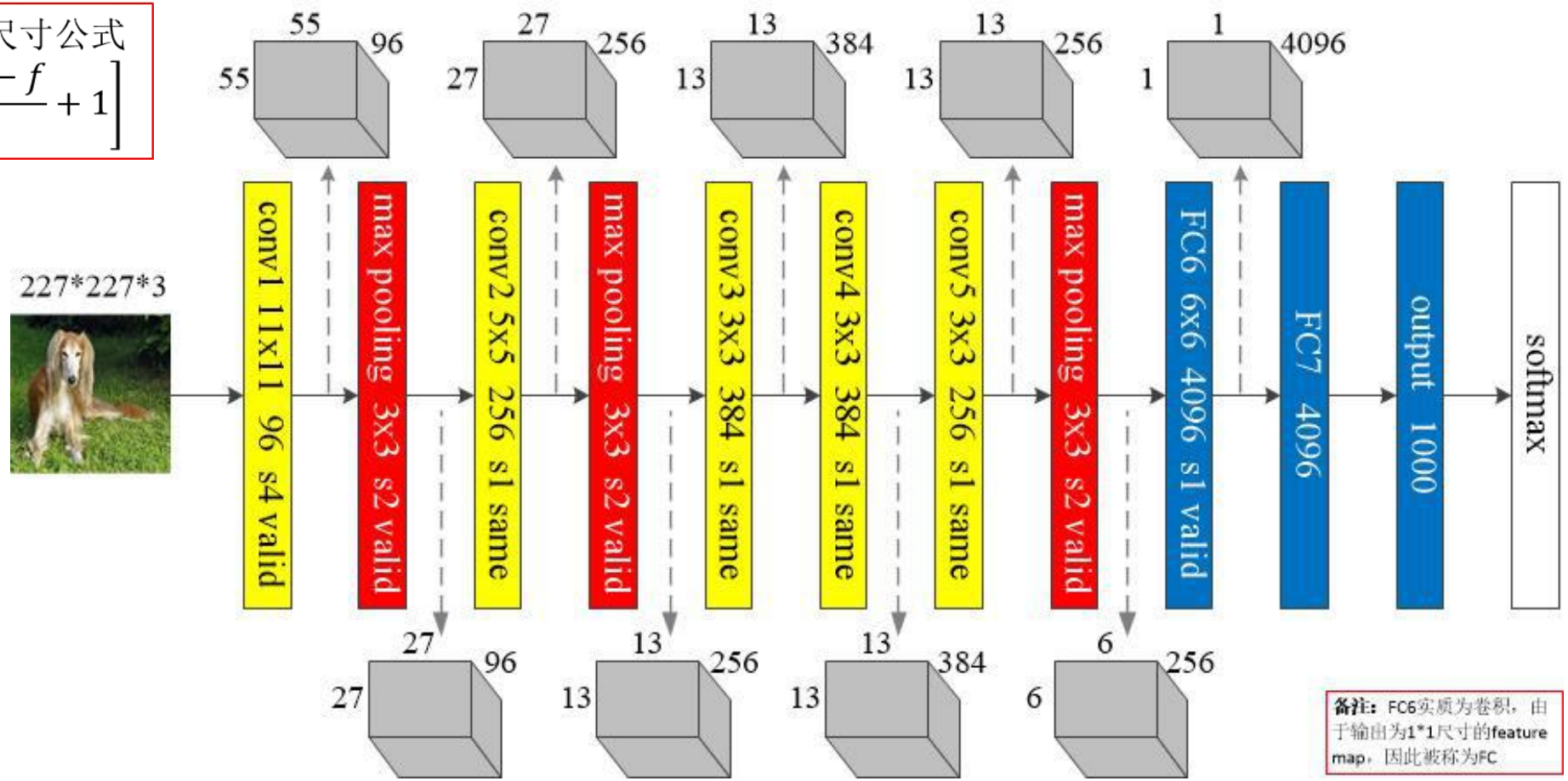
1、确立了
先卷积
后池化
最后全连接
的套路，沿用至今

2、模式设计尚且稚嫩
激活函数为Tanh
池化也有对应权值

3.1：AlexNet－深度CNN与BD的首次触电

输出FM尺寸公式

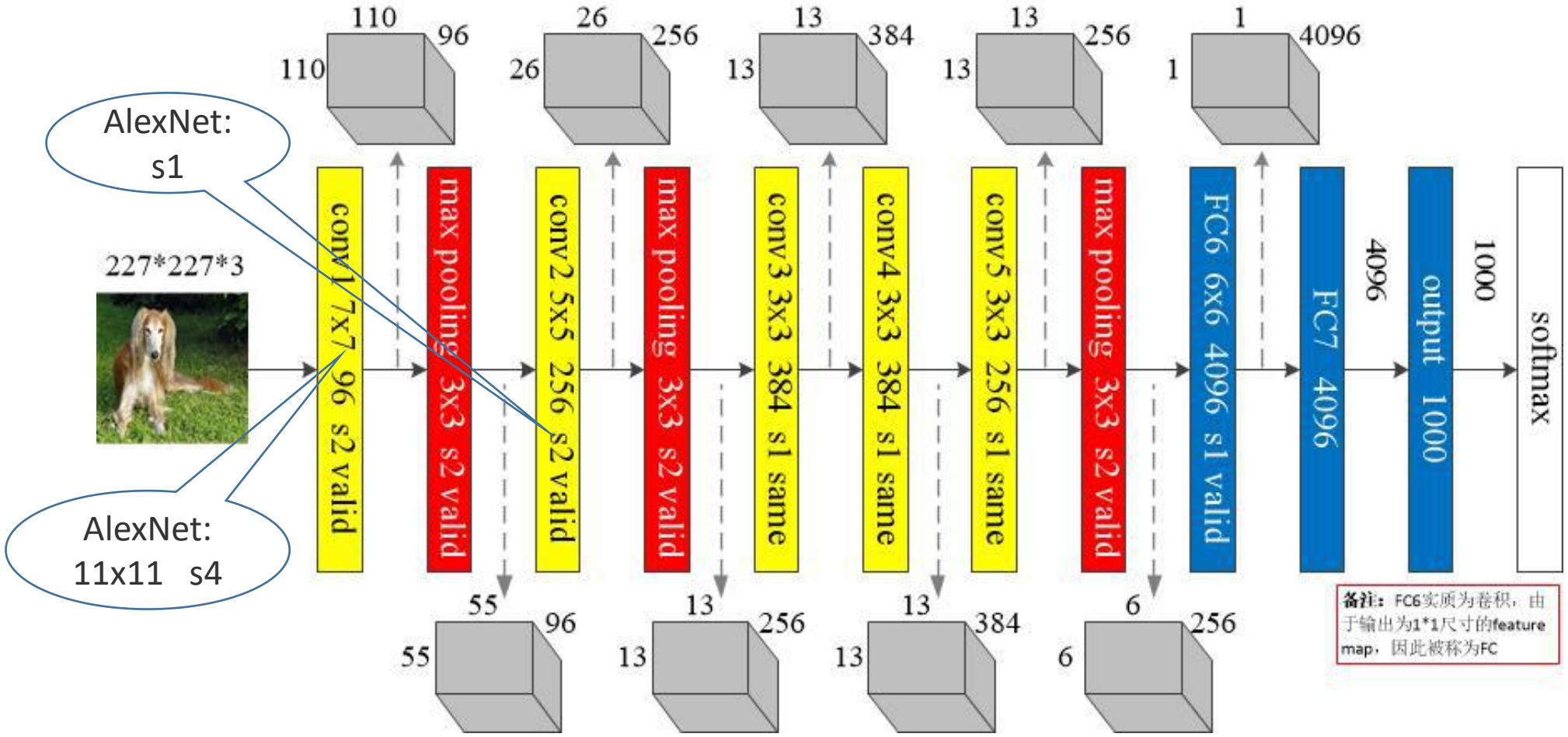
$$\left[\frac{n+2p-f}{s}+1\right]$$



共8层，5（卷积+池化）+3（全连接）

- 用ReLU解决了网络层数变深的问题
- 用数据增强、GPU训练解决了大数据的问题
- 发明了一堆没有用的东西Dropout（正在逐步推出历史舞台）、局部响应归一化（LRN）

3.2：ZFNet（2013）：过渡

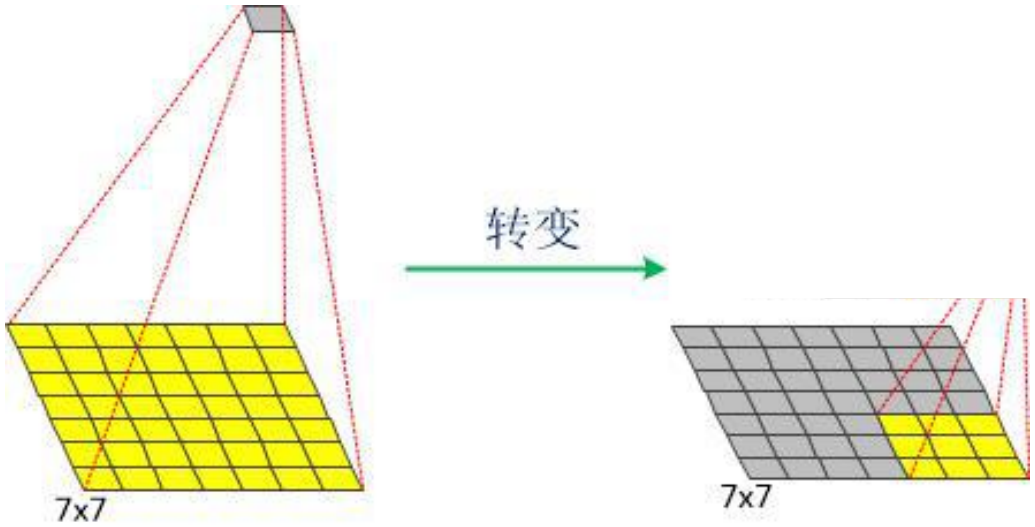


共8层，5（卷积+池化）+3（全连接）

- 帮助AlcxNet选了选超参（什么是“超参”？）

4.1：VGG（2014） – “标准模块+堆叠”

为啥选7*7就比11*11好呢？能只选一个“标准”的卷积核尺寸不？
所有卷积都用一个尺寸（f）岂不是方便很多？



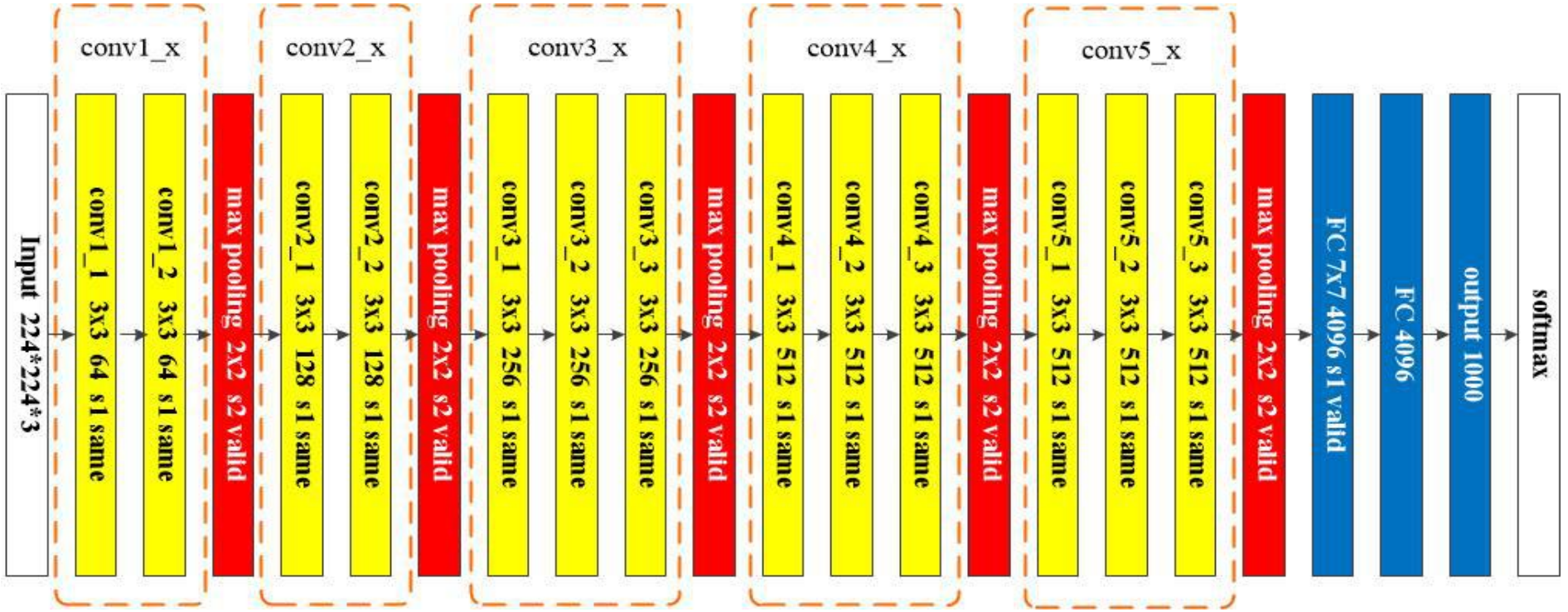
1个7x7 filter相当于3个3x3 filter

输出FM平面尺寸计算公式

$$\left[\frac{n-f}{s}+1\right]$$

- 采用连续三个3*3替代一个7*7，多经过二次非线性的激活函数，特征描述变得更加精细（参见第三讲university的证明）。
- 减少了参数的数量，7*7卷积参数为：49个；3个3*3卷积参数为：27个，相差近一倍；
- 实际上，三个3*3的卷积核权值如果选择合适，可能会完全等价于一个7*7，即输出的FM也完全相等；

4.1：VGG16 - “堆叠” block

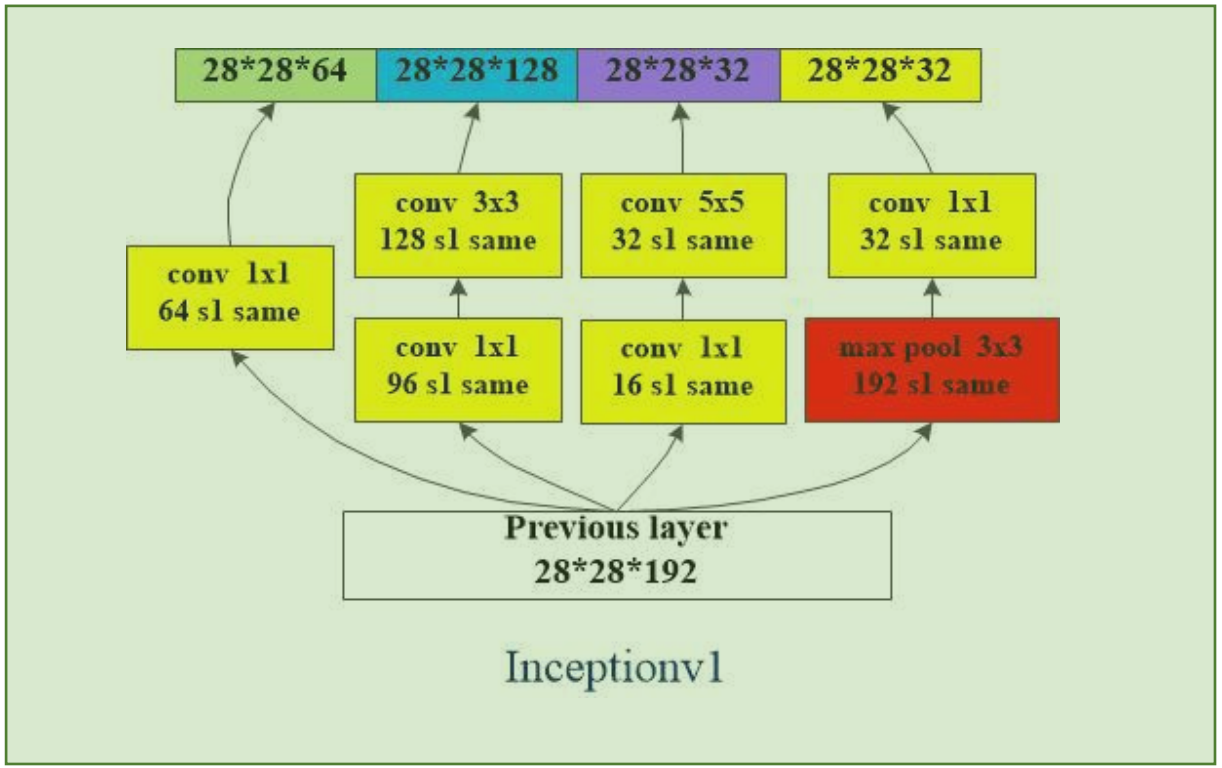


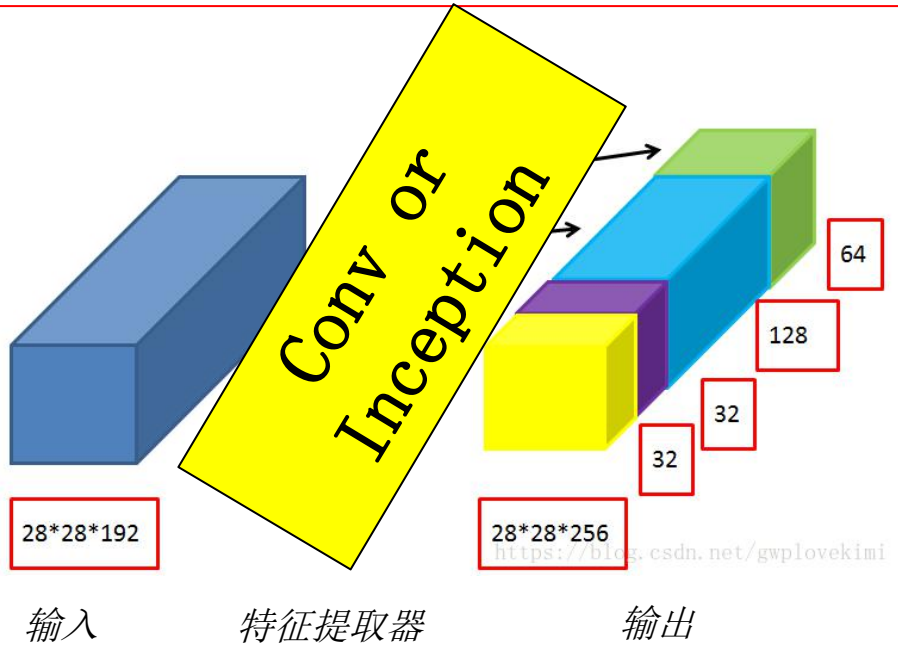
共16层，13（卷积/池化）+3（全连接），层数越深，通道数越深

- 与ZFNet相比，层次加深，使用了“标准化”的block结构，所有卷积核的平面尺寸均为3*3
- 模型中的3*3是否等价成别的卷积核了？选择的权利已经交给了模型自己
- VGG的网络设计涉嫌“偷懒”，人选择的超参少了，但机器需要做的运算多了

4.2：GoogleNet －关键词“手动定制”

Inception：一个“定制”的卷积姿势

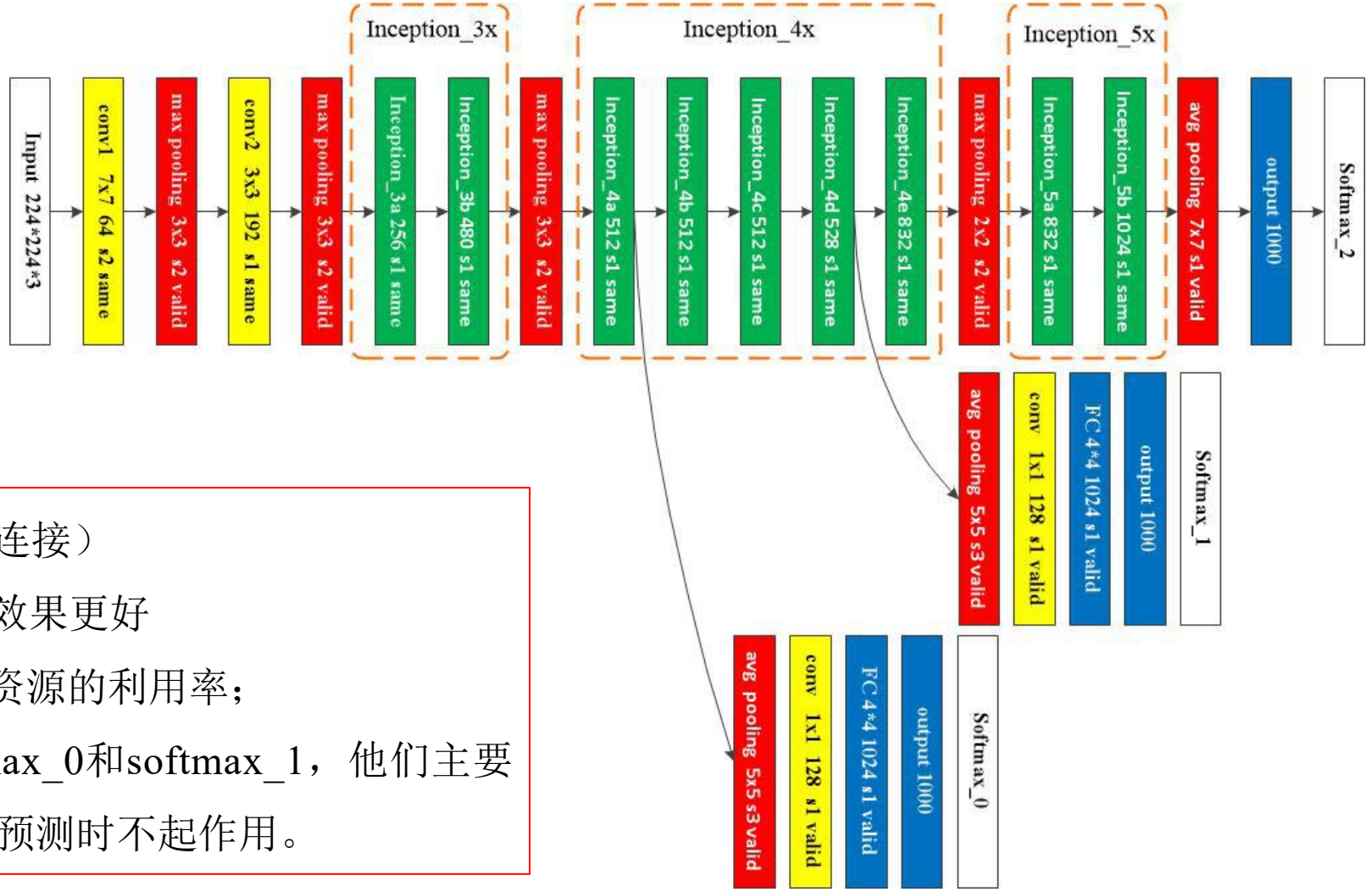




- 采用了1*1的卷积核来降低channel的维度，所以进一步减少了参数。
- 多个尺度的filter滤波，然后“concatenation”，符合视觉图像处理的特点
- 缺点是需要选取的“超参”多了好多，人们设计网络时更辛苦了

4.2：GoogleNet-Iception V1（2014）

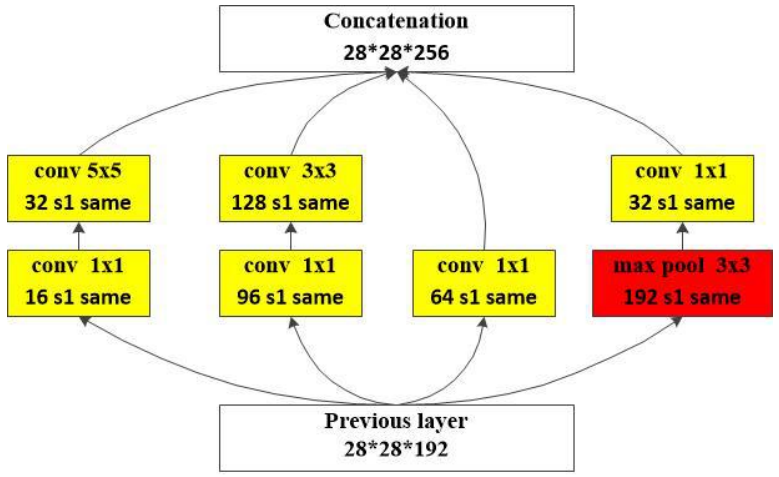
GoogleNet-Inceptionv1：



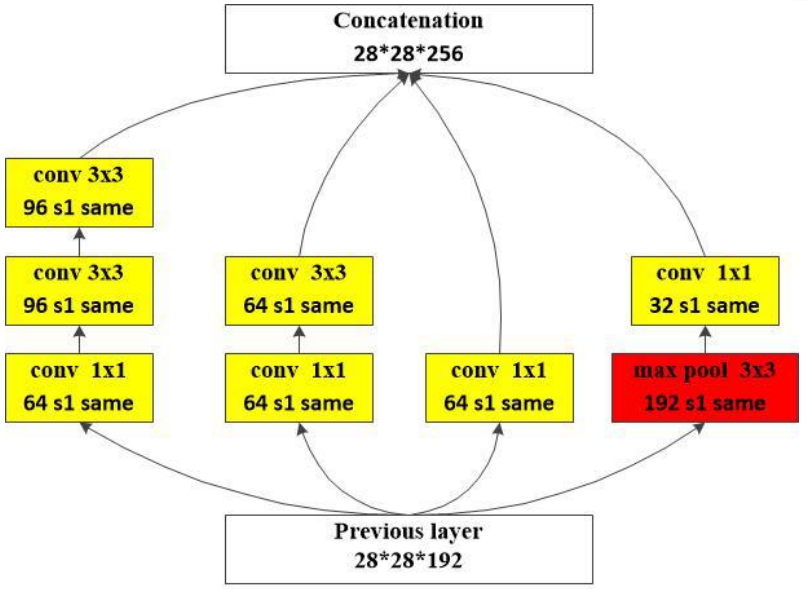
共22层，21（卷积/池化）+1（全连接）

- 与ZFNet和VGG相比，层数更深，效果更好
- 引入inception结构，够提升了计算资源的利用率；
- 增加了2个额外的辅助分类器softmax_0和softmax_1，他们主要的作用是回传误差（delta），但在预测时不起作用。

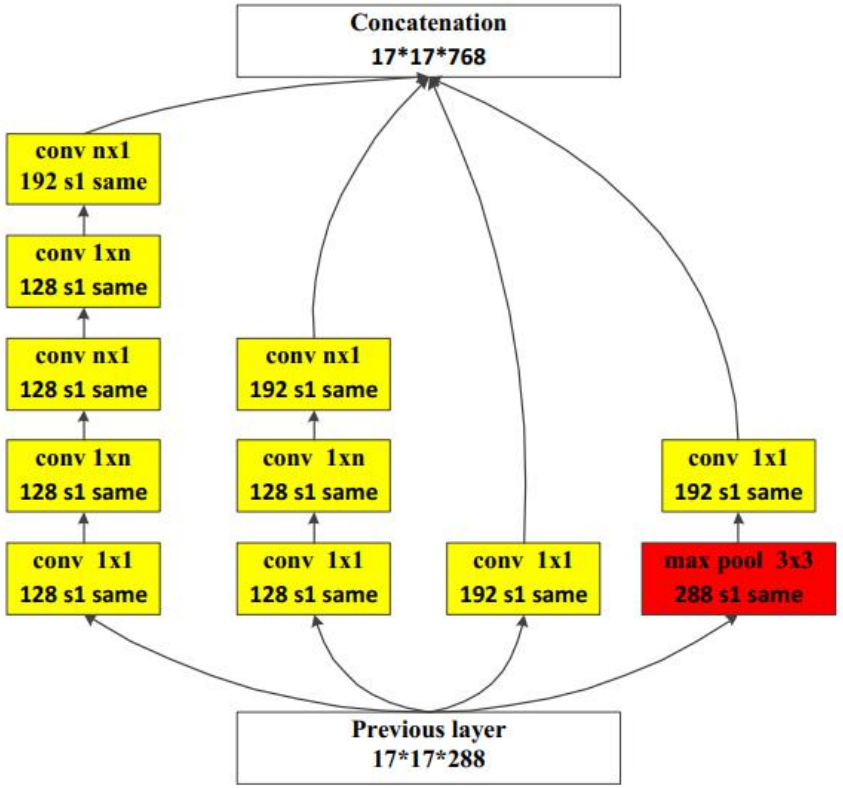
4.2：GoogleNet家族



Inceptionv1



Inceptionv2

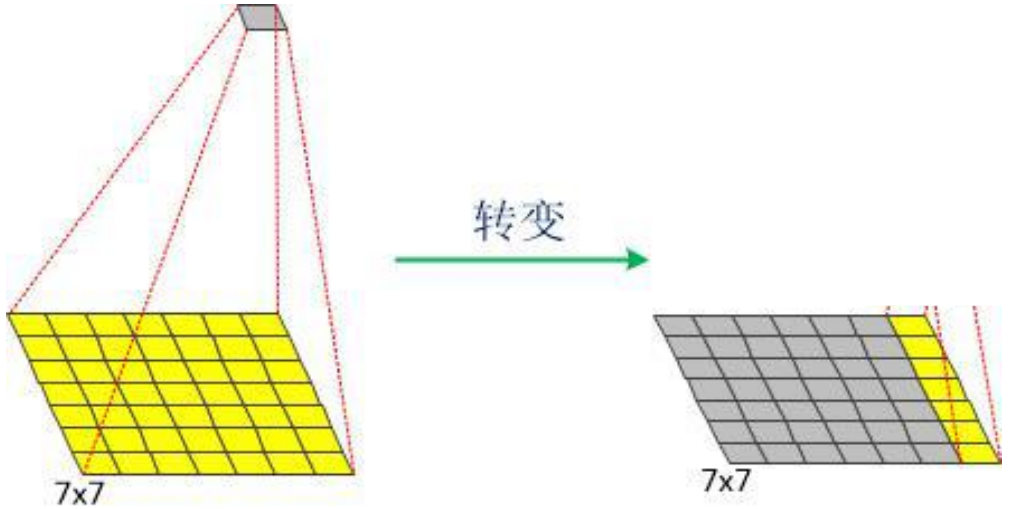


Inceptionv3

- V1版本提出了Inception的概念，大胆使用了1*1的卷积核来压缩通道数
- V2版本借鉴了VGG的理念（定制Inception时，在其内部采用标准化卷积核）
- V3（2015）版本将VGG的理念发扬光大，将“标准化”推广到一般情况，并加入了BN；
- V4（2016）版本在V3基础上选定了更合适的超参，没有引入残差的情况下，网络层数仍旧达到了76层。

4.2：Inception V3 卷积核分解

既然三个3*3等价于一个7*7
那有没有更纯粹、更一般的卷积核等价分解方法？

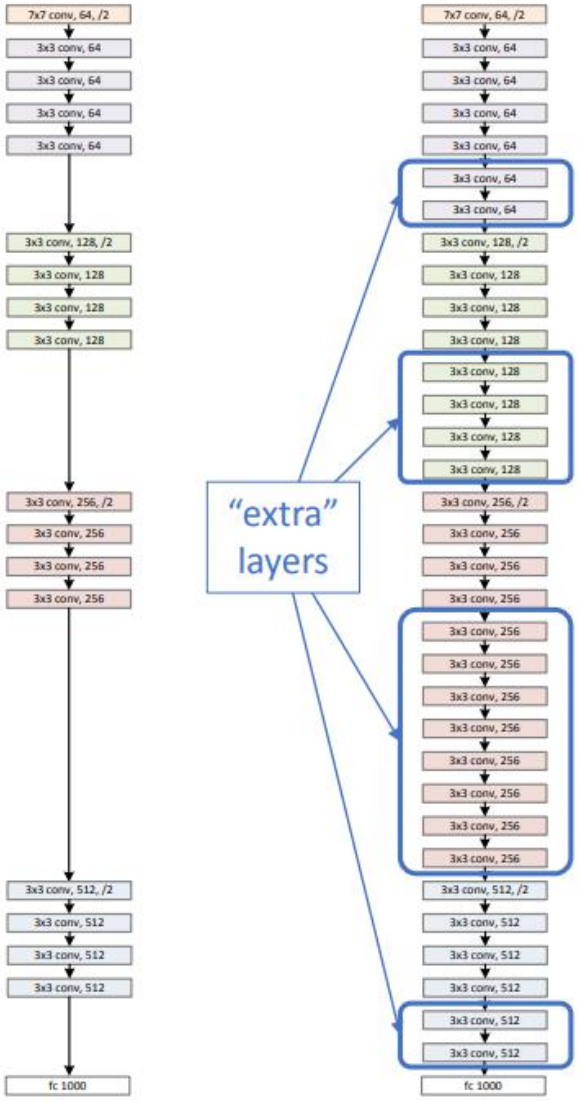


1个7x7 filter相当于7x1和1x7 filter

- Inception-v3将VGG中卷积核（ filter ）分解的思想发扬光大，一个n*n卷积核，可分解为一个1*n和一个n*1
- 进一步减少了参数的个数，7*7共有49个参数，而一个7*1和一个1*7共有14个参数。

5：网络可以有多深？

a shallower model (18 layers)



a deeper counterpart (34 layers)

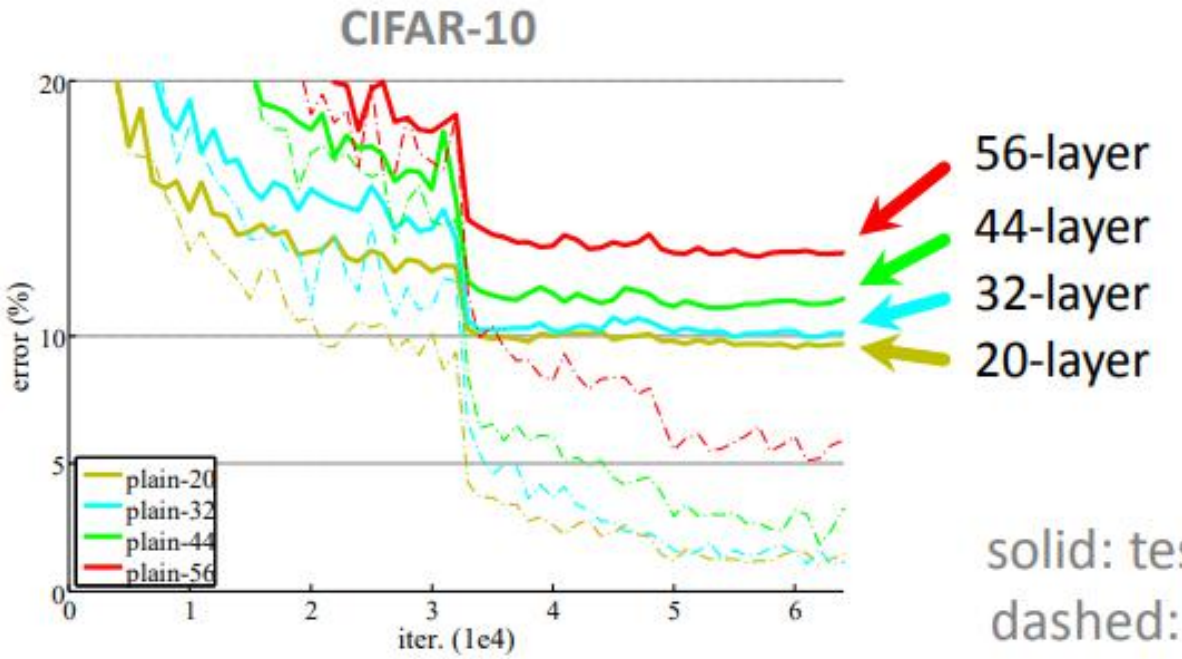
理论上网络越深越好

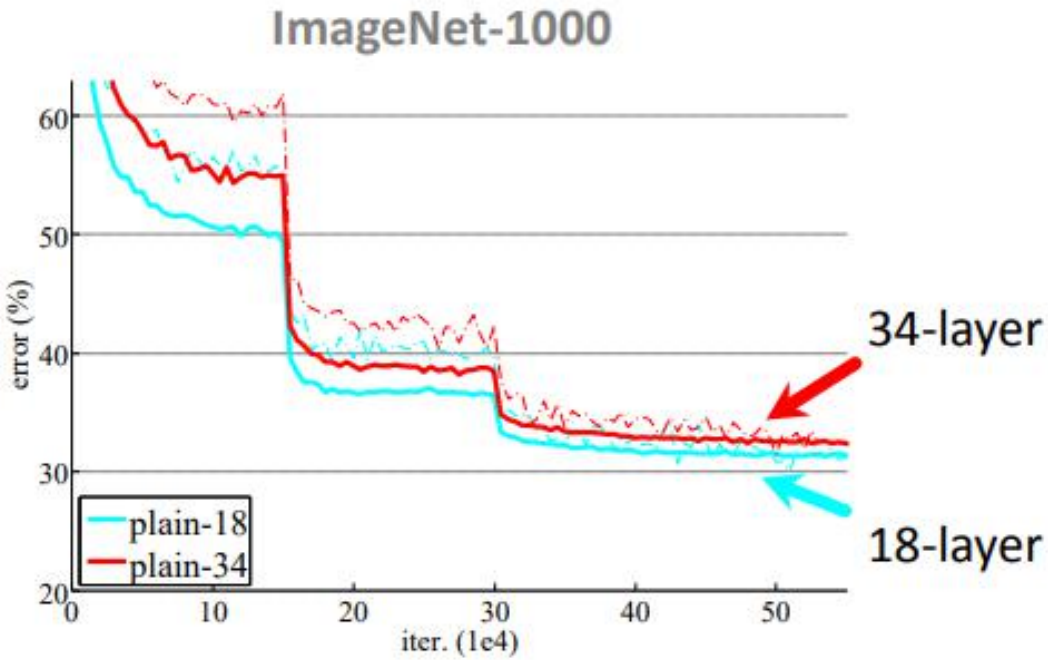
- A deeper model should not have **higher training error**
- A solution *by construction*:
  - original layers: copied from a learned shallower model
  - extra layers: set as identity
  - at least the same training error
- Optimization difficulties: solvers cannot find the solution when going deeper…

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". arXiv 2015.

5.1：网络退化

实际上，强行加深
会出现网络退化问题





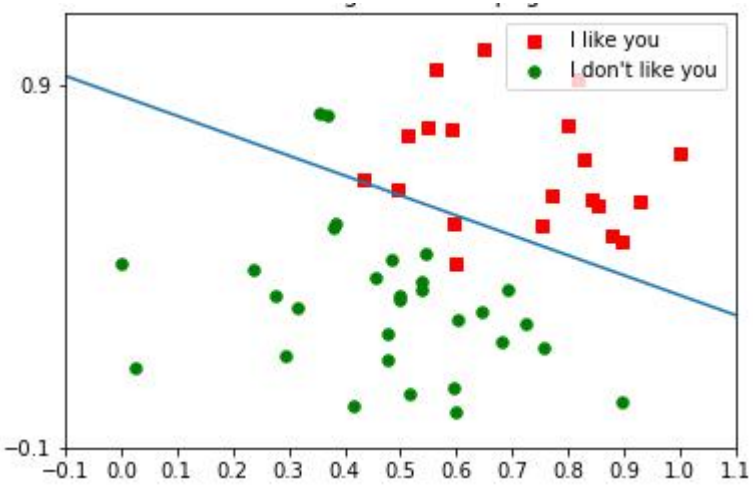
5.2：网络退化的原因？

造成网络退化的原因是什么？

参考文献
【1】He K, Zhang X, Ren S, et al. Deep residual learning for imagerecognition[C]//Proceedings of the IEEE conference on computer vision andpattern recognition. 2016: 770-778.
【2】Srivastava R K, Greff K, Schmidhuber J. Highway networks[J]. arXiv preprintarXiv:1505.00387, 2015.
【3】Orhan A E, Pitkow X. Skip connections eliminate singularities[J]. arXivpreprint arXiv:1701.09175, 2017.
【4】Shang W, Sohn K, Almeida D, et al. Understanding and ImprovingConvolutional Neural Networks via Concatenated Rectified Linear Units[J].2016:2217-2225.
【5】Greff K, Srivastava R K, Schmidhuber J. Highway and Residual Networks learnUnrolled Iterative Estimation[J]. 2017.
【6】Jastrzebski S, Arpit D, Ballas N, et al. Residual connections encourageiterative inference[J]. arXiv preprint arXiv:1710.04773, 2017.

翻译一下

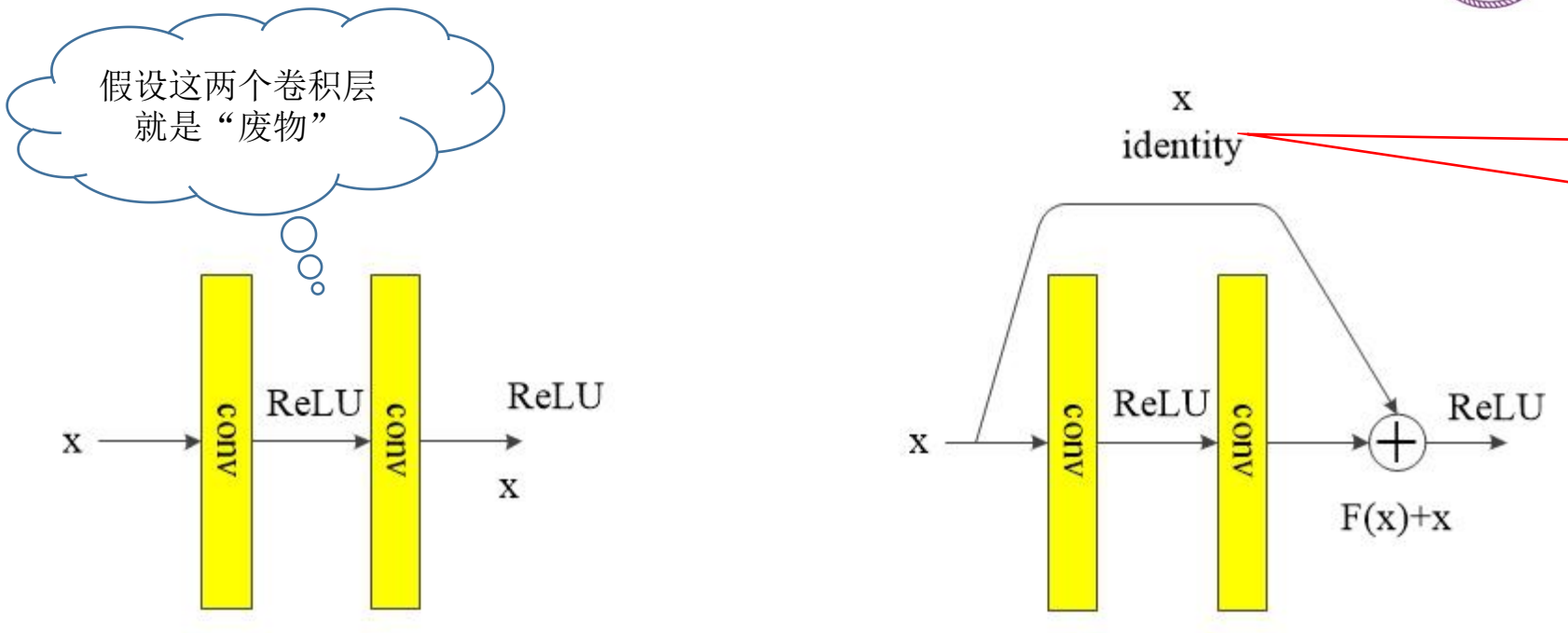
- 具体为啥，我们不能十分确定，完备、公认的数学证明目前还没有
- 肯定不是因为梯度消失，反向传播没出问题
- 肯定不是因为信号前馈，前馈传播也没出问题



感性理解：在越多维的空间中画出决策边界越难

$$\theta^T X = \theta_0 + \theta_1 x_1 + \theta_2 x_2 = -5.6 + 4.2x_1 + 6x_2$$

5.3：ResNet(2015)- 打破限制，超越人类



“废物检测器”
请原有卷积单元通过学习
自证不是“废物”

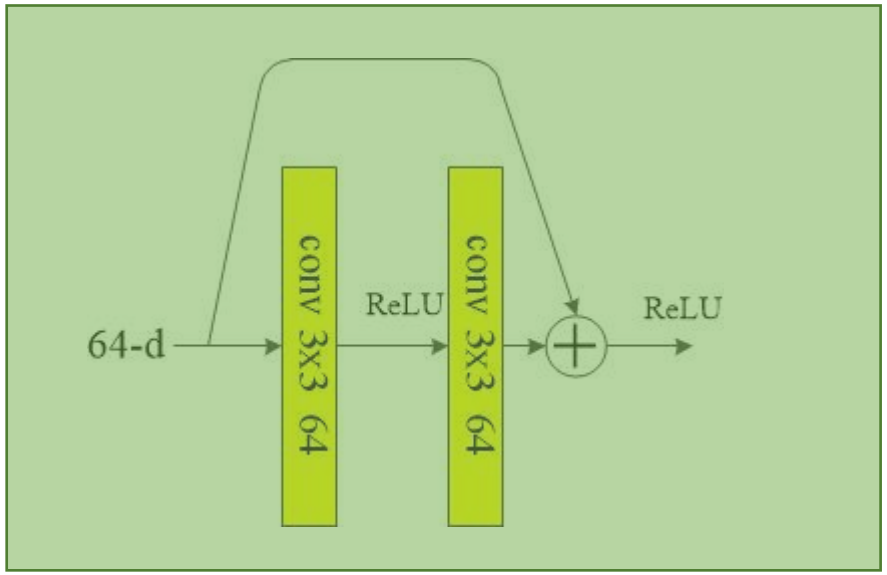
residual block示意图

原论文中的解释：
想学成“identity”难度很大
但把所有参数学成“0”相对容易

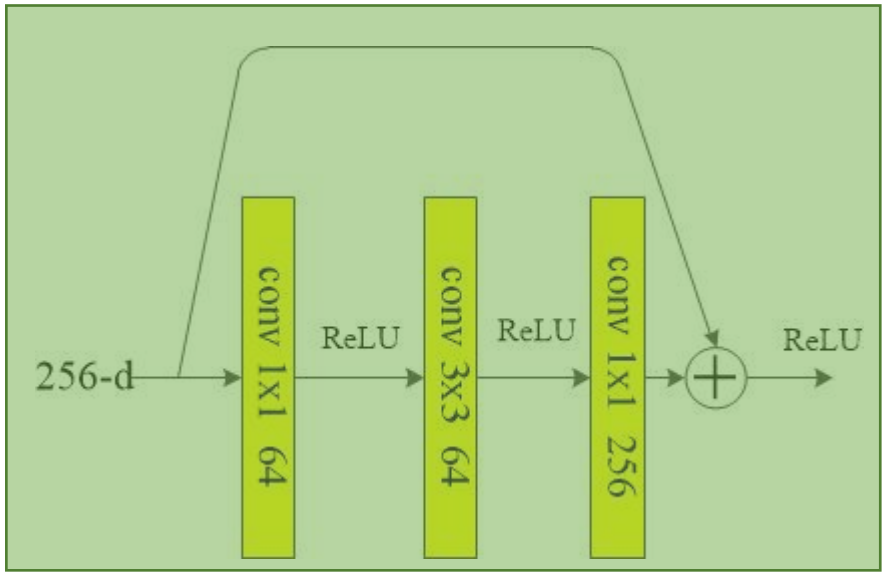
操作办法：

1. 引入shortcut connnection，将原有卷积单元转变为残差单元来解决退化问题；
2. shortcut connnection：将输入的浅层信号，直接接到输出FM中，元素级相加；
3. 这要求输出FM的尺寸与输入的x尺寸必须完全相等

5.3：ResNet(2015)- 打破限制，超越人类

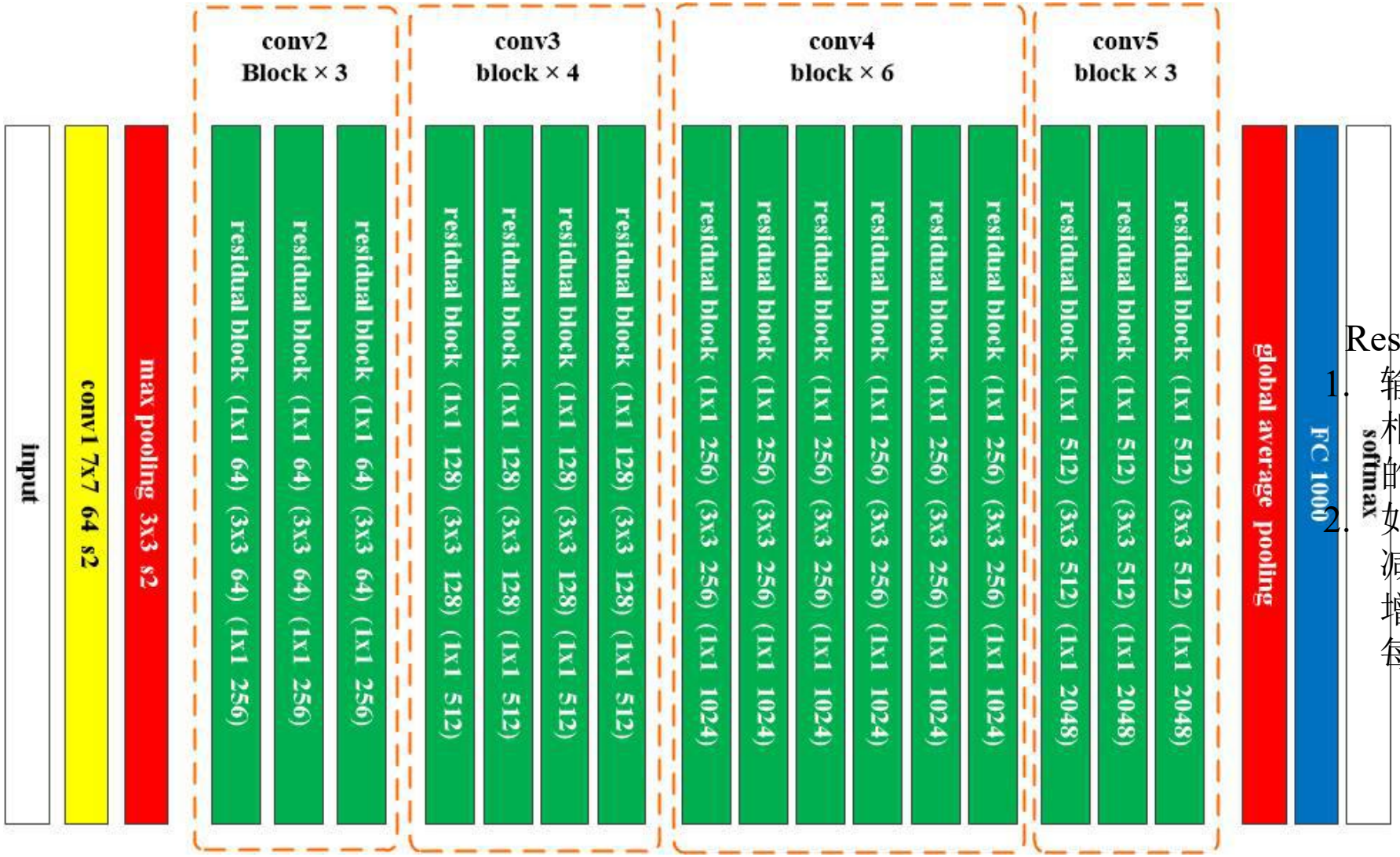


ResNet34- building block
借鉴了VGG结构



ResNet50/101/152 building block
借鉴了GoogleNet中1*1卷积核设计，参数更少

5.3：ResNet(2015)- 打破限制，超越人类

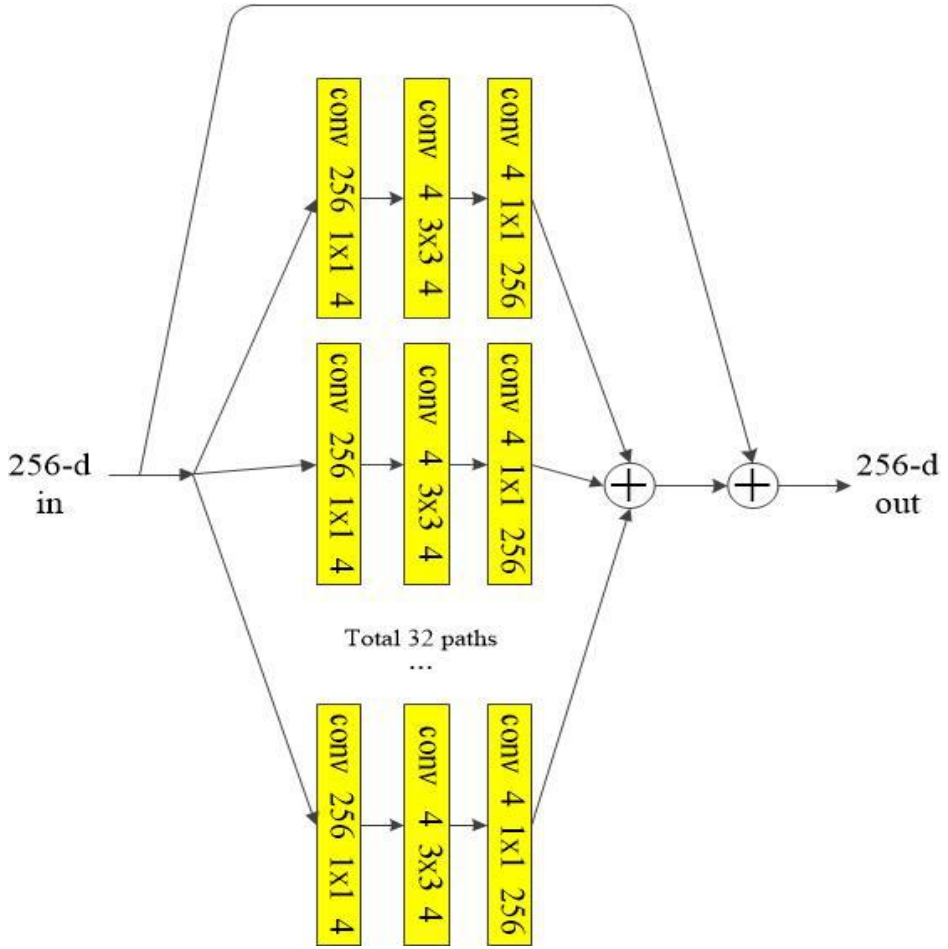


ResNet50 示意图：

1. 输出feature map 尺寸相同的卷积层，filter的数量相同；
2. 如果feature map尺寸减半，则filter的数量增加一倍，用于保证每层的时间复杂度。

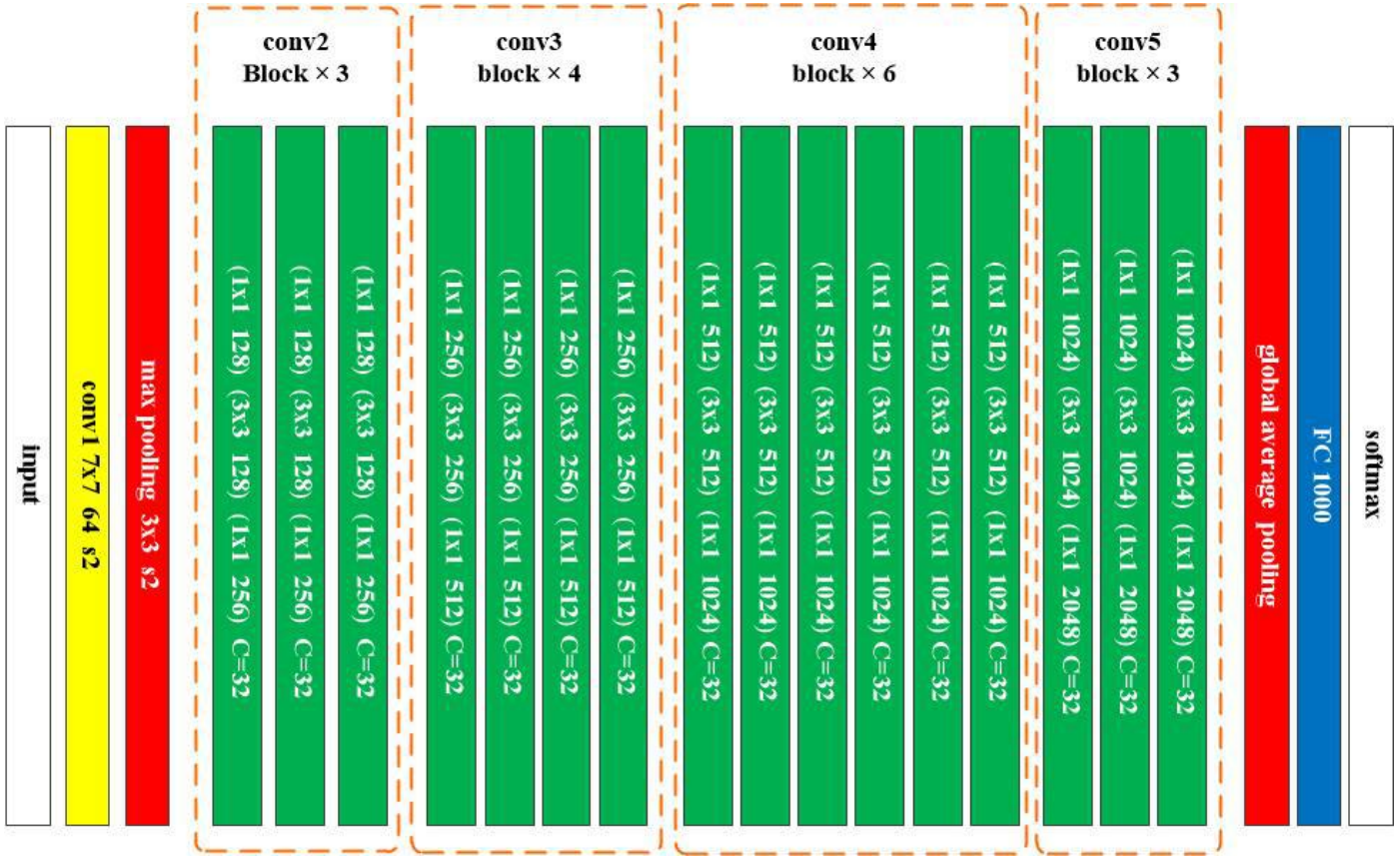
打破了“深度限制”，让算法在分类任务上的表现超过了人类

5.4：ResNeXt（2016）：ResNet Plus



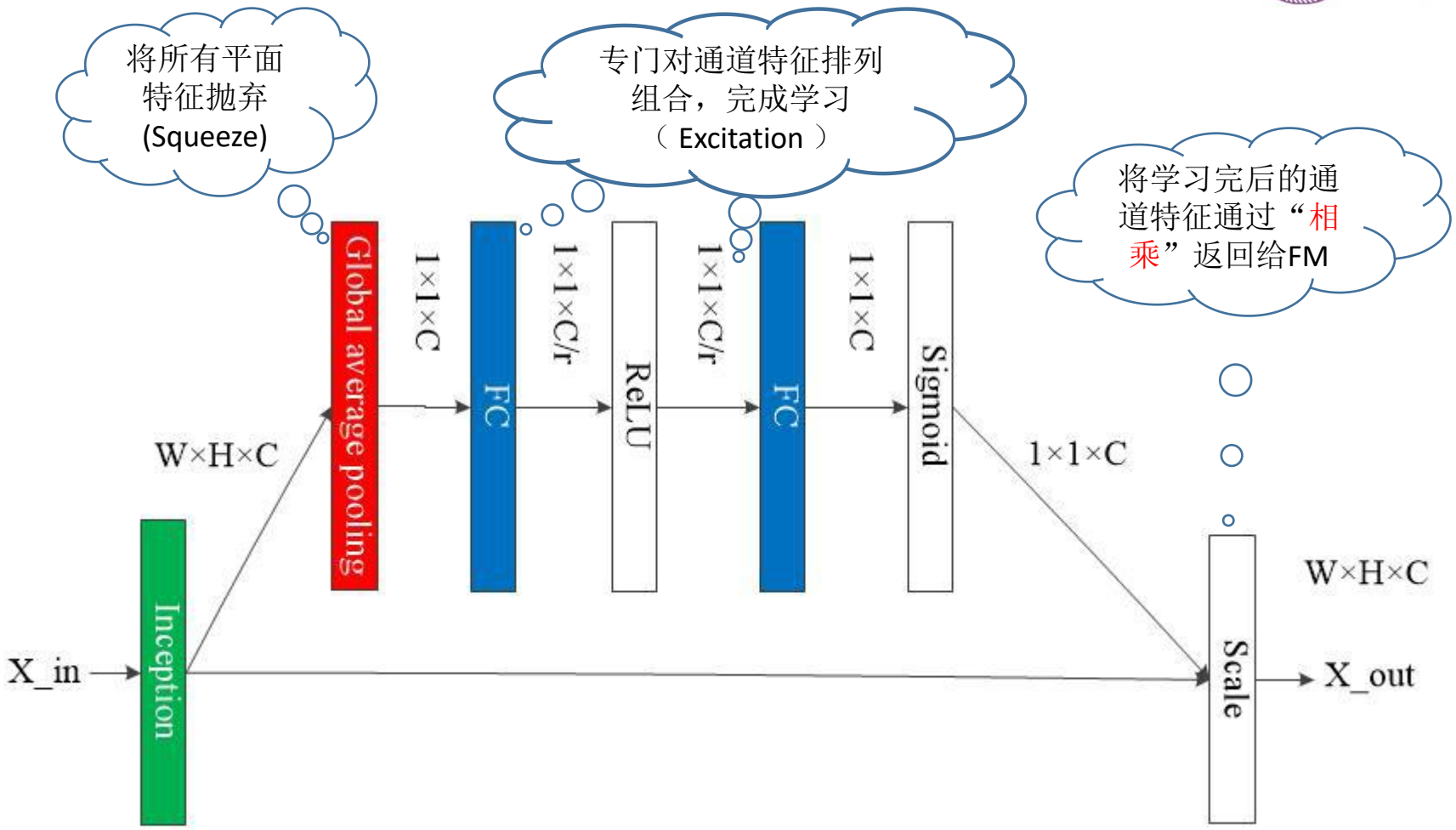
将ResNet的building block进一步“定制”
完全融合了GoogleNet和VGG的设计思想
将单元变“宽”了，但需要选择的超参也不多

5.4： ResNeXt



- 由于完全融合了GoogleNet和VGG的设计思想，与 ResNet 相比，相同的参数总量，ResNeXt 结果更好；

6.1： SENet（2017）- 集大成者

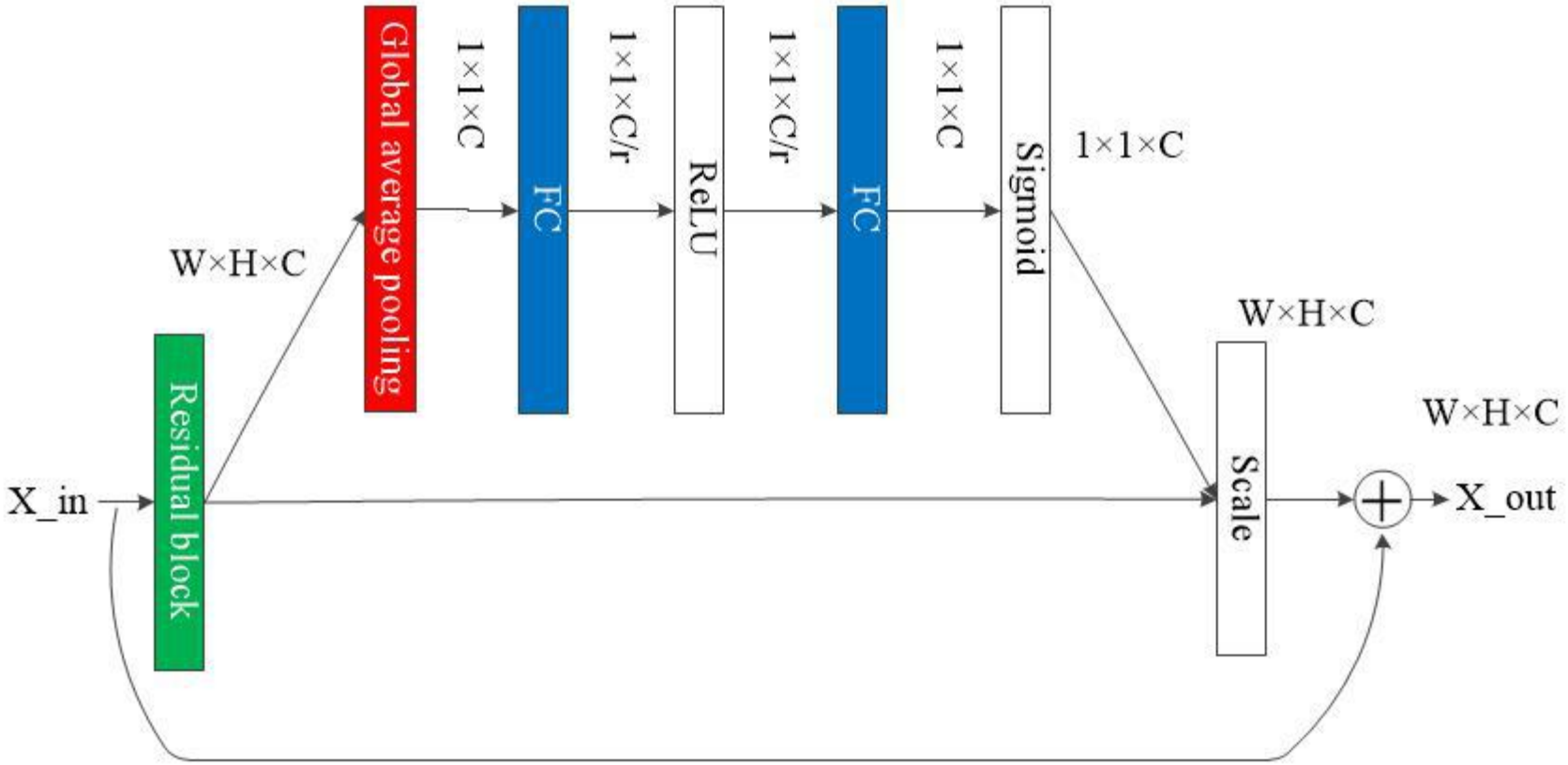


将SE 模块嵌入到 Inception 结构的示例

特点：
通过Squeeze 和 Excitation 显式地建模特征通道之间的相互依赖关系。

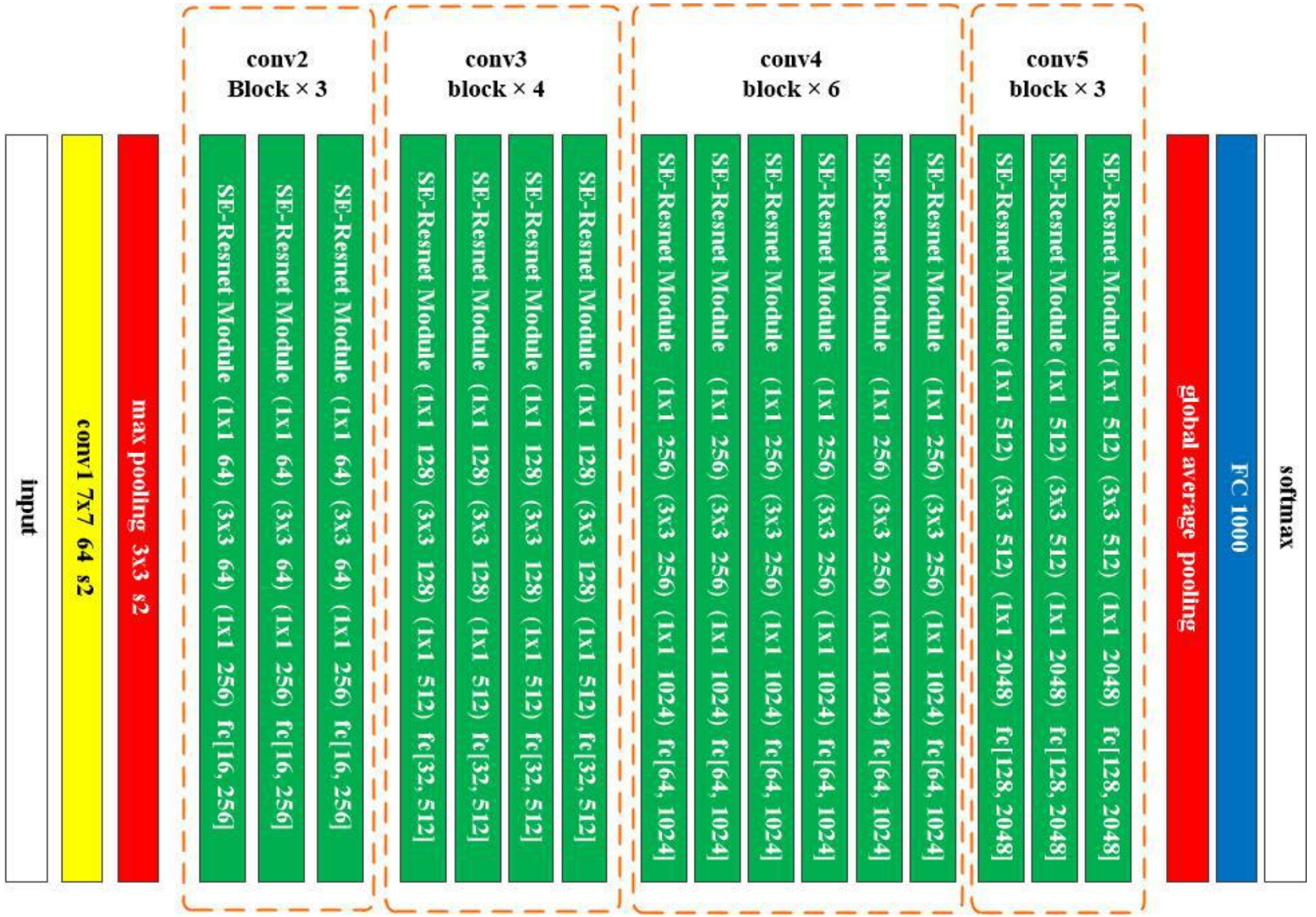
使用 global average pooling 作为 Squeeze 操作。紧接着两个 Fully Connected 层组成一个 Bottleneck 结构去建模通道间的相关性，并输出和输入特征同样数目的权重。

6.1： SENet-专门学通道特征



将 SE 嵌入到 ResNet 模块中的示例

6.1： SENet



1、结合了
VGG（卷积分解）
Inception（定制单元）
ResNet（残差）
等前人的优秀经验

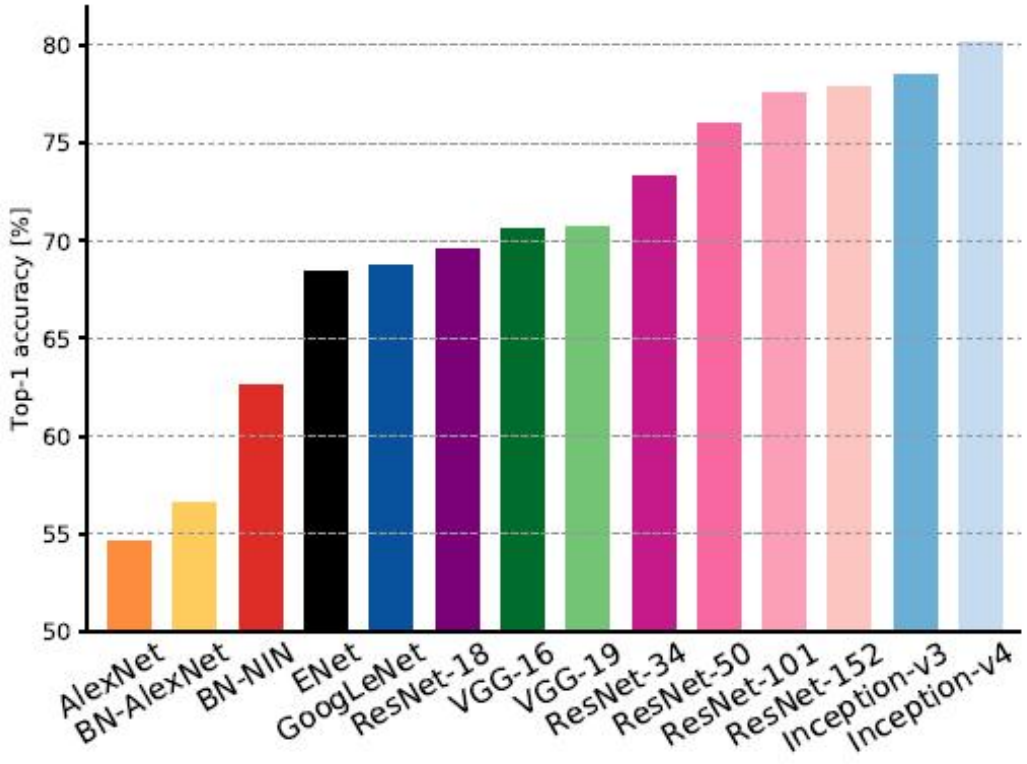
2、专门对之前被忽视的通道特征进行了专门学习，取得了最后一界ImageNet挑战赛的冠军

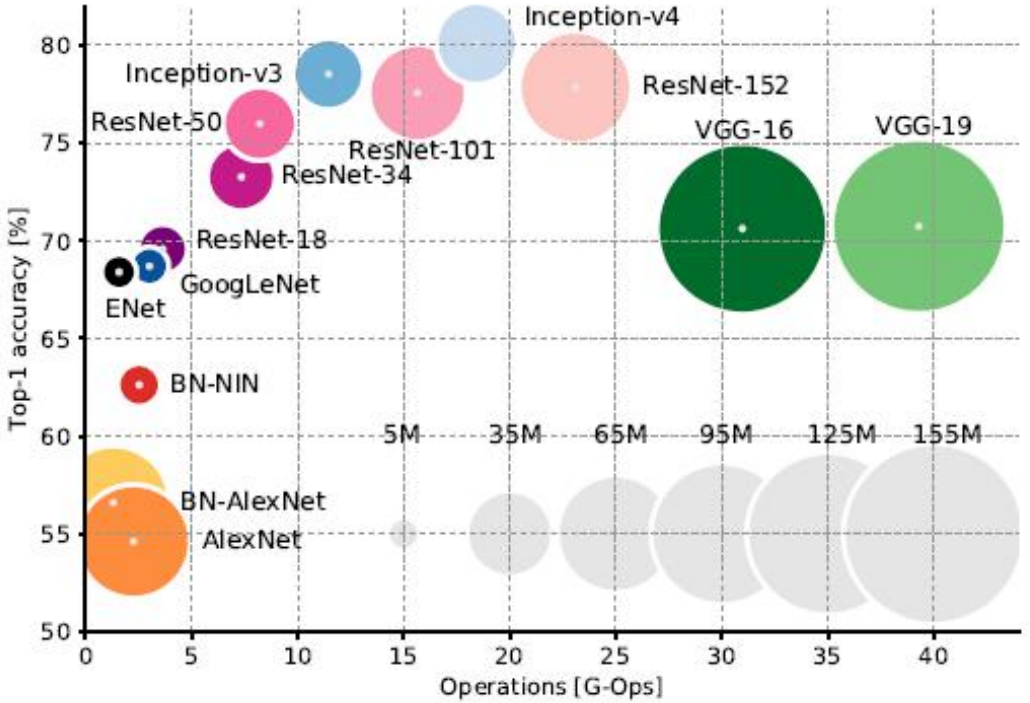
6.2：小结


分类任务
精度提高
LeNet（原点）
AlexNet（DCNN+BD）
VGG（人懒机器累）
GoogleNet（人烦机器爽）
ResNet（解除网络深度封印）
SENet（补全通道特征）
速度变快
模型压缩
Deep Compression
结构优化
MobileNet
ShuffleNet

6.2：小结

Canziani A, Paszke A, Culurciello E. An analysis of deep neural network models for practical applications[J]. arXiv preprint arXiv:1605.07678, 2016.





1、AlexNet发布于2012年，Inception-v4发布于2016年。4年时间算法的进步令人咋舌；

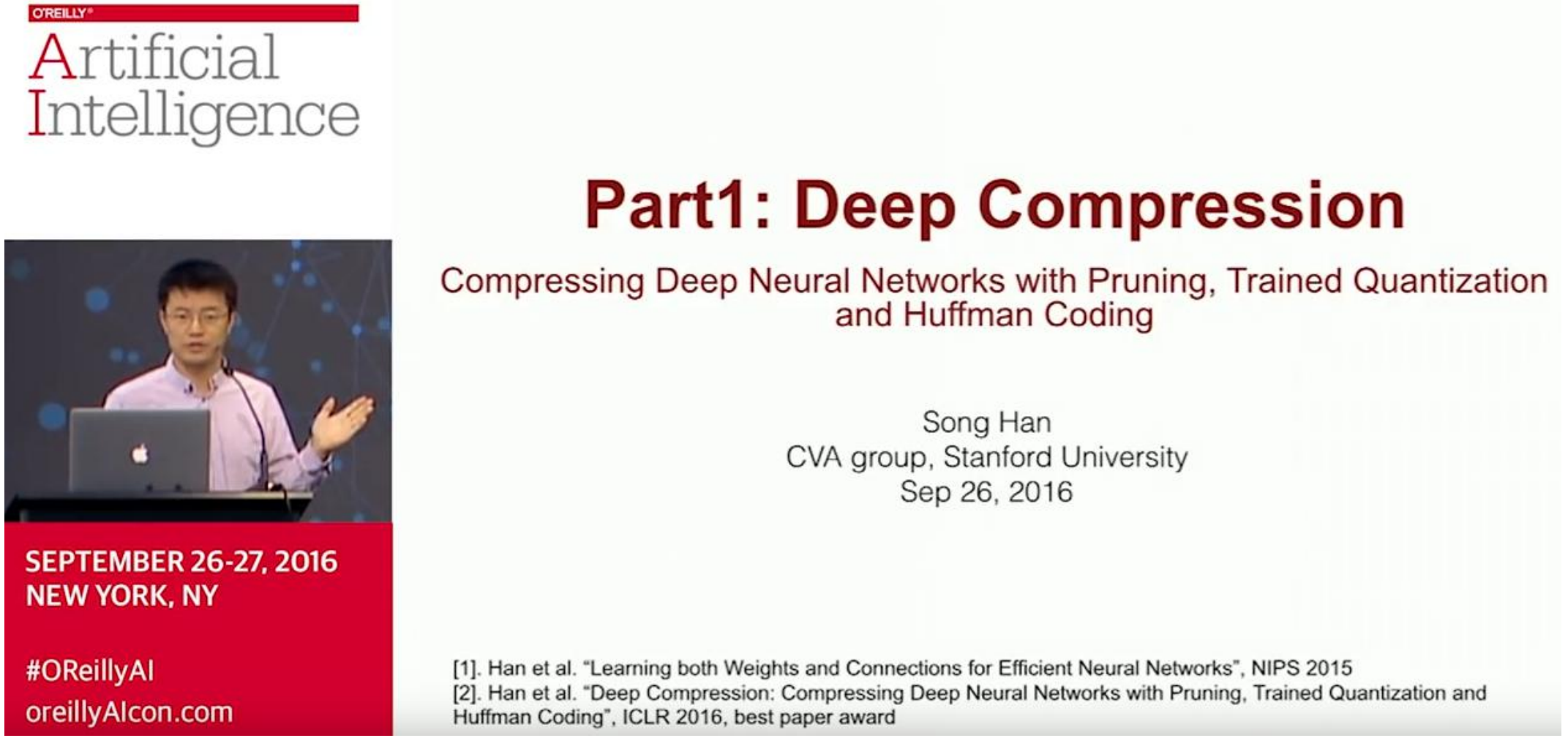
2、右图圆形的大小代表参数的总量，单位是million。层数最深的是ResNet-152，参数最多的是VGG-19；

3、并不是参数越多精度越高，也不是网络越深精度越高。真正发挥作用的指标是“参数效率”

同样是参数，效率还能不一样呢？

7.1：模型压缩-Deep Compression

源自ICLR 2016 best paper



7.1：Deep Compression

这种模型压缩的方法有多厉害？

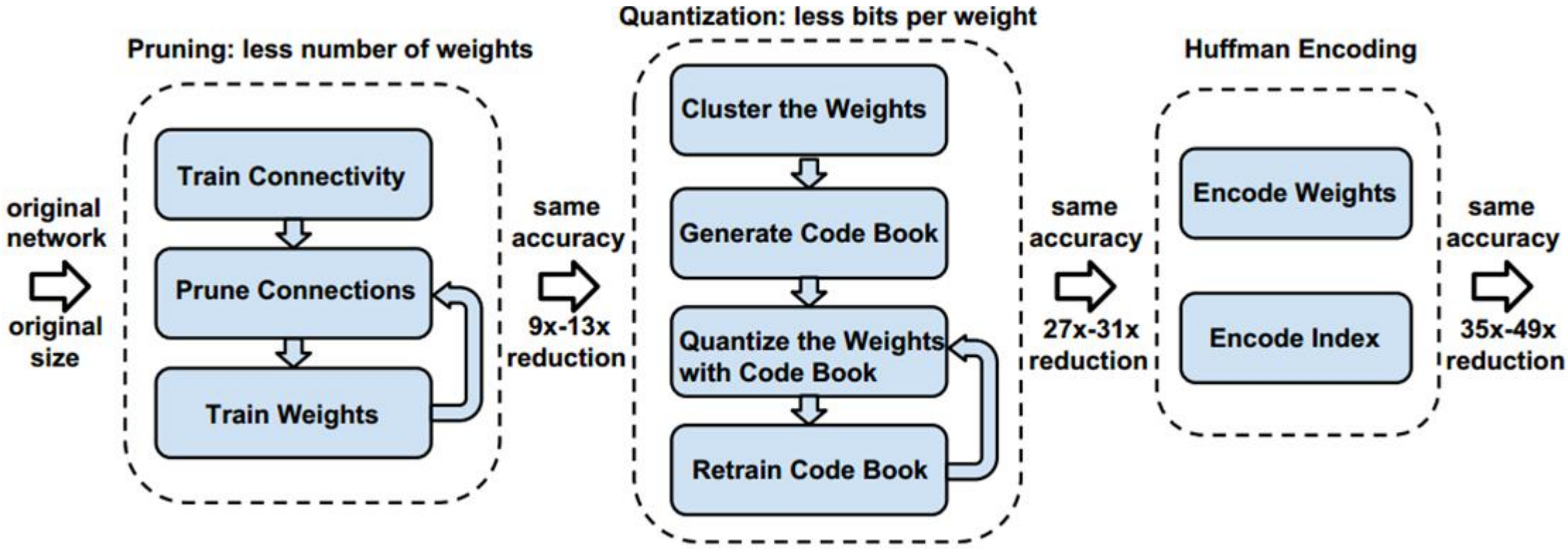
关注一下压缩比，GoogleNet的效率确实比VGG高得多

Deep Compression Overview

- AlexNet: 35x, 240MB => 6.9MB
- VGG16: 49x, 552MB => 11.3MB
- GoogLeNet: 10x, 28MB => 2.8MB
- SqueezeNet: 10x, 4.8MB => 0.47MB
- No loss of accuracy on ImageNet12
- Weights fits on-chip SRAM cache, taking 120x less energy than DRAM memory

7.1 ： Deep Compression

具体应该怎么做？

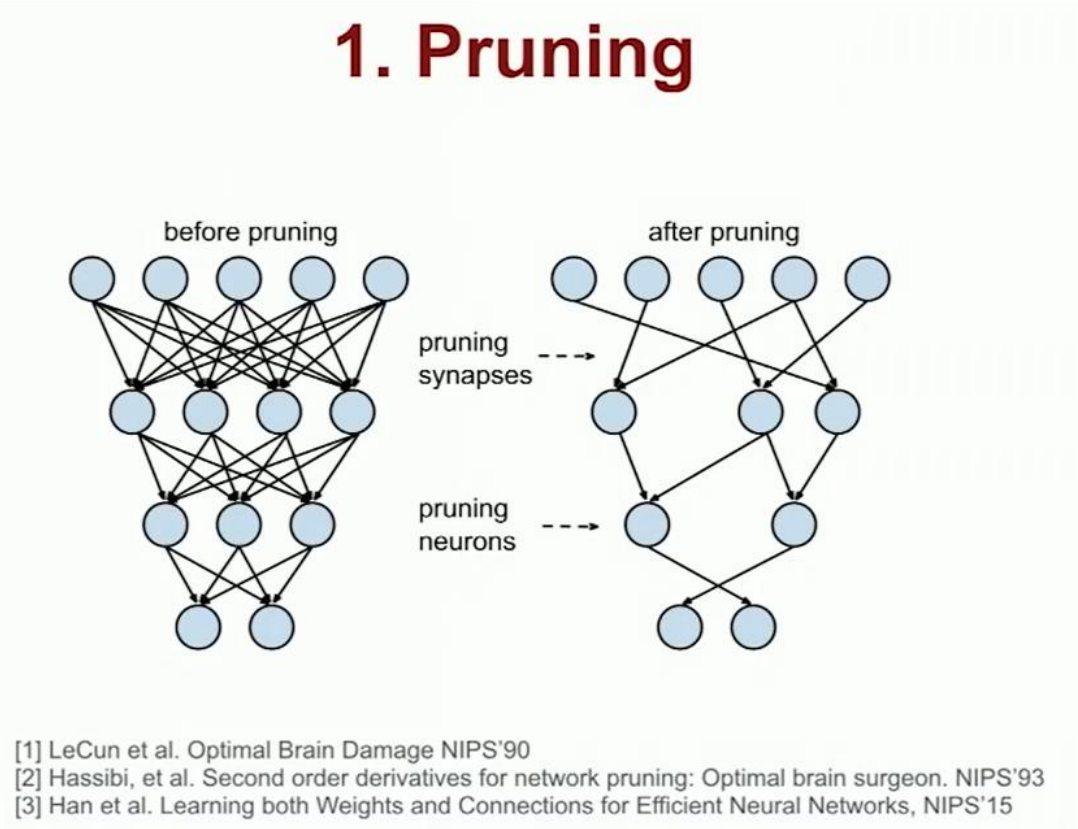


操作办法：

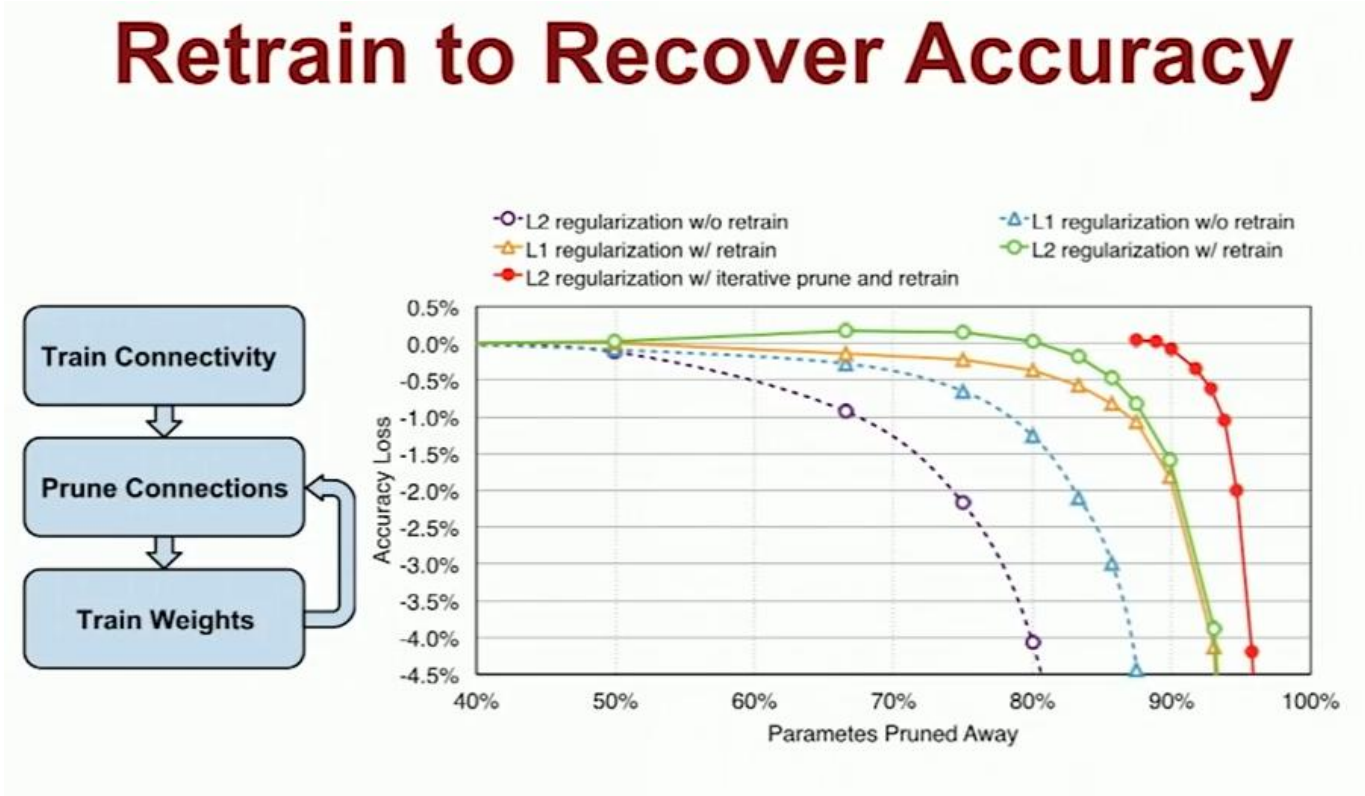
1. 直接剪支+再训练；
2. 参照通信原理进行编码储存；
3. 哈夫曼编码再次压缩。

7.1 ：网络可以有多快？

第一步：剪支+再训练



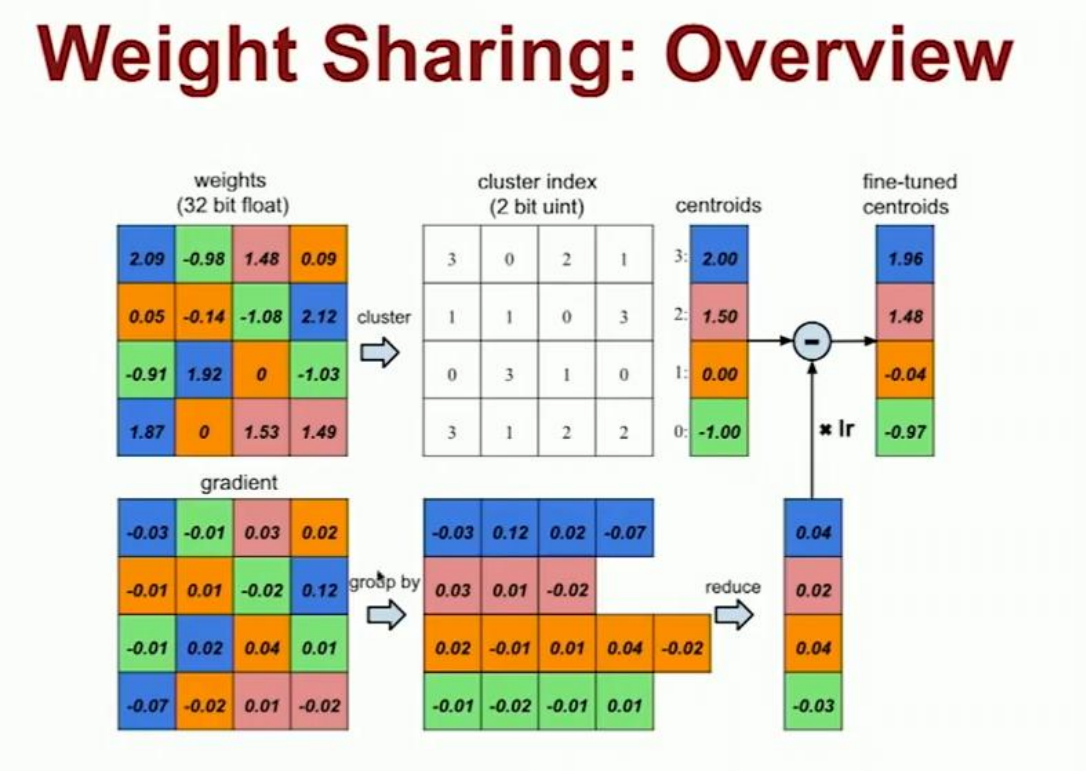
以某个特定阈值进行直接剪支

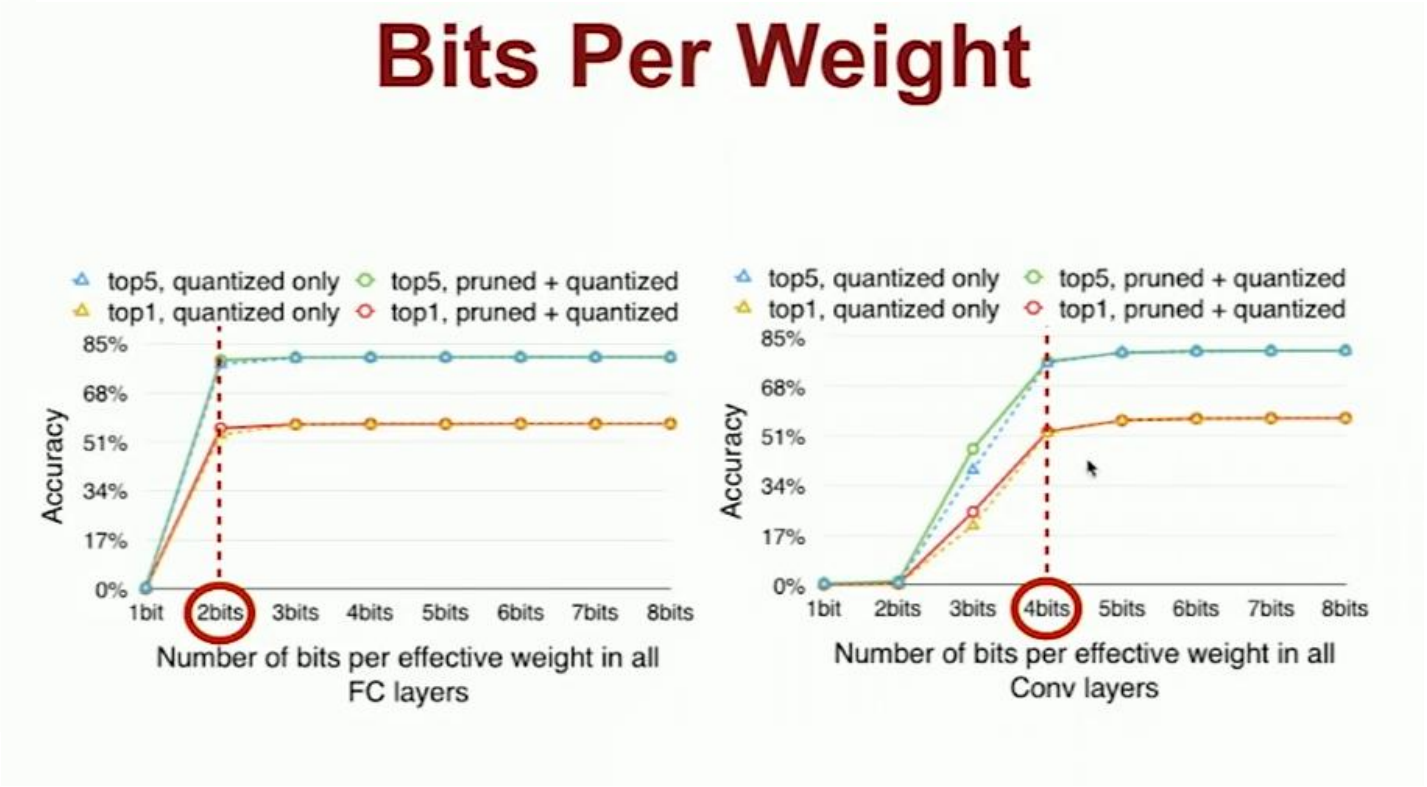


重新训练后课恢复大部分精度

7.1 ：网络可以有多快？

第二步：通过聚类压缩（量化编码）



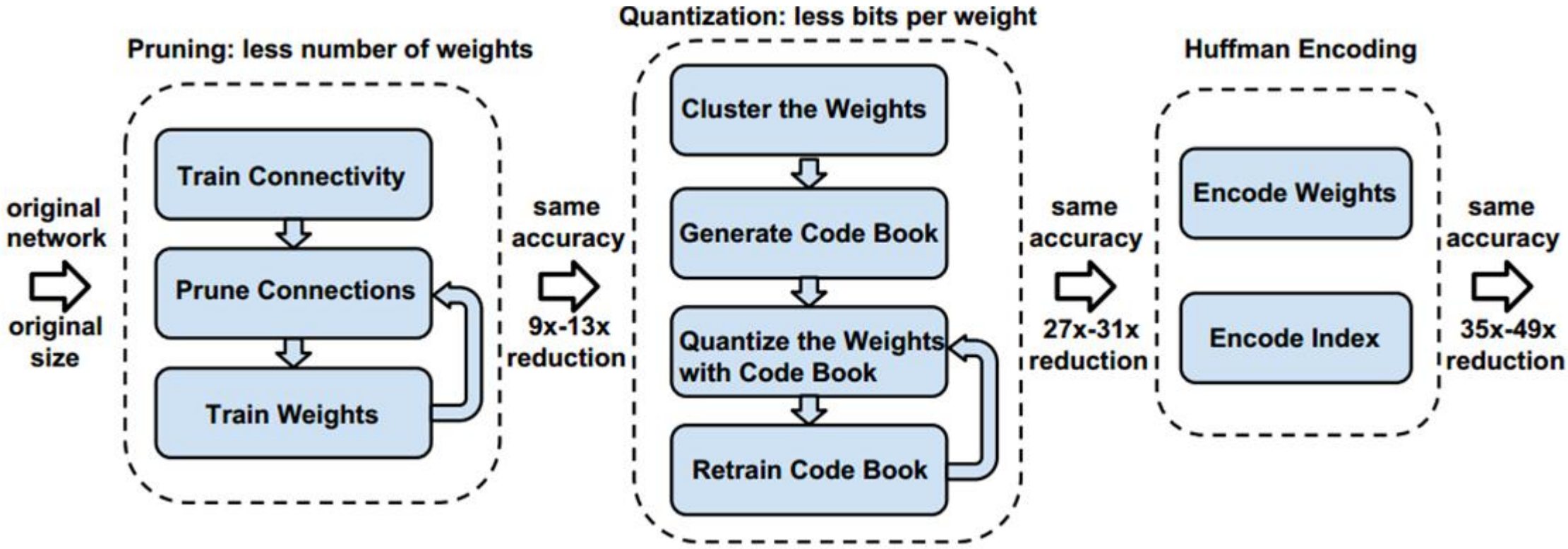


利用K-means先聚类。完成后再编码

要聚多少类？

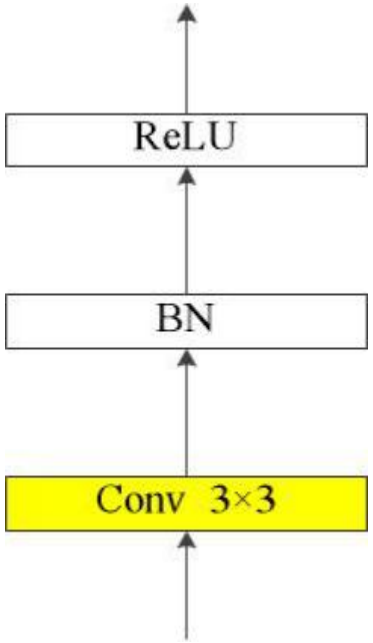
7.1 ： Deep Compression

回顾

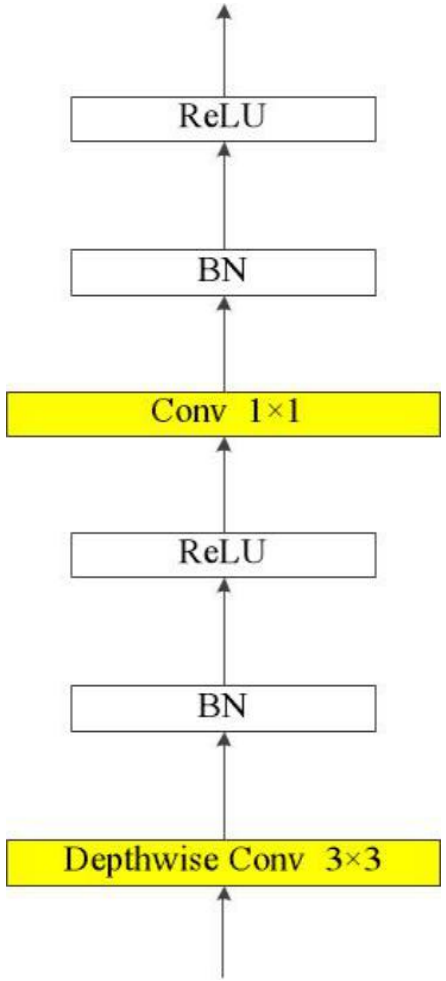


机器学习与通信学科交叉创新
结合了硬件特点，效果十分显著

7.2： MobileNet

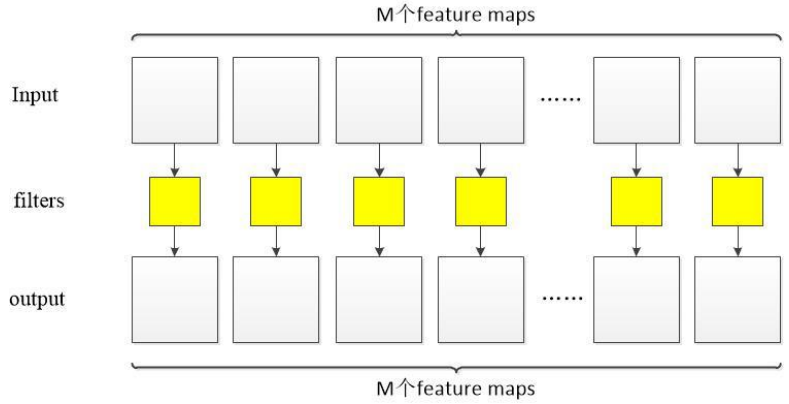


Standard convolutional layer with batchnorm and ReLU

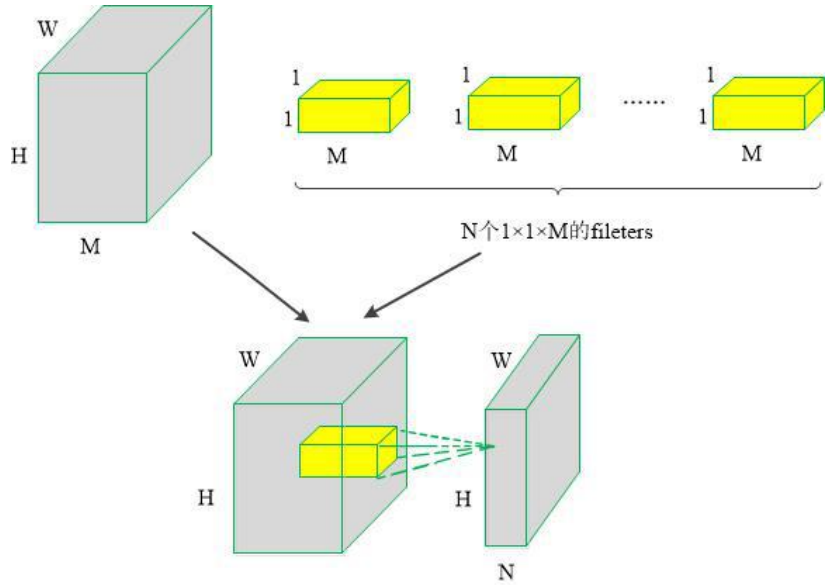


Depthwise Separable convolutions with Depthwise and Pointwise layers followed by batchnorm and ReLU.

7.2：MobileNet – 轻量化（参数少）



Depth wise convolution



Point wise convolution

说明：

1. 输入FM尺寸为：W*H*M，通道数为M，共有M个平面FM
2. 设置M个尺寸为f*f的平面卷积核，分别做平面卷积操作；
3. 输出FM尺寸同样为： W*H*M
4. 与传统卷积的区别是，参数大大减少了（f*f*M VS f*f*M*M）但是不同通道之间没联系了

Table 8. MobileNet Comparison to Popular Models

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
GoogleNet	69.8%	1550	6.8
VGG 16	71.5%	15300	138

说明：

1. 如果想让输出FM尺寸为W*H*N；
2. 则选取N个尺寸为1*1*M的卷积核；
3. 这次操作新增的参数量仅为N*M个。

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Ecient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)

8 ： 尾声&作业说明

课程说明：
本次课程已经定性的将各位听众带到了学术的最前沿。

各位可以直接将课程内容应用于实践

作业说明，本次作业有两版：
1、CPU基础版注重原理验证，无需大量计算资源即可完成，是作业打卡及领取证书的依据

2、GPU进阶版更接近真实情况，但需要大量计算资源支撑方可完成。

大数据

清华大学 Tsinghua University | 数据科学研究院 Institute for Data Science

扫码加好友进群

关注直播间公告