



中国科学院大学
University of Chinese Academy of Sciences

卷积神经网络





神经网络的问题



AI DISCOVERY

导入

➤ 考虑一种情形：

- ✓ 以1000*1000的灰度图像作为输入层
- ✓ 希望隐层有和输入相同的神经单元

➤ 全连接：

- ✓ $10^6 * 10^6 = 1M * 1M = 1T$ 个参数，一个参数以浮点数4Byte的方式存储
- ✓ 需要不小于4T内存

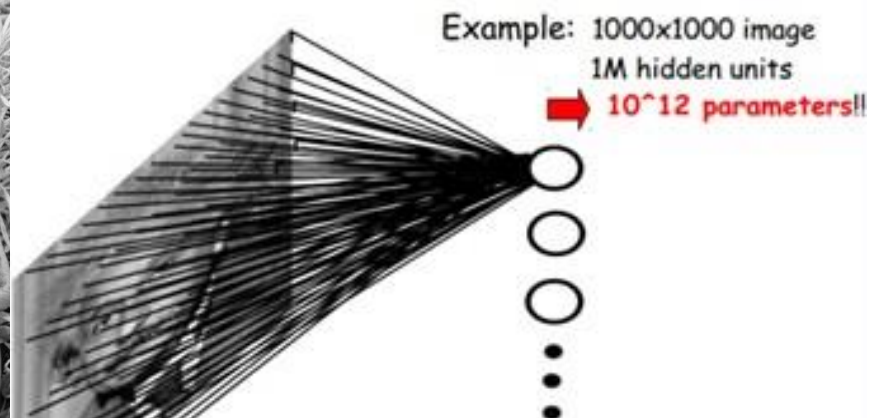
1000-pixels



: 这才是第一个隐层，有什么机器能受得了这种计算？

: 从入门到放弃系列。。

FULLY CONNECTED NEURAL NET



对于图像处理，我们是不是可以利用图像的某些模式或者说是特点，简化每一层的计算过程？



AI DISCOVERY



卷积神经网络的诞生

图像模式的特性——小结

➤ 第一个发现对应的可能的做法：

1. 定义一种提取局部的特征的方法，
可有效响应特定局部模式
2. 用这种方法遍历整张图片

卷积：平移不变模式

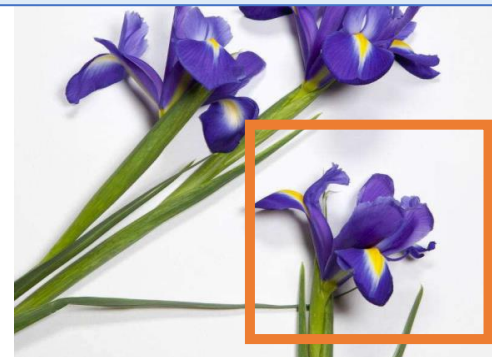
➤ 第二个发现对应的可能的做法：

在神经网络逐层累加的过程中，可以直接对图像进行缩放

池化：下采样被检测物体不变模式

那是不是卷积和池化操作就够了？

我们是不是还需要更深的模型？





卷积层



AI DISCOVERY

卷积的直觉

计算机如何能知道图上有什么物体？

检测图像的边缘

卷积计算 = 特征抽取

灰度图像

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

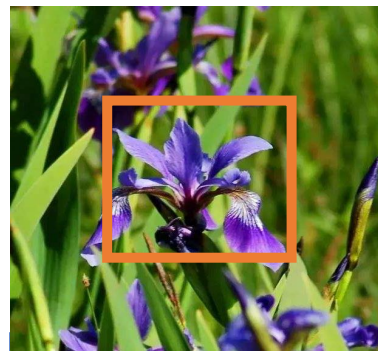
垂直边缘

滤波器

1	0	-1
1	0	-1
1	0	-1

结果图像

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0



卷积：抽取局部模式



vertical edges



AI DISCOVERY



卷积层

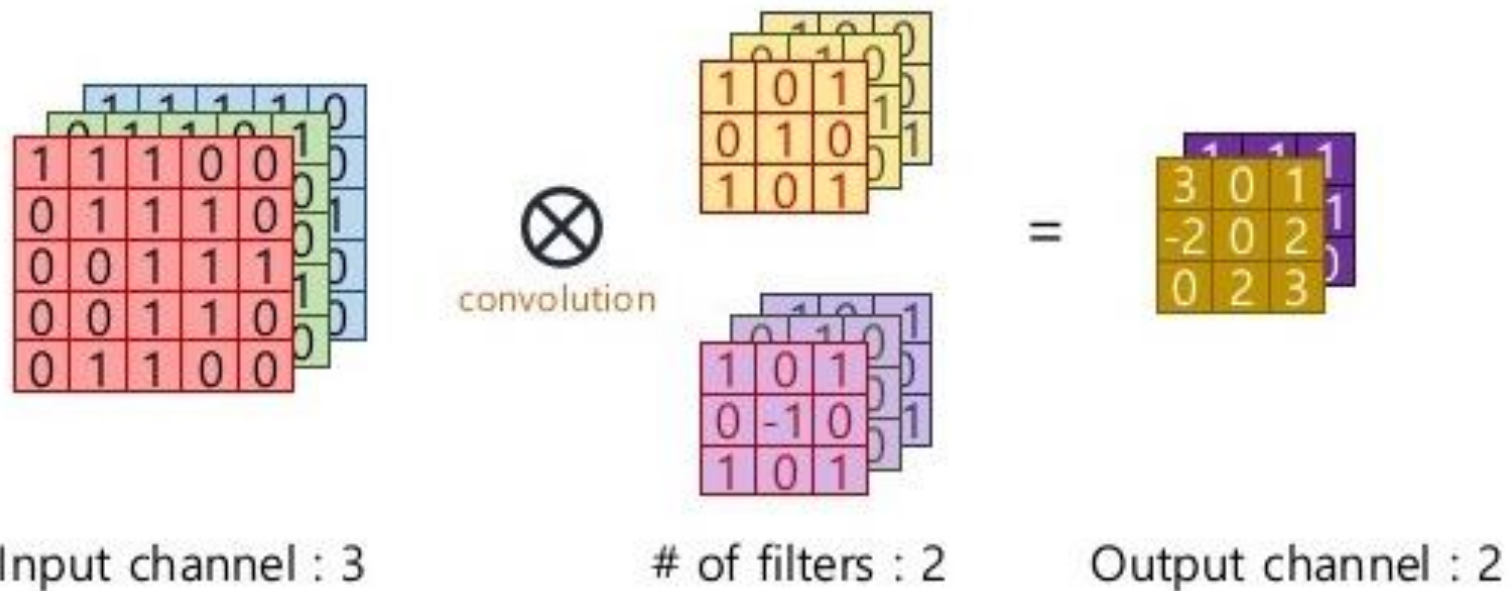


AI DISCOVERY

RGB图像上使用多卷积核：多个不同特征的提取

➤ 多个卷积核

- ✓ 一个卷积核提取一种局部模式
- ✓ 在对每一层的卷积操作中，都要同时使用多个卷积核，提取多种不同的局部模式



AI DISCOVERY



卷积层



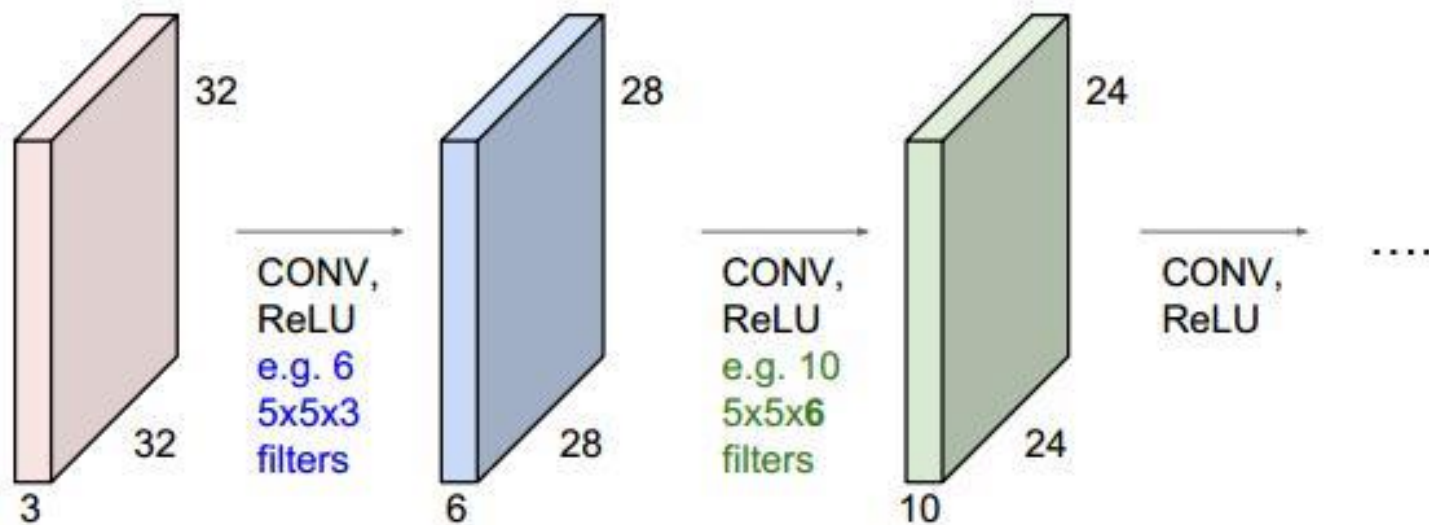
AI DISCOVERY

卷积隐层的堆叠

➤ Notes 2

- ✓ 卷积核的个数 = 下一层数据的深度
度 = 下一卷积层 卷积核的深度
- ✓ 卷积核的个数 = 提取特征的数量，超参数，可以自己调节

Preview: ConvNet is a sequence of Convolutional Layers, interspersed with activation functions



AI DISCOVERY



卷积层

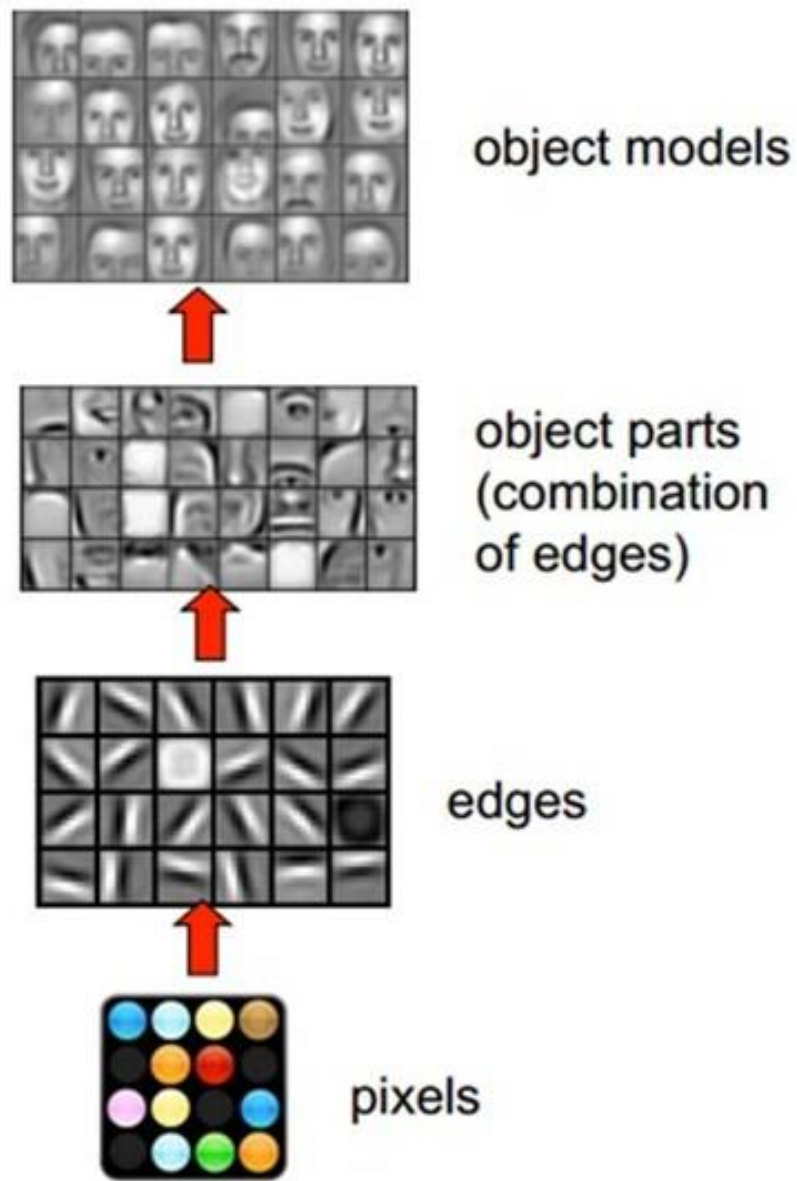


AI DISCOVERY

隐层的卷积：特征组合

✓ 多层卷积：

一层卷积得到的特征往往是局部的
层数越高，学到的特征就越全局化



AI DISCOVERY



卷积层



AI DISCOVERY

需要注意的参数: stride

➤ Stride

- ✓ 一次滑动的步长
- ✓ 有height上的stride和width上的stride
- ✓ 图片中的stride = 2, 指在两个维度上的stride都为2

10	10	10	0	0
10	10	10	0	0
10	10	10	0	0
10	10	10	0	0
10	10	10	0	0

Stride = 1: 一次滑动1格

1	0	-1
1	0	-1
1	0	-1

0	30	30
0	30	30
0	30	30

10	10	10	0	0
10	10	10	0	0
10	10	10	0	0
10	10	10	0	0
10	10	10	0	0

Stride = 2: 一次滑动2格

1	0	-1
1	0	-1
1	0	-1

0	30
0	30



AI DISCOVERY





卷积层

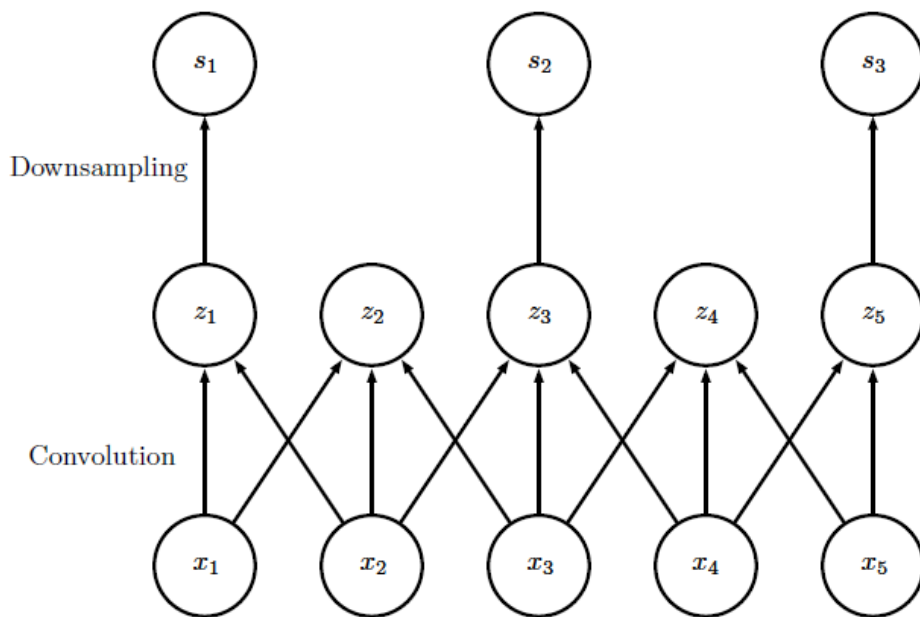
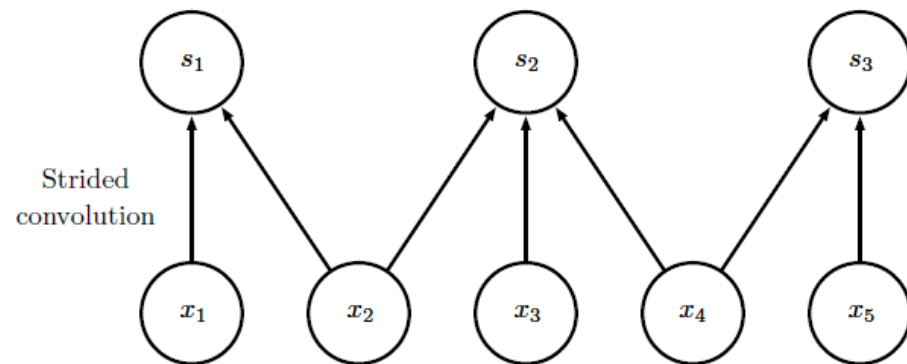


AI DISCOVERY

需要注意的参数: stride

➤ Stride

- ✓ stride 设置为超过1的参数，就相当于在 stride=1 的卷积结果中作了下采样
- ✓ 实际上是跳过去不计算，能够成倍减少计算量





卷积层



AI DISCOVERY

需要注意的参数: padding

➤ Padding = valid

- ✓ 不进行补零操作, $s=1$ 时, 每卷积一次, 宽和高方向的数据维度下降 $F-1$, 其中 F 为卷积核大小

➤ Padding = same

- ✓ 在输入的周围进行0或复制填充
- ✓ 卷积前width=卷积后width, 卷积前height=卷积后height
- ✓ $F=3$, $\text{stride}=1$, $\text{pad}=1$

10	10	10	0	0
10	10	10	0	0
10	10	10	0	0
10	10	10	0	0
10	10	10	0	0

1	0	-1
1	0	-1
1	0	-1

padding: valid

0	30	30
0	30	30
0	30	30

0	0	0	0	0	0	0
0	10	10	10	0	0	0
0	10	10	10	0	0	0
0	10	10	10	0	0	0
0	10	10	10	0	0	0
0	10	10	10	0	0	0
0	0	0	0	0	0	0

1	0	-1
1	0	-1
1	0	-1

padding: same

-20	0	20	20	0
-30	0	30	30	0
-30	0	30	30	0
-30	0	30	30	0
-20	0	20	20	0



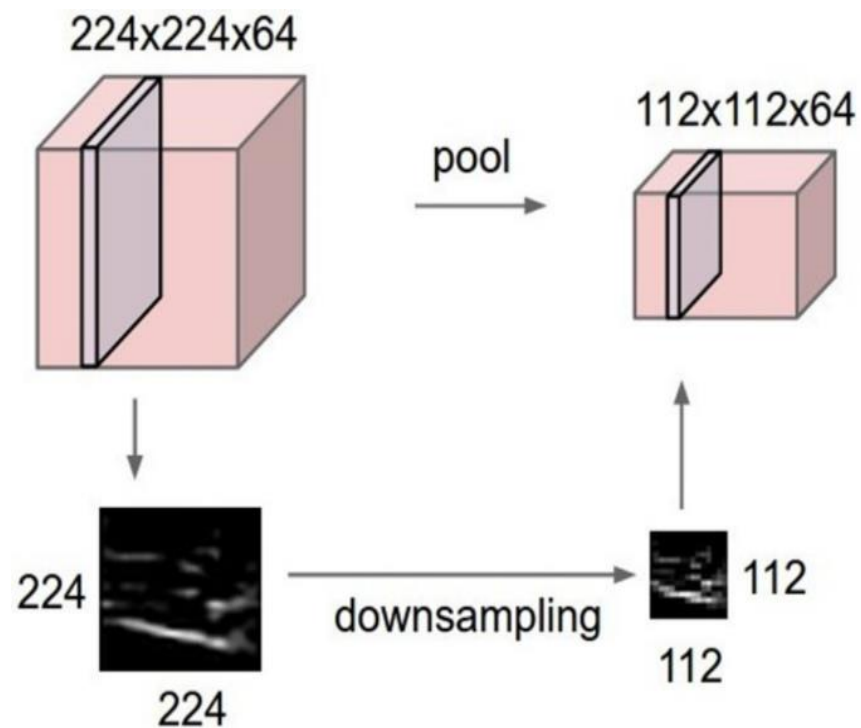
池化层



AI DISCOVERY

➤ 池化

- ✓ 在width和height维度上进行下采样，不改变depth的维度
- ✓ 右图相当于对输入数据使用了 2×2 ， $\text{stride} = 2$ 的卷积核，但是该卷积核不是通过学习获得，而是人为定义的卷积核（不算做参数）
- ✓ 能够成倍减少计算量
- ✓ 相比stride，池化层可以选择进行下采样的方式



AI DISCOVERY

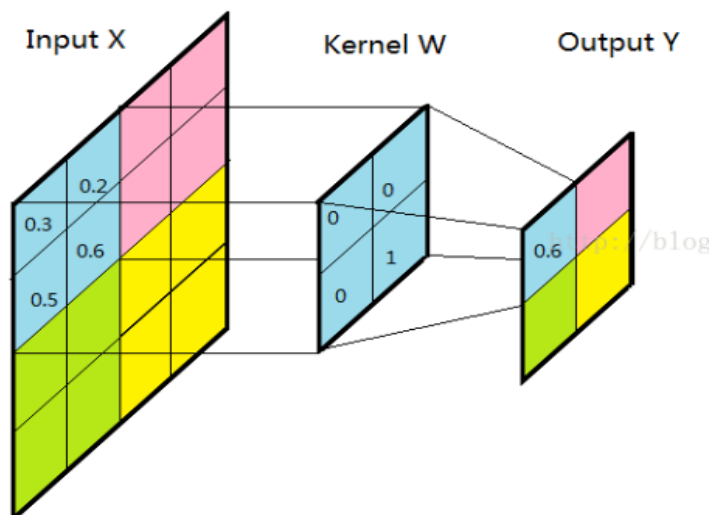


池化层



AI DISCOVERY

max-pooling

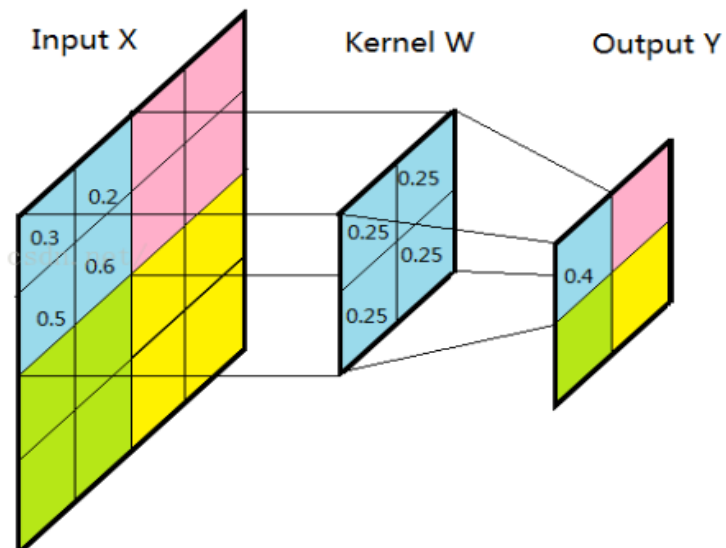


Max-Pooling:

对邻域内特征点取最大作为最后的特征值

$$\max(0.3, 0.2, 0.5, 0.6) = 0.6$$

mean-polling



Mean-Pooling:

对邻域内特征点取平均作为最后的特征值

$$\frac{0.3 + 0.2 + 0.5 + 0.6}{4} = 0.4$$





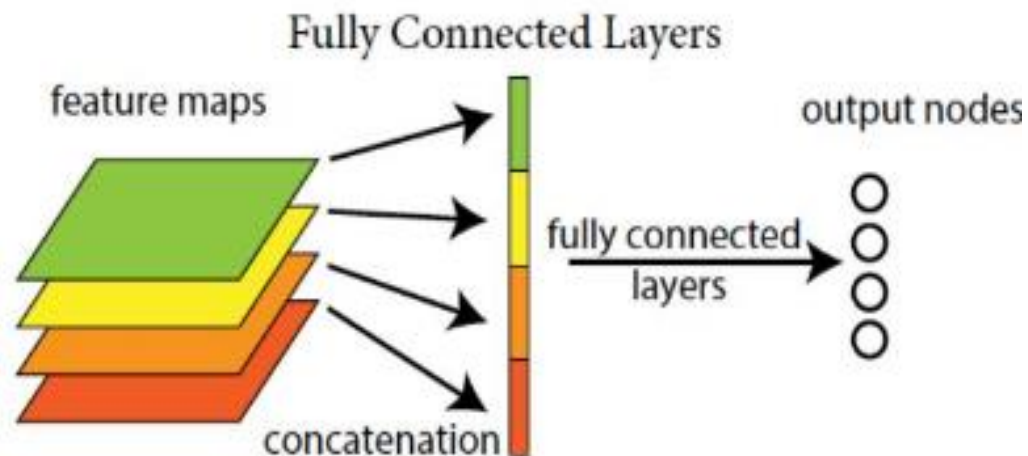
全连接层分类



AI DISCOVERY

➤ Notes

- ✓ 将多层的特征映射抻直成一个一维的向量
- ✓ 采用全连接的方式将向量连接向输出层，
打破卷积特征的空间限制
对卷积层获得的不同的特征进行加权
最终目的是得到一个可以对不同类别进行区分的得分
- ✓ 输出层就是对应每个类别的得分



AI DISCOVERY



网络搭建小结

AI DISCOVERY

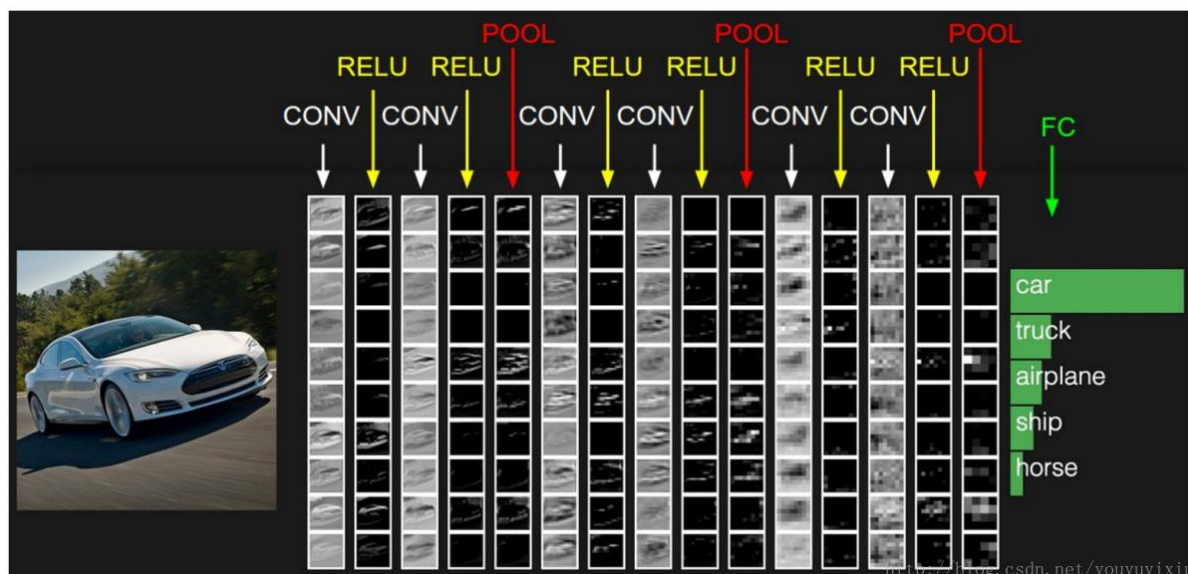


卷积神经网络的一般结构

1. 卷积层+ReLU和 池化层 的组合多次出现
2. 多个全连接 或 特殊的CNN结构 作为输出层

提取特征

作分类器/检测器/分割器



AI DISCOVERY



模型的泛化

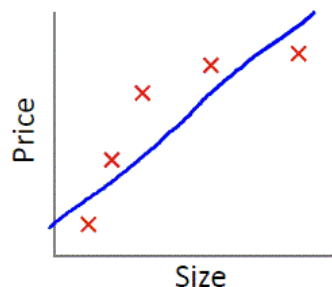


AI DISCOVERY

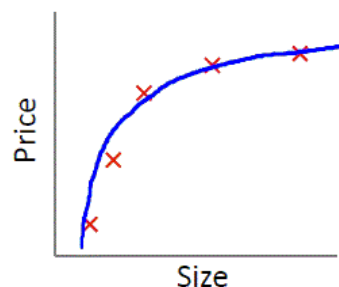
◆ 深度神经网络的泛化能力

1) **高模型容量**——高拟合各种函数的能力，模型偏向于**过拟合**

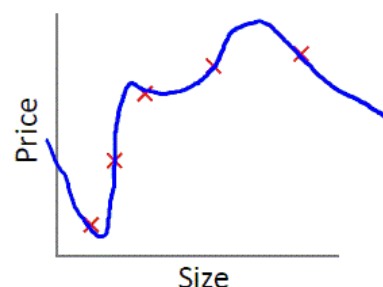
2) **正则化**——对学习算法的修改，为了减少测试误差（泛化误差）而不是训练误差



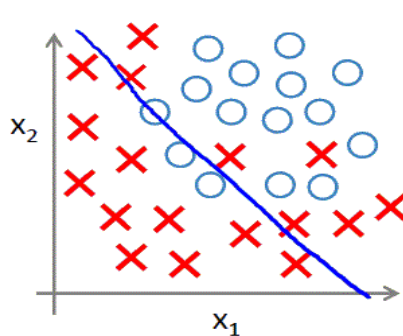
$$\theta_0 + \theta_1 x$$



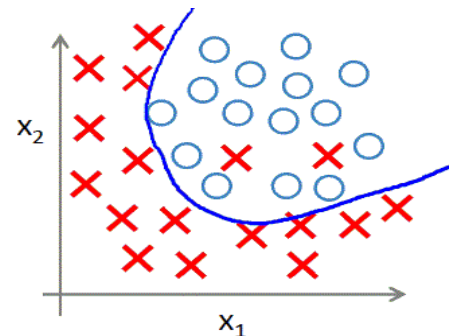
$$\theta_0 + \theta_1 x + \theta_2 x^2$$



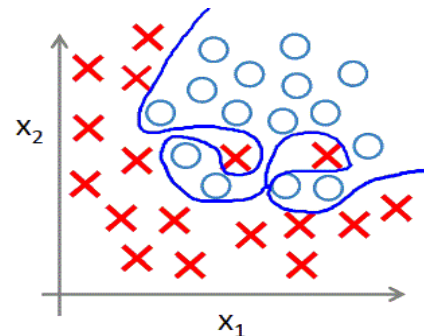
$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$



欠拟合



正确拟合



过拟合



AI DISCOVERY





MNIST



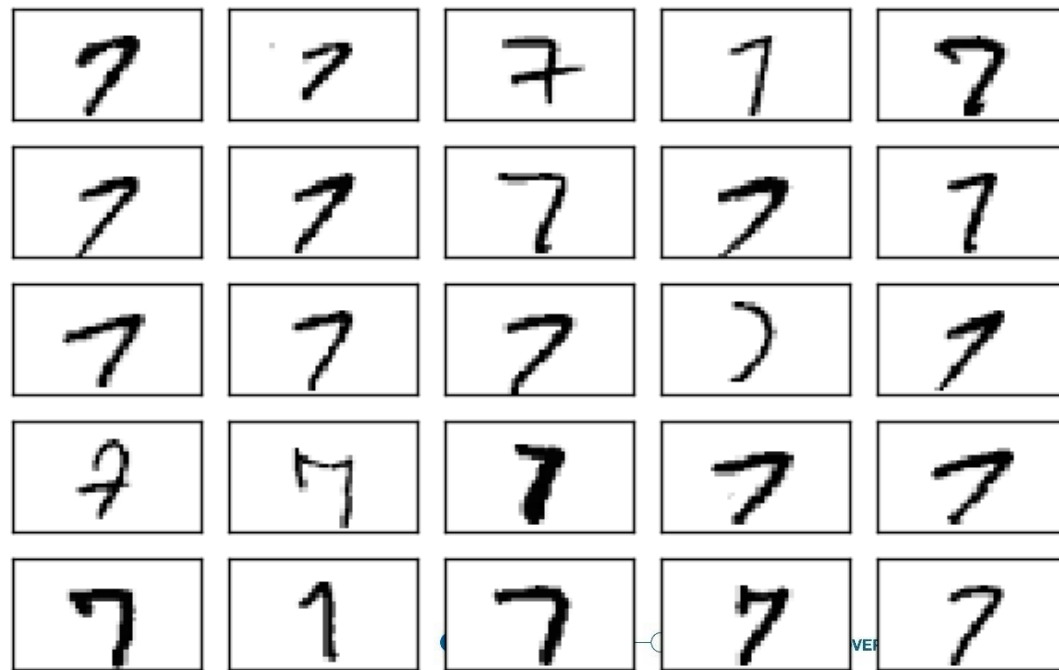
AI DISCOVERY

<http://yann.lecun.com/exdb/mnist/>

- ✓ 手写数字
- ✓ 10个互斥的类别，28*28的灰度图像
- ✓ 美国国家标准与技术研究所, National Institute of Standards and Technology (NIST)
- ✓ 250个人的手写数字。50%来自人口普查局工作人员，50%来自高中学生。
- ✓ 训练：60,000个样本
- ✓ 测试：10,000个样本

Four files are available on this site:

[train-images-idx3-ubyte.gz](#): training set images (9912422 bytes)
[train-labels-idx1-ubyte.gz](#): training set labels (28881 bytes)
[t10k-images-idx3-ubyte.gz](#): test set images (1648877 bytes)
[t10k-labels-idx1-ubyte.gz](#): test set labels (4542 bytes)





CIFAR



AI DISCOVERY

<http://www.cs.toronto.edu/~kriz/cifar.html>

- ✓ 数据集的类别分布见右图
- ✓ 60000张32*32 的RGB图像
- ✓ 10个互斥的类别，每类6000张图片
- ✓ 训练：50000张，
5个训练批，每批10000张
每类图像随机抽取，张数不平均
- ✓ 测试：10000张，
单独构成一批，每类1000张

Download

If you're going to use this dataset, please cite the tech report at the bottom of this page.

Version	Size	md5sum
CIFAR-10 python version	163 MB	c58f30108f718f92721af3b95e74349a
CIFAR-10 Matlab version	175 MB	70270af85842c9e89bb428ec9976c926
CIFAR-10 binary version (suitable for C programs)	162 MB	c32a1d4ab5d03f1284b67883e8d87530

airplane



automobile



bird



cat



deer



dog



frog



horse



ship



truck





ImageNet



AI DISCOVERY

<http://image-net.org/>

- 1) Total number of non-empty synsets: 21841
 - 2) Total number of images: 14,197,122
 - 3) Number of images with bounding box annotations: 1,034,908
 - 4) Number of synsets with SIFT features: 1000
 - 5) Number of images with SIFT features: 1.2 million
- ✓ 可供用于图像分类、目标定位、目标检测、实例分割等多个计算机视觉任务。
 - ✓ ImageNet国际计算机视觉挑战赛(ILSVRC)

Computer Vision Tasks

Classification



CAT

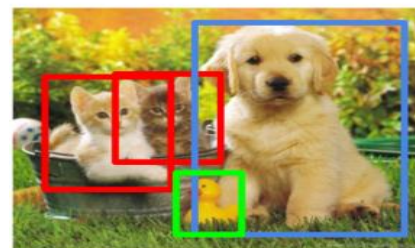
Classification + Localization



CAT

Single object

Object Detection



CAT, DOG, DUCK

Instance Segmentation



CAT, DOG, DUCK

Multiple objects

http://blog.csdn.net/xingwei_09



经典CNN



AI DISCOVERY

早期尝试

Hubel & Wiesel

使用MNIST数据集，
这是最早用于数字识别的CNN

LeNet

Dropout
ReLU

历史突破

AlexNet

2012 ILSVRC 远超第2名

ZF Net, 基于AlexNet, 2013 ILSVRC 冠军

发展和演化

网络加深

VGG

2014 ILSVRC, 图像识别略差于
GoogLeNet, 但是在很多图像分析
问题(比如object detection)上效果好

增强卷积模块功能

NIN

GoogLeNet

Inception V3,V4

增加新的功能单元

Inception V2, BN

2014 ILSVRC 冠军

融合

ResNet

2015年ILSVRC,

Classification 获得第一名



AI DISCOVERY



VGG



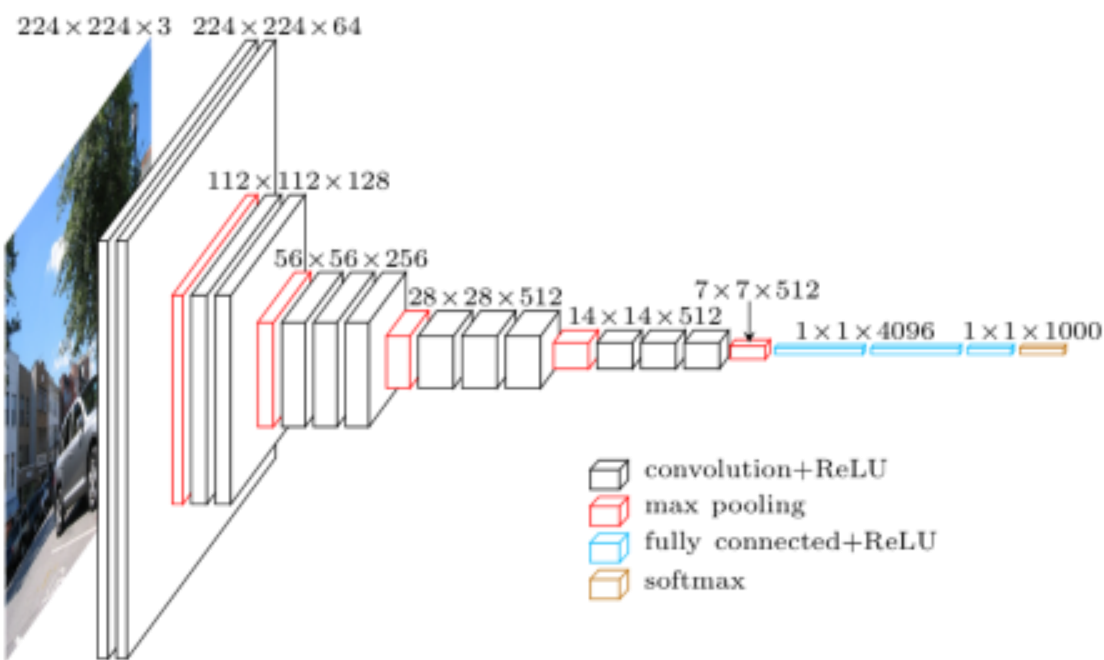
AI DISCOVERY

牛津大学计算机视觉组

深度增加 & 小卷积核 → 对网络最后的分类识别效果有很大作用

小卷积核: 3×3 : 表示上下、左右、中心这些概念的最小卷积核尺寸

深度: AlexNet 8 层 → VGG 最深19层



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					



GoogLeNet/Inception

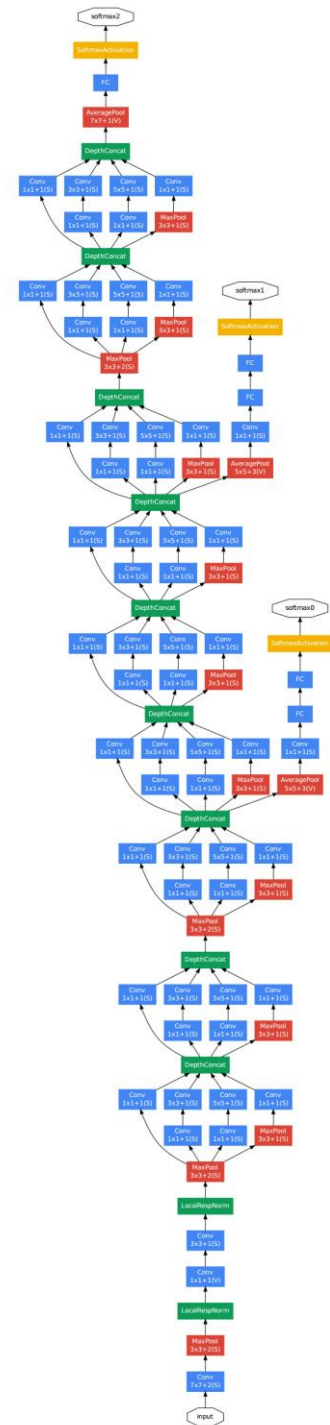
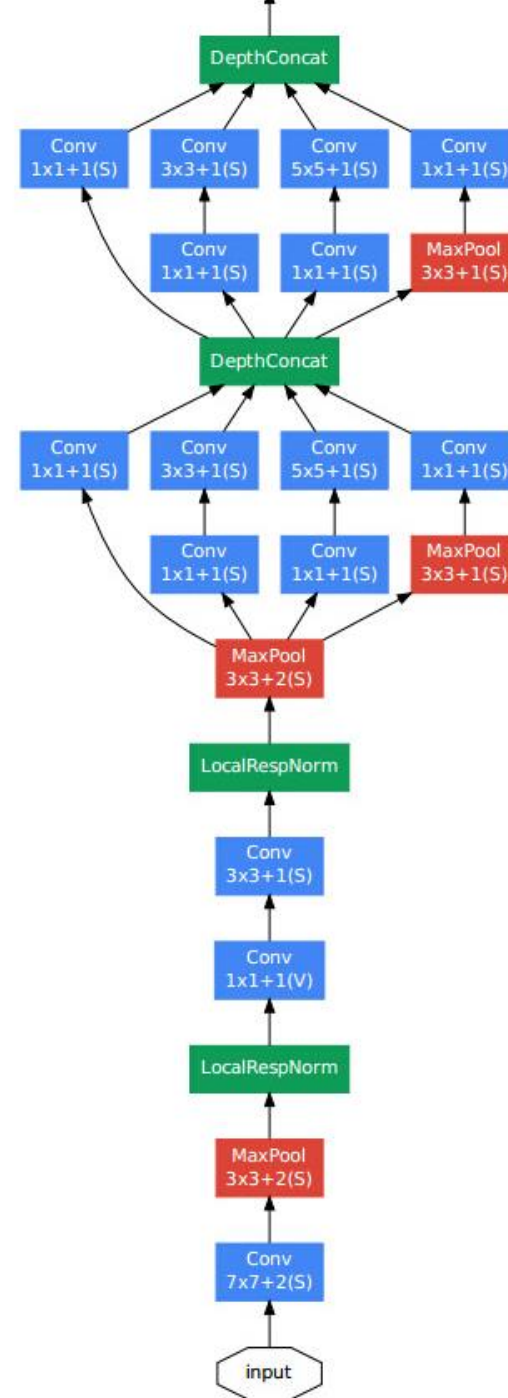
Google提出的神经网络，一共有V1-V4共4个不同的模型

✓ 使用了Inception模块

- 可以并行执行多个具有不同尺度的卷积运算或池化操作
- 将多个卷积核卷积的结果拼接成一个非常深的特征图

✓ 使用了大量的trick提高网络性能

- Bottleneck（瓶颈）：1*1的卷积核，借鉴NIN
- 使用全局平均池化GAP代替全连接
- 在v2中，采用Batch Normalization（批归一化）
- 在v3中，采用非对称卷积降低运算量
- 在v4中，结合了ResNet中的思想，发现Residual Connections貌似只能加快网络收敛速度，是更大的网络规模提高了精度





ResNet



AI DISCOVERY

微软提出的神经网络，CVPR当年的最佳论文

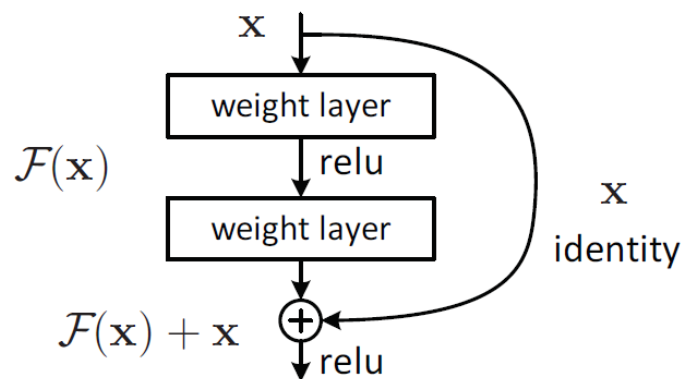
✓ 使用了恒等映射

传统神经网络训练的函数为 $F(x)$

添加恒等映射后，神经网络训练的函数变为 $F(x)+x$

作者认为这样训练出来的网络，相当于是在对 x 作修正，修正的幅度就是 $F(x)$ ， $F(x)$ 在数学上称为残差

所以作者提出的网络称为**残差网络**



method	top-1 err.	top-5 err.
VGG [41] (ILSVRC'14)	-	8.43 [†]
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	19.38	4.49

ResNets @ ILSVRC & COCO 2015 Competitions

• 1st places in all five main tracks

- ImageNet Classification: "Ultra-deep" 152-layer nets
- ImageNet Detection: 16% better than 2nd
- ImageNet Localization: 27% better than 2nd
- COCO Detection: 11% better than 2nd
- COCO Segmentation: 12% better than 2nd



ResNet

AI DISCOVERY

➤ 一点直觉：

平凡网络中，为什么深度增加到一定程度后，训练的效果反而下降了？

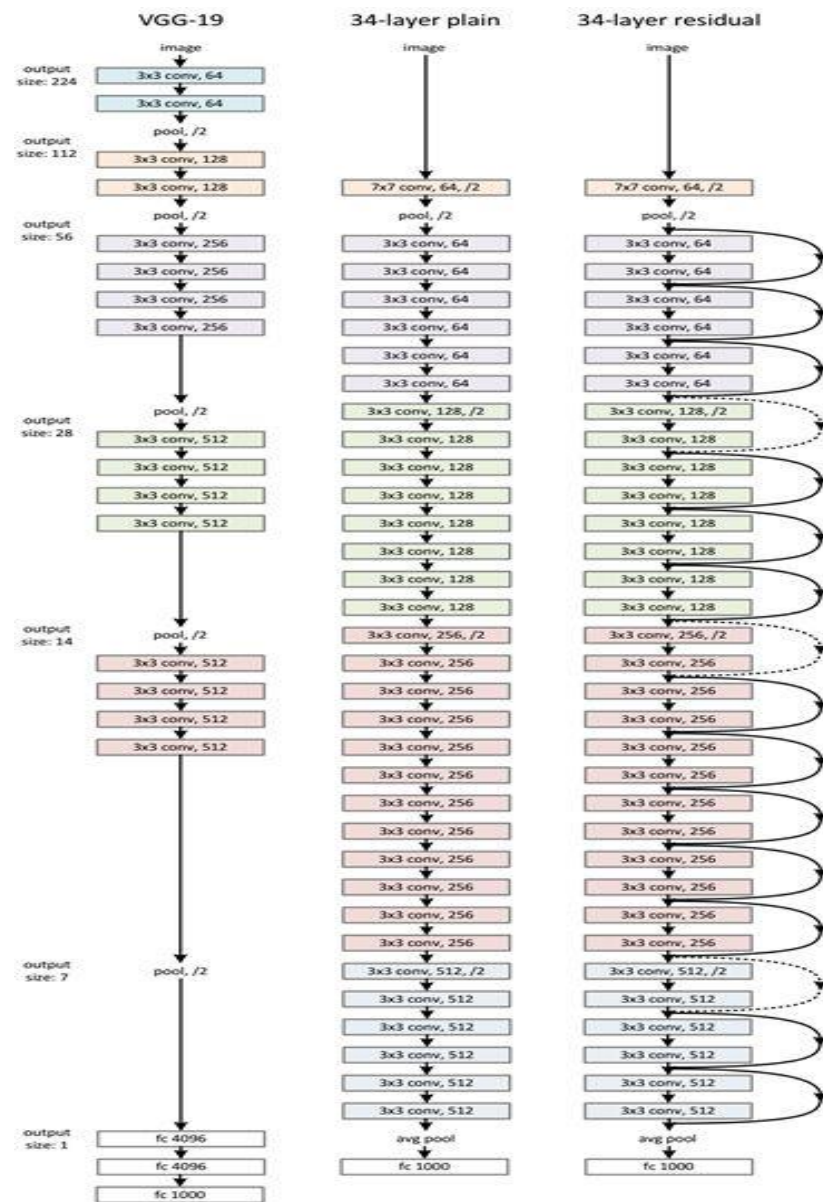
梯度下降算法本身的缺陷

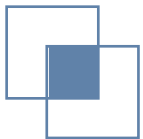
深度增加后，相比浅一些的网络来说梯度减小。误差传播过程变慢，网络的优化速度就慢

残差网络中，为什么能够避免这种现象？

恒等映射下，虽然网络的深度加深，但是每层中都会有足够多的由梯度承载的信息量，梯度不会太小

加快了深层网络的收敛速度





小结

如何提升一个神经网络的性能?

- Aspect 1: 修改网络: 增加深度, 增加宽度
 减少参数量, 防止过拟合, 解决梯度消失问题
- Aspect 2: 数据集: 尽可能多的数据防止过拟合

✓ AlexNet:

Group Convolution(分组卷积), Dropout, Data Augmentation (数据增强)

✓ VGG:

深度, 小卷积核

✓ Inception (Google-Net) :

同一层使用多个类型的卷积核, Bottleneck, Batch Normalization (批归一化)

✓ ResNet:

skip/identity connection 残差的引入