CAMALAB
计算机动画与多媒体分析实验室

现代CNN模型结构演化

余 宙
yuz@hdu.edu.cn
2017.8.31

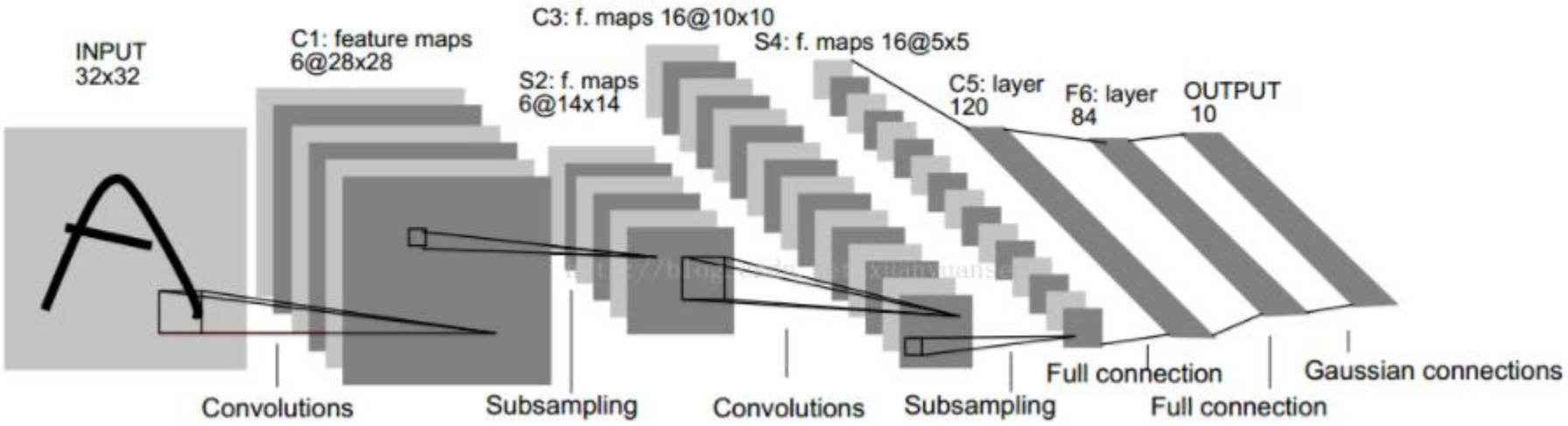
提纲

- Part I. CNN基本结构回顾
- Part II. 从AlexNet到ResNet, 现代CNN结构发展史
- Part III. CNN发展新方向

PART I. CNN基本结构回顾

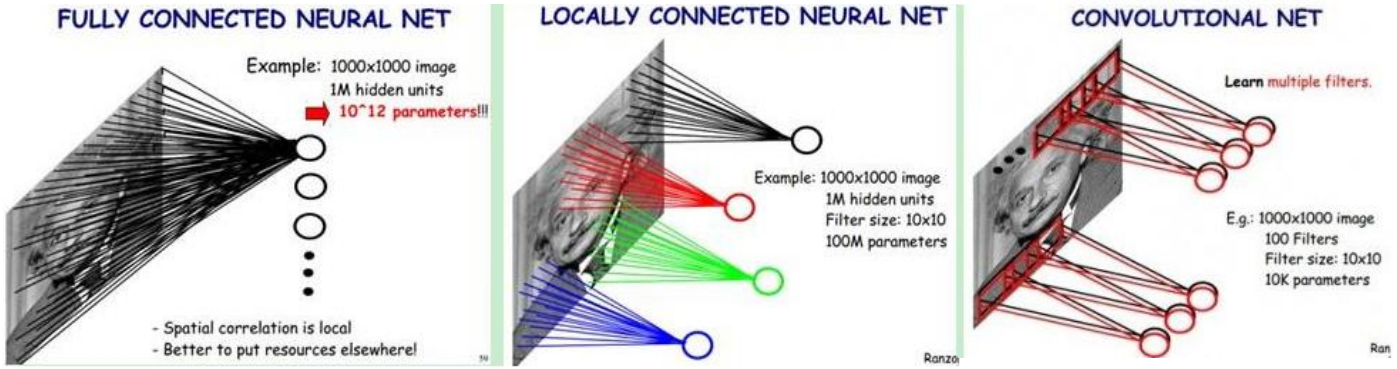
CNN模型回顾

- LeNet5

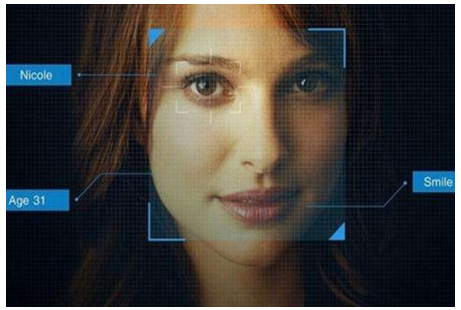


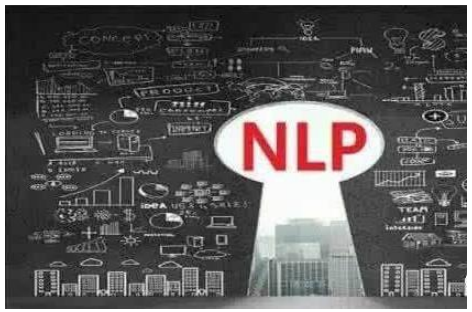
CNN vs. DNN

- CNN相比于DNN优势
  - 更少的参数



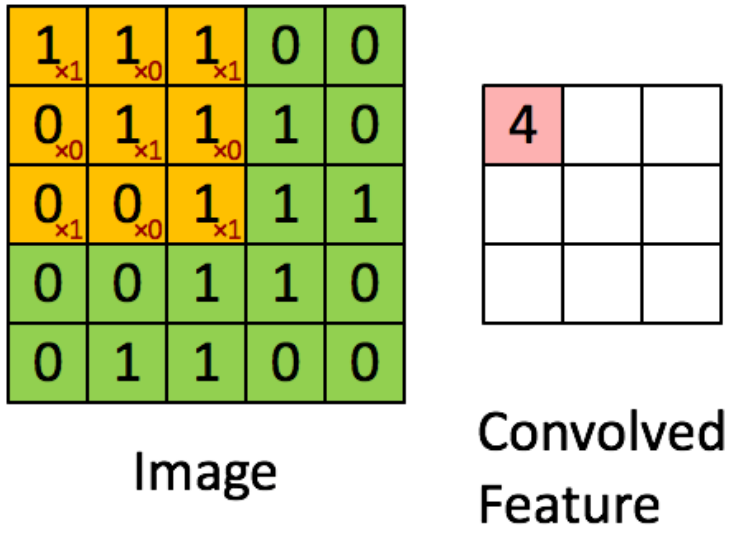
  - 局部感受野与局部相关性 (**important!**)





CNN基本单元

- 卷积层(Convolutional Layer)
  - Kernel-size(ks)

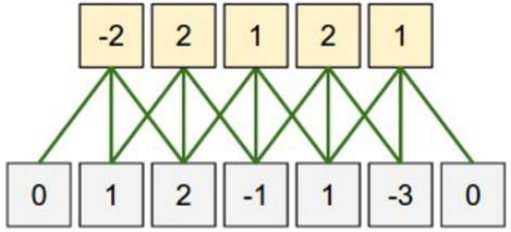


输出特征图尺寸计算方式:

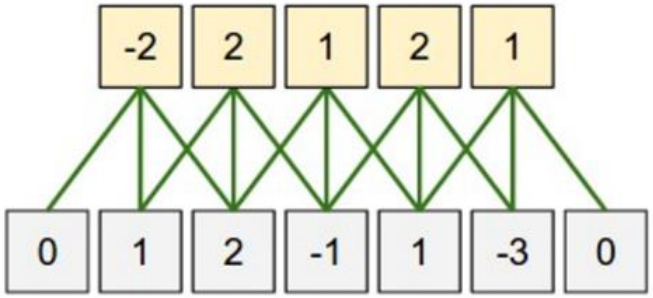
$$x_{out} = x_{in} - ks + 1$$

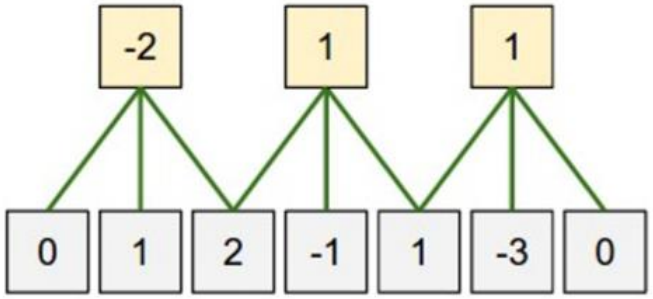
CNN基本单元

- 卷积层(Convolutional Layer)
  - Zero-padding



  - Stride



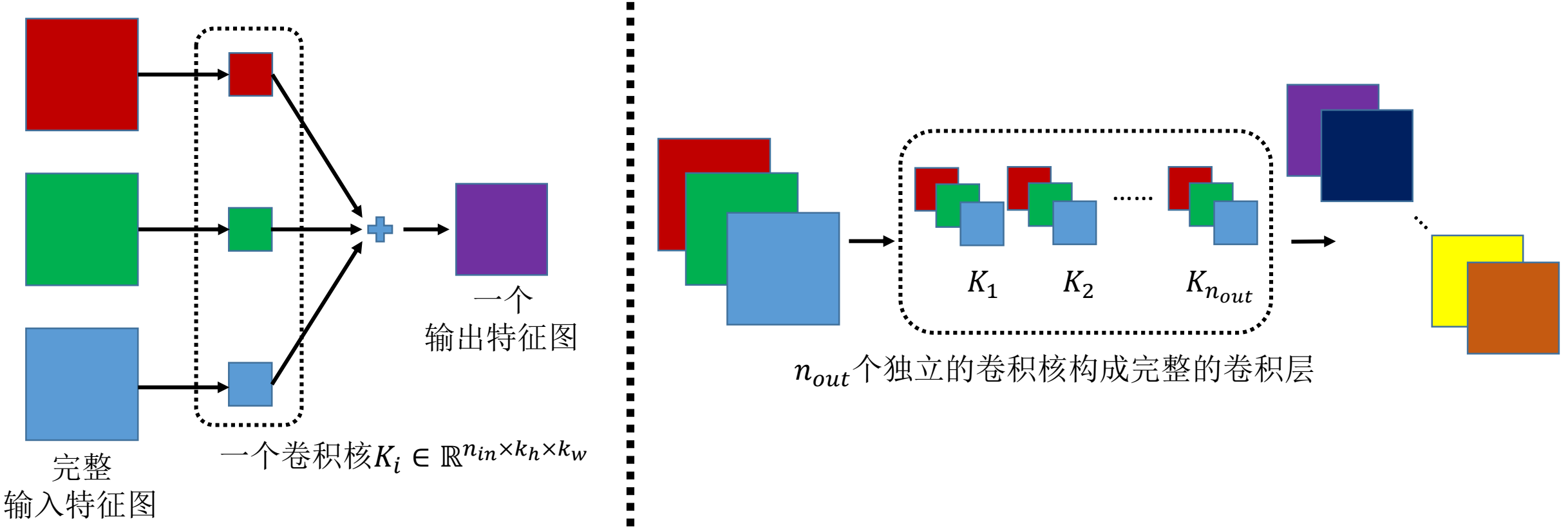


输出特征图尺寸完整计算方式:

$$x_{out} = \left\lfloor \frac{x_{in} - ks + 2 * pad}{stride} \right\rfloor + 1$$

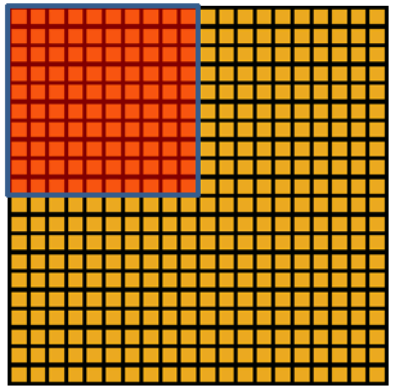
CNN基本单元

- 卷积层(Convolutional Layer)
  - 一个卷积层参数$K \in \mathbb{R}^{n_{in} \times n_{out} \times k_h \times k_w}$

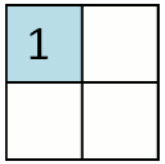


CNN基本单元

- 池化层(Pooling Layer)



Convolved feature



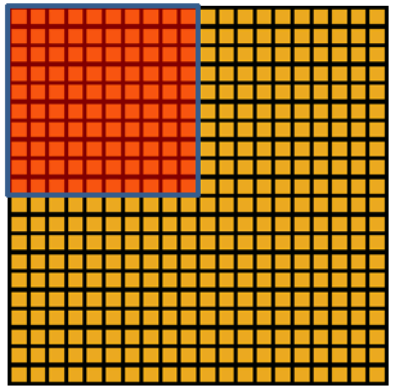
Pooled feature

输出特征图尺寸计算方式:

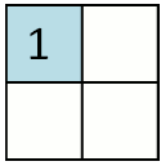
$$x_{out} = \left\lceil \frac{x_{in} - ks + 2 * pad}{stride} \right\rceil + 1$$

CNN基本单元

- 池化层(Pooling Layer)



Convolved feature



Pooled feature

输出特征图尺寸计算方式:

$$x_{out} = \left\lceil \frac{x_{in} - ks + 2 * pad}{stride} \right\rceil + 1$$

Q1：conv和pooling层在计算输出尺寸时为什么一个取上界一个取下届？

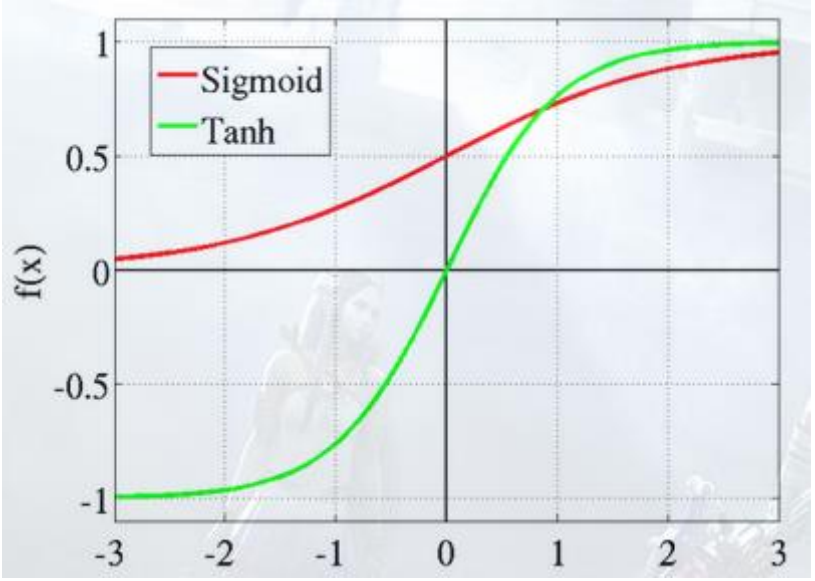
CNN基本单元

- 非线性激活层(Activation Layer)
  - Sigmoid激活函数

$$f(x) = \frac{1}{1+e^{-x}}$$

  - Tanh激活函数

$$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$$



Sigmoid和Tanh激活函数

CNN基本单元

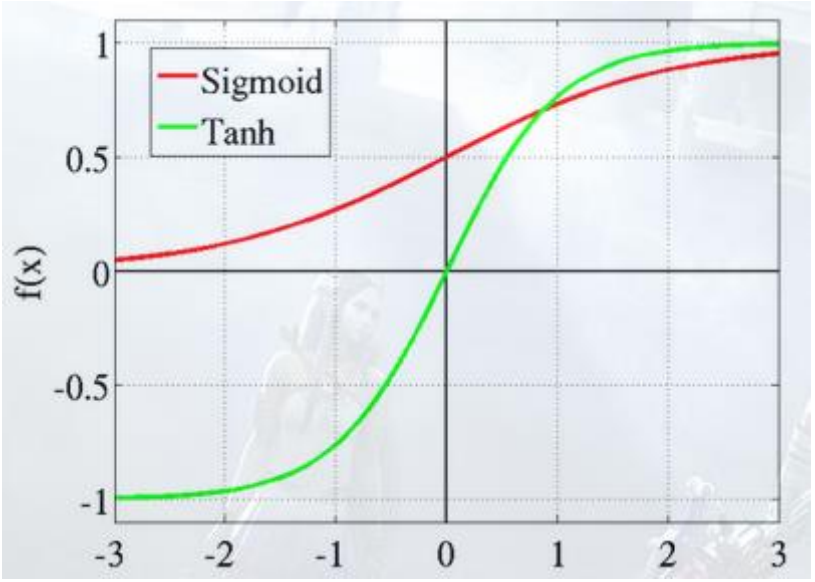
- 非线性激活层(Activation Layer)
  - Sigmoid激活函数

$$f(x) = \frac{1}{1+e^{-x}}$$

  - Tanh激活函数

$$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$$

Q2：sigmoid和tanh函数之间在数学上有什么关联？

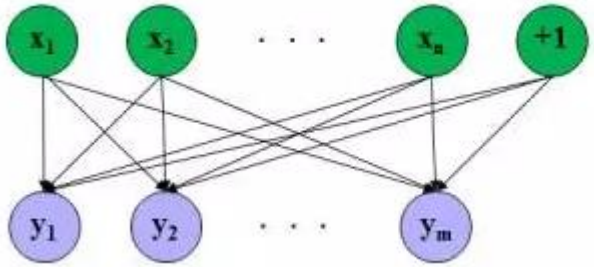


Sigmoid和Tanh激活函数

CNN基本单元

- 全连接层(Fully-connected layer)

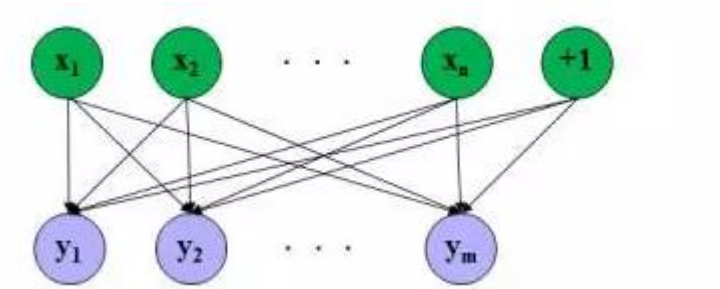
$$y = W^T x + b,\ W \in \mathbb{R}^{d_{out} \times d_{in}}, b \in \mathbb{R}^{d_{out}}$$



CNN基本单元

- 全连接层(Fully-connected layer)

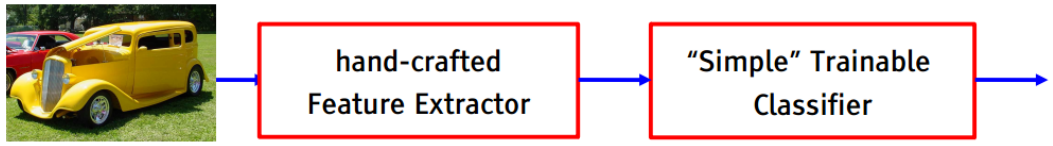
$$y = W^T x + b,\ W \in \mathbb{R}^{d_{out} \times d_{in}}, b \in \mathbb{R}^{d_{out}}$$



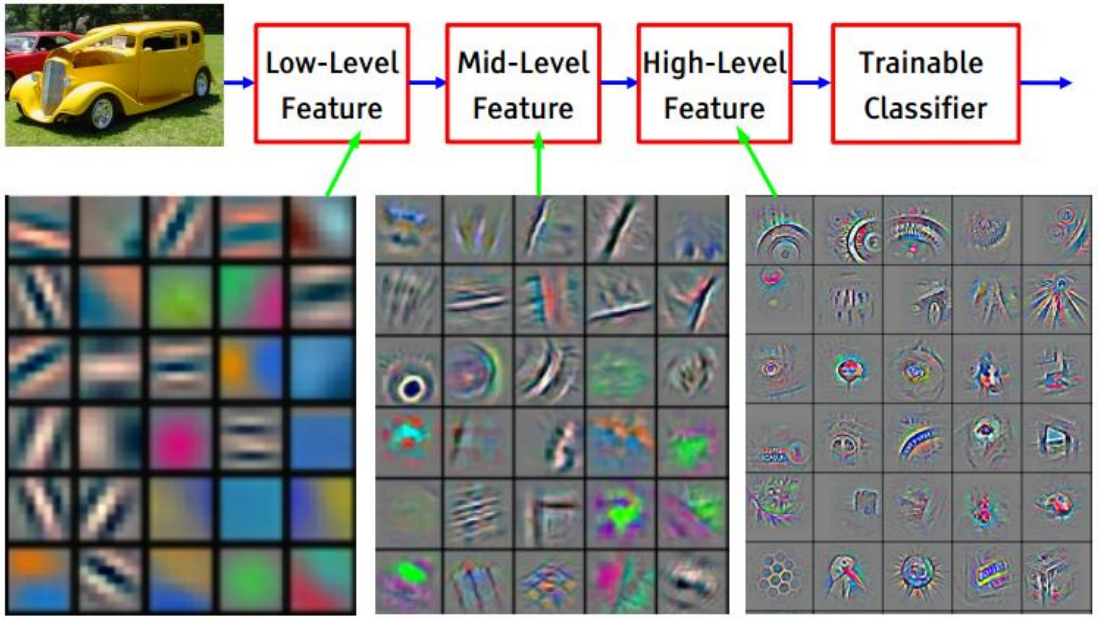
Q3: 卷积和全连接 层在形式上有什么关联？
Q4: 两者可否复用一套代码实现?

理解CNN

- CNN以原始图像的2D原始像素作为输入，使用**多层的**可学习的卷积层和池化层逐层对图像学习复杂的非线性变换，基于输出层定义的损失函数使用反向传播算法**端到端(End-to-end)**学习，从而自动学习得到图像**底层到高层的层次化语义表达**



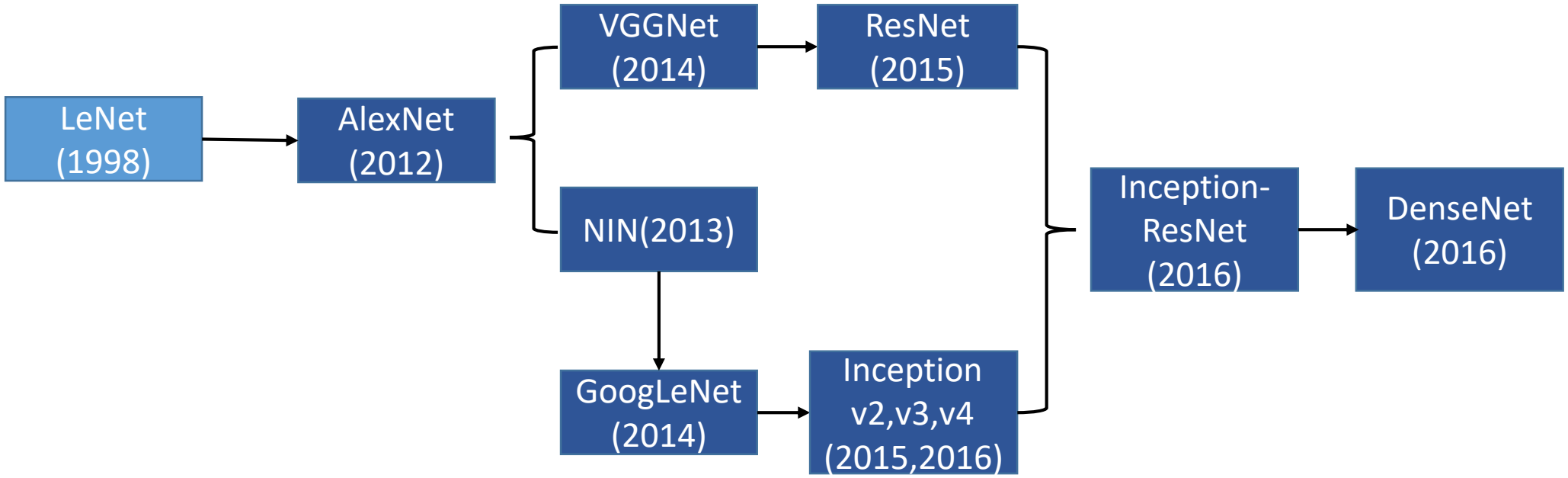
传统方法



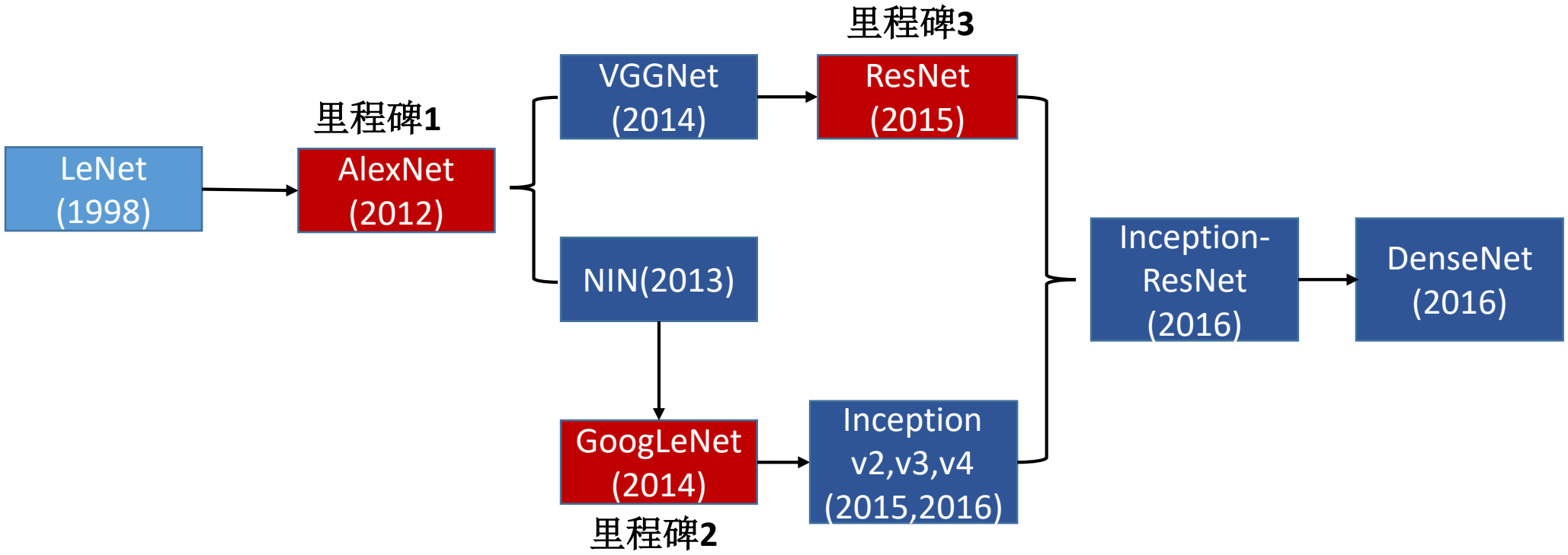
CNN

PART II. 现代CNN结构发展史

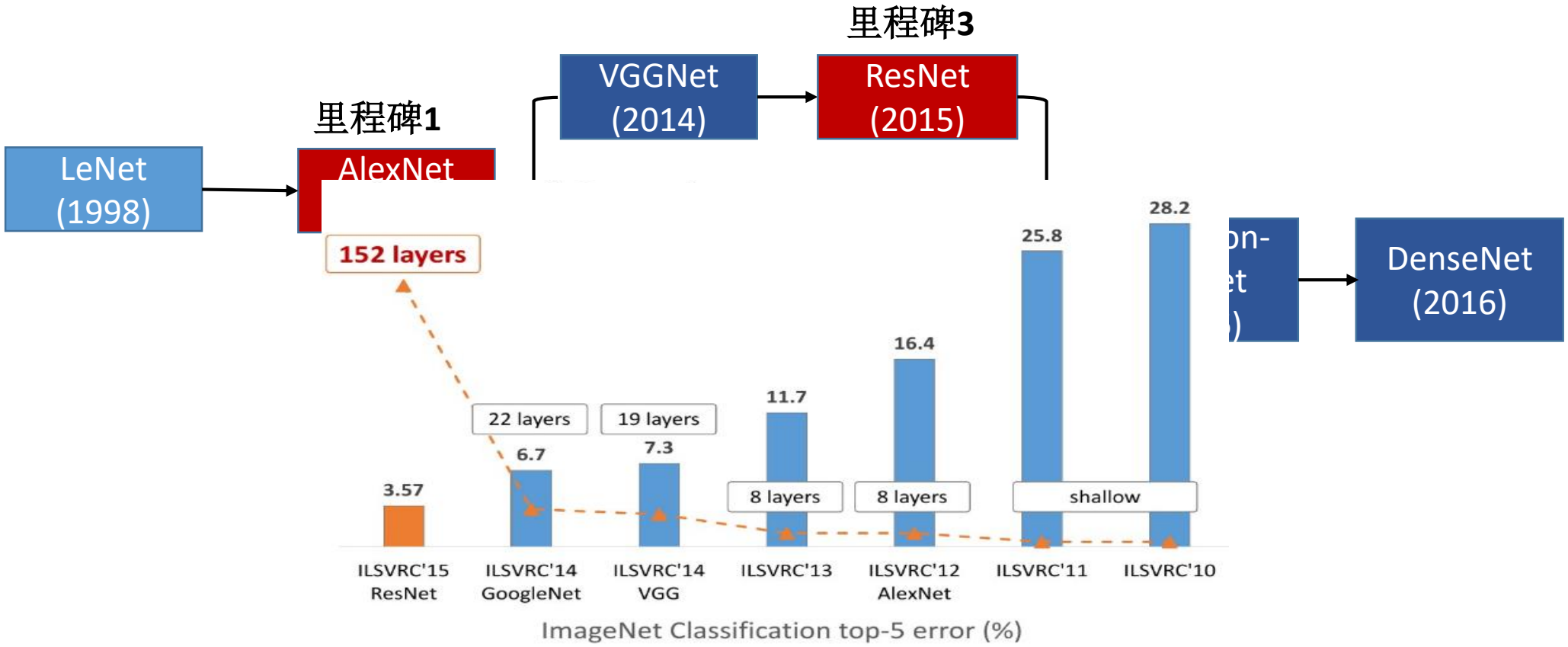
现代CNN模型结构概述



现代CNN模型结构概述

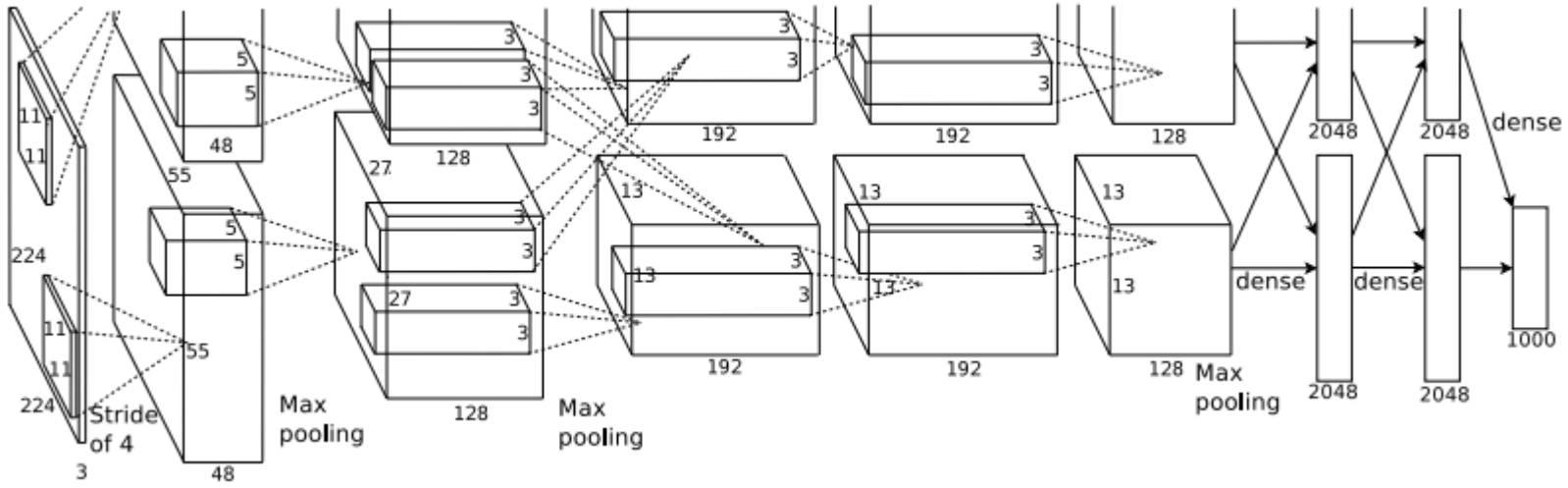


现代CNN模型结构概述



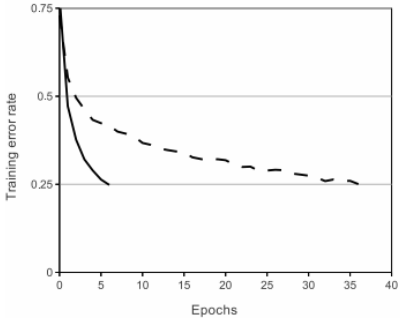
AlexNet(2012)[1]

- 使用百万规模完全标准的图像数据集ImageNet训练
- 设计大规模CNN网络从原始图像训练端到端分类模型
- 使用GPU大大加速训练
- ImageNet数据集上top-5正确率比之前最好方法提升9%

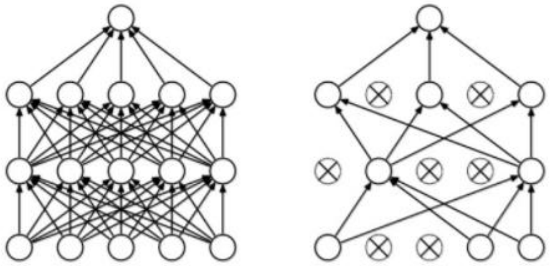


AlexNet(2012)

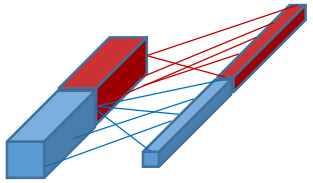
- 关键技术
  - 使用ReLU激活函数代替传统sigmoid和tanh，成为CNN中标准模块
    - 相比tanh，收敛速度提升6倍



  - 引入Dropout层防止过拟合
    - 在每个mini-batch的forward-backward过程中，将全连接层的输出特征以一定概率将特征随机置为0



  - 引入Group convolution减少计算量和显存占用
    - 卷积层之间输入和输出通道之间局部连接，



NIN(2013) [2]

- 提出一种Network in Network的思想，使用非线性的MLP卷积代替传统卷积
  - 实现中使用1x1 conv + ReLU+conv来实现MLPconv。1x1 conv后来成为CNN中标准模块
- 提出使用global average pooling代替fc层，大大减少参数量
  - AlexNet中全连接层占总参数量95%(68million)

VGGNet(2014) [3]

- ImageNet竞赛 2014年亚军方案
- 使用连续的3x3小卷积代替AlexNet中7x7的大卷积，控制参数量的同时网络加深。3x3卷积成为后来CNN结构中的标配
- 设计16和19层的VGG-16和VGG-19网络，成为后来的各种计算机视觉任务的基准模型

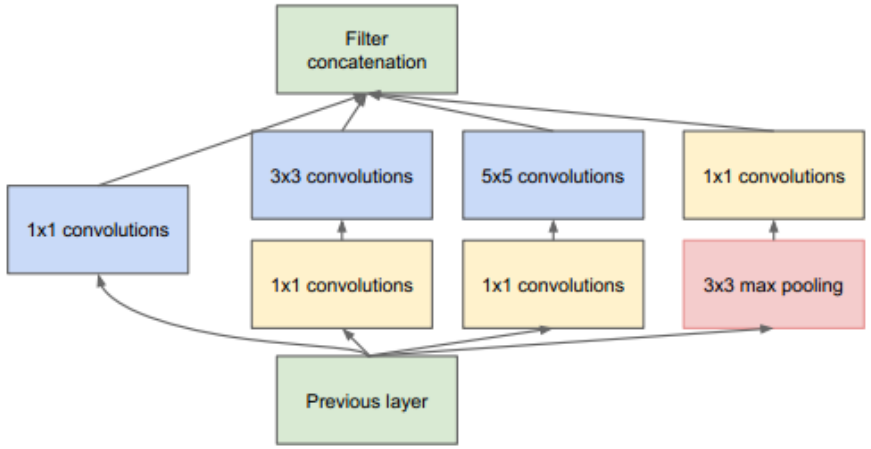
VGGNet(2014) [3]

- ImageNet竞赛 2014年亚军方案
- 使用连续的3x3小卷积代替AlexNe
  同时网络加深、3x3卷积成为后来
- 设计16和19层的VGG-16和VGG-19
  觉任务的基准模型

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(2014) [4]

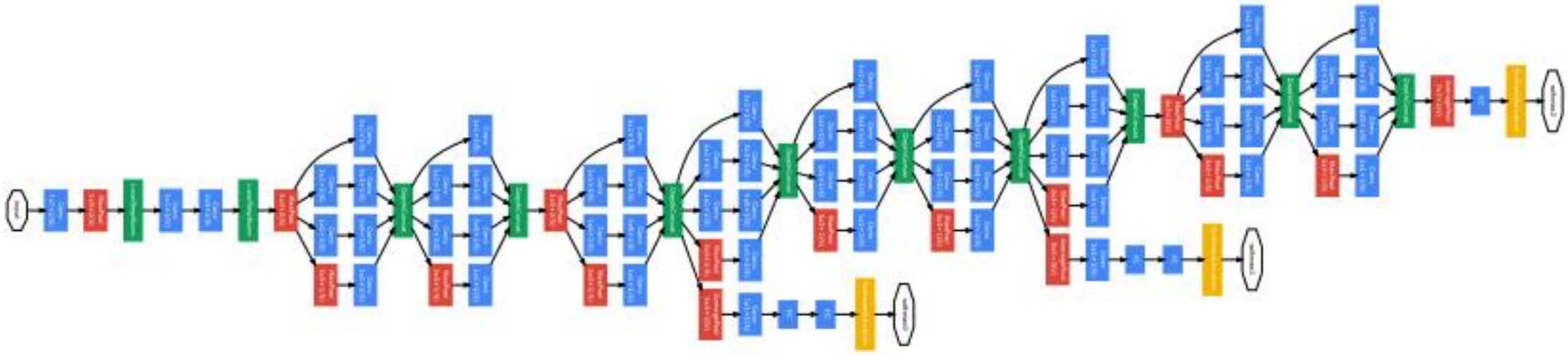
- ImageNet竞赛 2014年冠军方案，来自google团队，致敬LeNet
- 23层网络，深度上超过之前所有模型，单参数相比VGG模型小很多(5million vs. 144 million)
- Inception模块化设计，不同尺度特征图串联，形成更强的表达能力
  - 引入NIN 中1x1 conv，用来降维，降低计算量



(b) Inception module with dimension reductions

GoogLeNet(2014) [4]

- ImageNet竞赛 2014年冠军方案，来自google团队，致敬LeNet
- 22层网络，深度上超过之前所有模型，总参数相比VGG模型小很



GoogLeNet Family

- GoogLeNet -> Inception v1
- Inception v2[5]：在v1基础上引入Batch Normalization(BN)层，加在所有卷积层后面，使得深度神经网络训练时梯度消失问题得到缓解，大大提升收敛速度。BN成为CNN网络中标准模块。

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

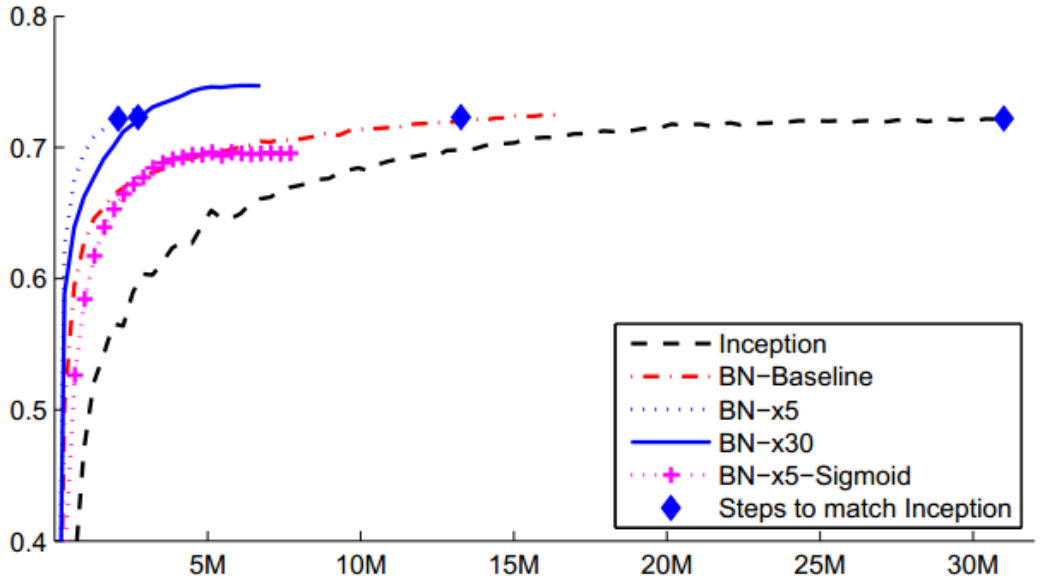
$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

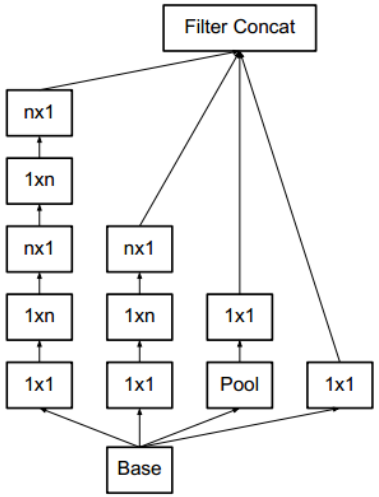
$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.



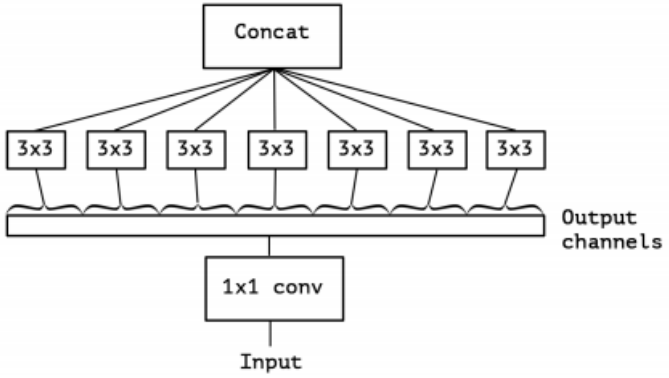
GoogLeNet Family

- Inception v3[6]: 在v2基础上引入“卷积核分解”的理念，提出使用连续非对称两个Nx1-1xN卷积层代替一个NxN卷积层，减少计算量



- Inception v4[7]: 在v3基础上略做修改，在稍微增加模型深度时进一步提升识别精度

- Xception[8]：改进Inception模块，提出Depth-wise Separate Conv（group=输出通道数）

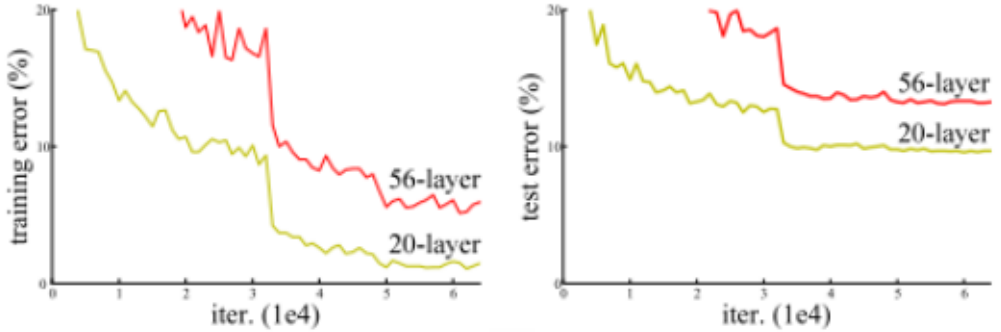


ResNet (2015) [9]

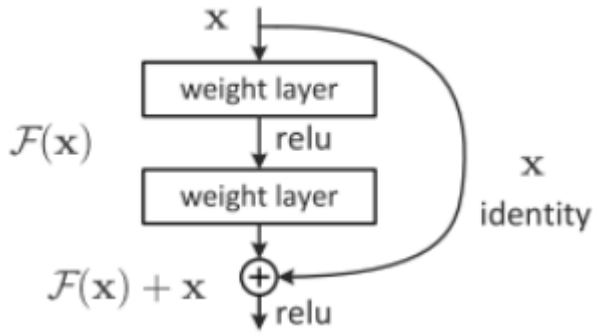
- ImageNet竞赛 2015年冠军方案，CVPR 2016 best paper。图像分类识别率超过人类水平
- 提出152层的CNN结构，大大超越之前最深的23层GoogLeNet网络
- 提出“残差学习”思想，使用简单的“跳层”连接（shortcut）大大缓解深度网络训练时的梯度消失问题

ResNet (2015)

- 动机：深度对识别效果影响很大，理论上越深的网络效果应该越好。实际上简单加深时效果反而变差，主要原因是梯度消失（弥散）

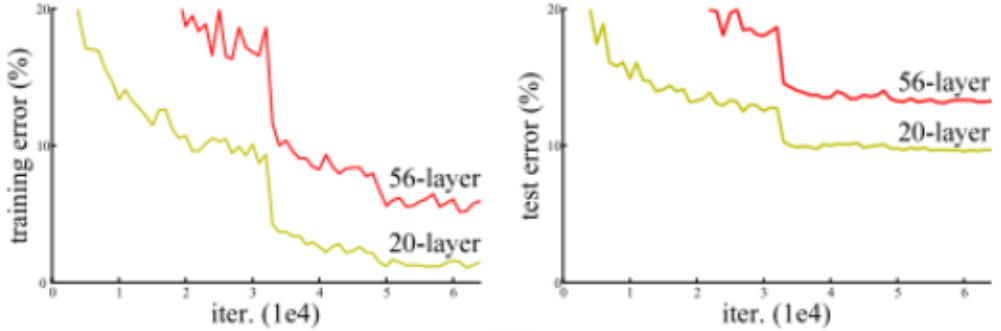


- 解决方案：使用跨层的连接，方向传播时部分梯度可以绕过residual path，直接沿着identity path跳过若干层，回到输入层，从而在网络很深时也可以避免梯度消失



ResNet (2015)

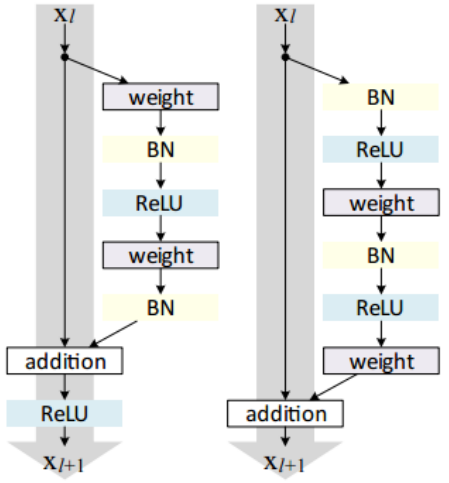
- 动机：深度对识别效果影响很大，理论上越深的网络效果应该越好。实际上简单加深时效果反而变差，主要原因是梯度消失（弥散）



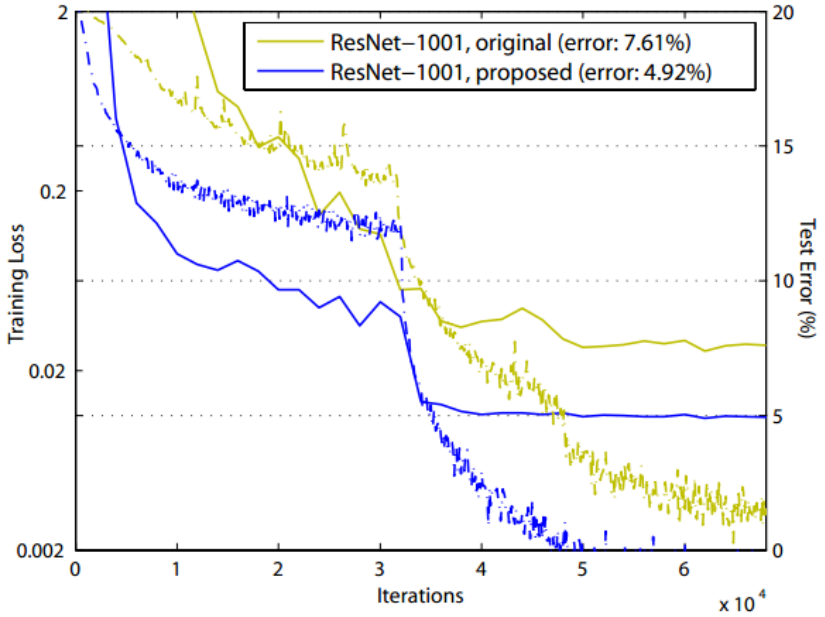
- 解决方案：使用跨层的连接，方向传播时部分梯度可以绕过residual path，直接沿着identity path跳过若干层，回到输入层，从而在网络很深时也可以避免梯度

ResNet Family

- ResNet-v2[10]: 把Residual block中Conv-BN-ReLU顺序换成了BN-ReLU-Conv，使得shortcut path比基本的ResNet更加干净，梯度回传基本可以无损从输出层完整传回输入层

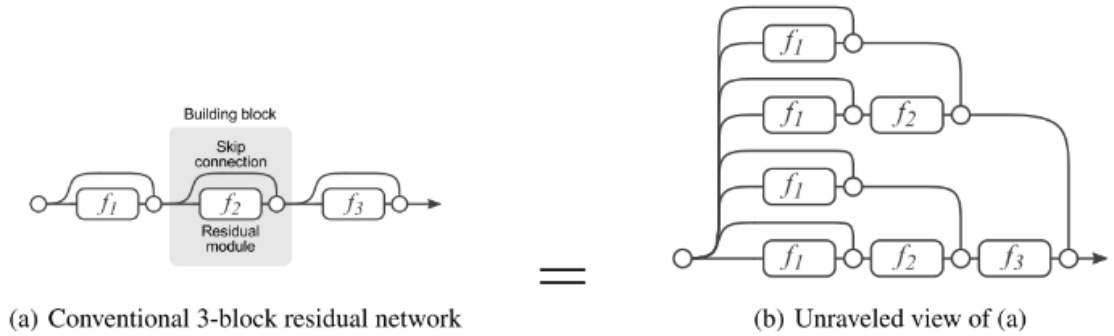


(a) original (b) proposed



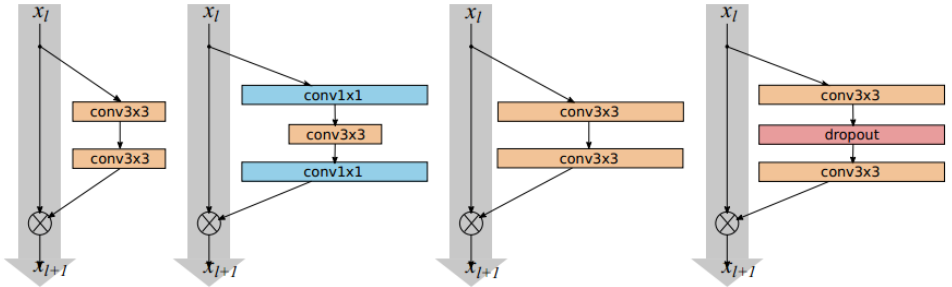
ResNet Family

- ResNet解读[25]：提出ResNet的一种理解方式，可以看成是一系列浅层网络的集成



(a) Conventional 3-block residual network (b) Unraveled view of (a)

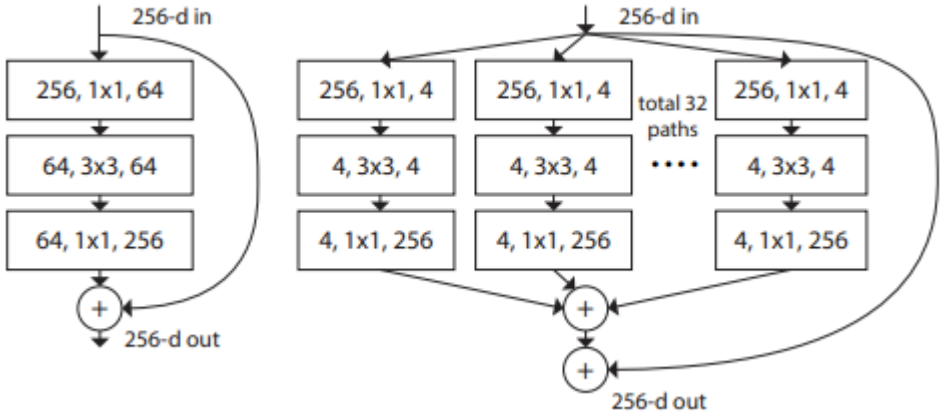
- Wide Residual Network[12]
  - 设计更浅但更宽（通道数更多）的ResNet



(a) basic (b) bottleneck (c) basic-wide (d) wide-dropout

ResNet Family

- ResNeXt[11]：把ResNet的residual path中的卷积部分使用AlexNet中group机制分组，维持参数量基本不变情况下增加网络的表达能力



- ResNet with DropLayer[24]：以一定概率随机把ResNet中的residual path输出置为0，只保留shortcut，等价于删除了该block

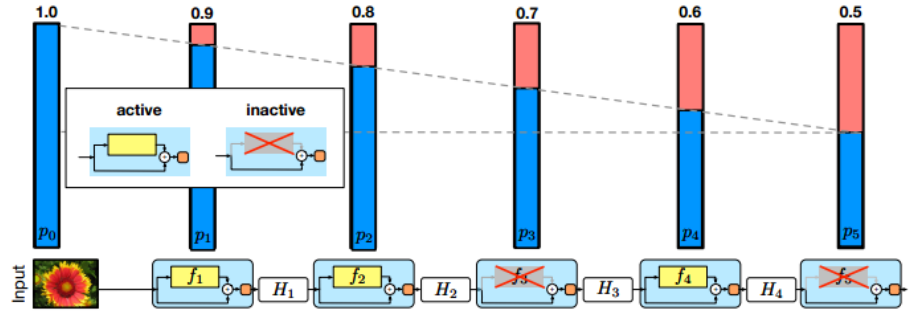
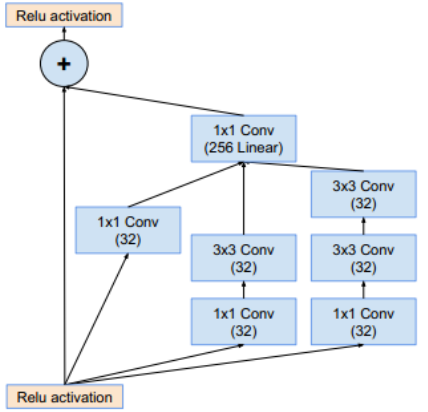


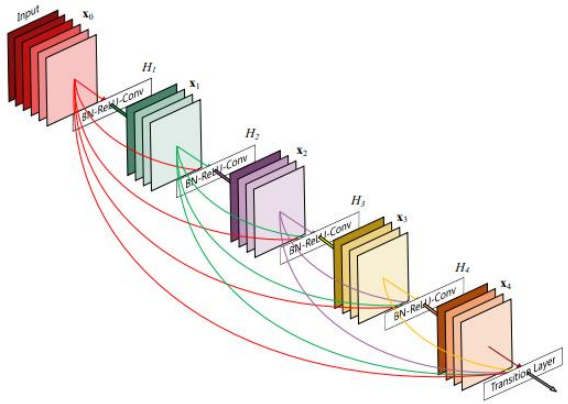
Fig. 2. The linear decay of p_ℓ illustrated on a ResNet with stochastic depth for $p_0 = 1$ and $p_L = 0.5$. Conceptually, we treat the input to the first ResBlock as H_0, which is always active.

融合ResNet和Inception

- Inception-ResNet[7]：把residual思想引入Inception结构中，进一步提升Inception-v4的识别效果



- DenseNet[13]：引入ResNet中跨层连接思想，但是还是使用Inception中的特征图concat而不是ResNet中的sum策略，比ResNet参数效率更高。CVPR2017 best paper



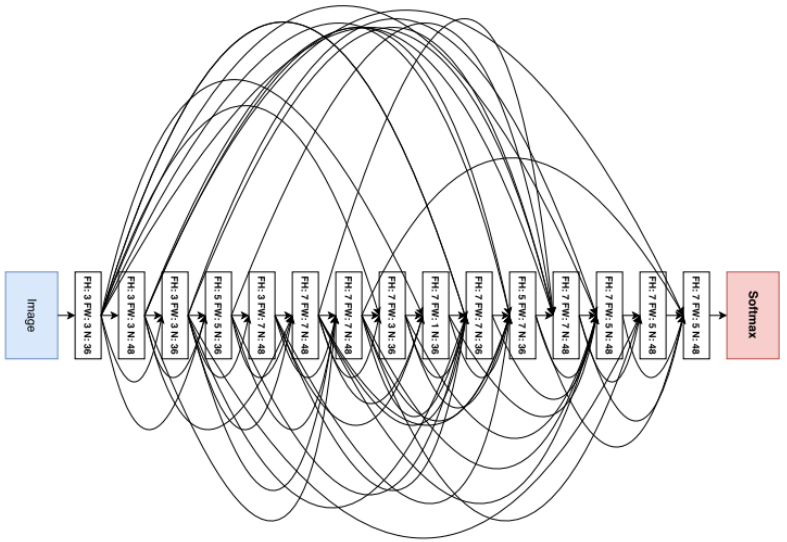
PART III. CNN发展新方向

CNN发展新方向

- 模型压缩，为CNN模型移动端部署提供条件
  - 基于卷积核分解：如Inception-v3中的非对称卷积
  - 减少卷积核个数：删除冗余的卷积核[14]
  - 模型迁移学习：大模型的知识迁移到小模型[15]
  - 设计更小的网络结构：如SqueezeNet [16]、MobileNet [17]、ShuffleNet [18]
  - 低精度表达网络：如二值的XOR-Net [19]， BNN [20]， 三值Ternary-Net [21]或四值网络Two-bit-Net [22]。
- 自动学习新的结构
  - 基于基本结构，使用强化学习算法[19]，自动学习最优的网络结构

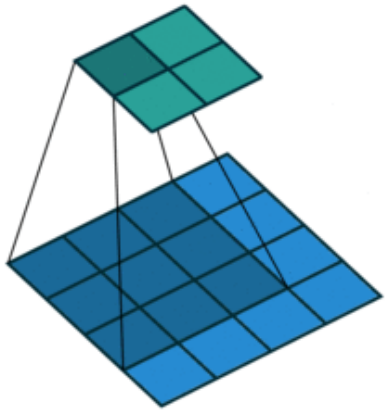
CNN发展新方向

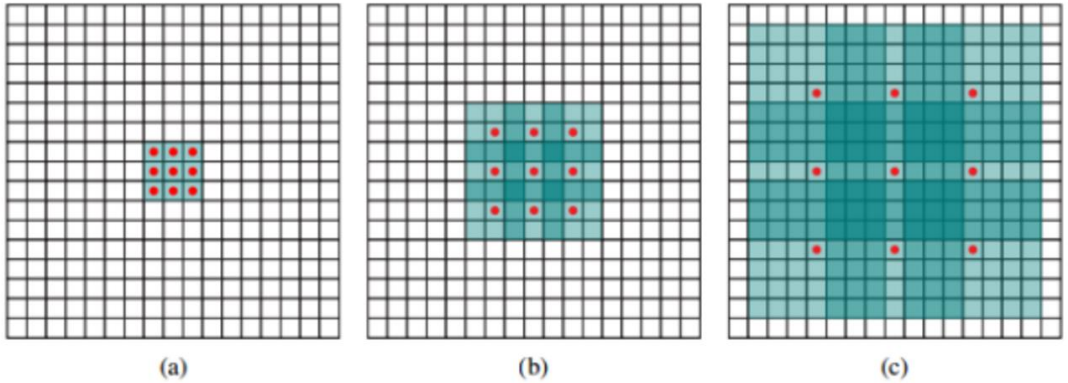
- 模型压缩，为CNN模型移动端部署提供条件
  - 基于卷积核分解：如Inception-v3中的非对称卷积
  - 减少卷积核个数：删除冗余的卷积核[14]
  - 模型迁移学习：大模型的知识迁移到小模型[15]
  - 设计更小的网络结构：如SqueezeNet[16]、MobileNet[17]、ShuffleNet[18]
  - Low-precision网络：如二值的XOR-Net[19]，Net[21]或四值网络Two-bit-Net[22]。
- 自动学习新的结构
  - 基于基本结构，使用强化学习算法[19]，自

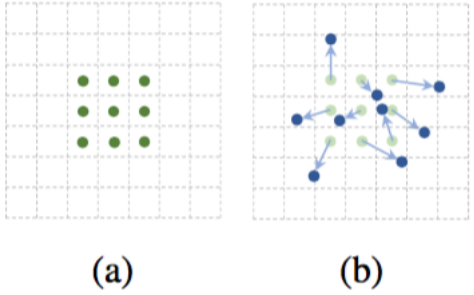


CNN发展新方向

- 卷积层改造
  - Deconv 反卷积[26]：也称转置卷积，用相比于传统卷积的缩小特征图尺寸，反卷积把特征图尺寸变大（带参数的差值）
  - Dilated Conv 带孔卷积[27]：卷积核中间“带孔”，不增加额外参数时增大感受野，常用在物体检测中
  - Deformable Conv 可变形卷积[28]：特殊的学习卷积核参数的同时学习偏移量，解决传统卷积核只能处理固定形状的问题

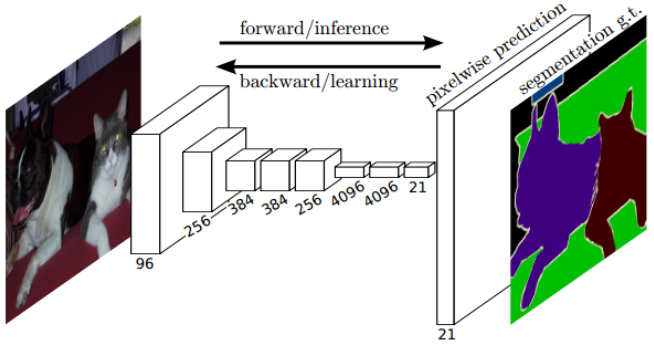




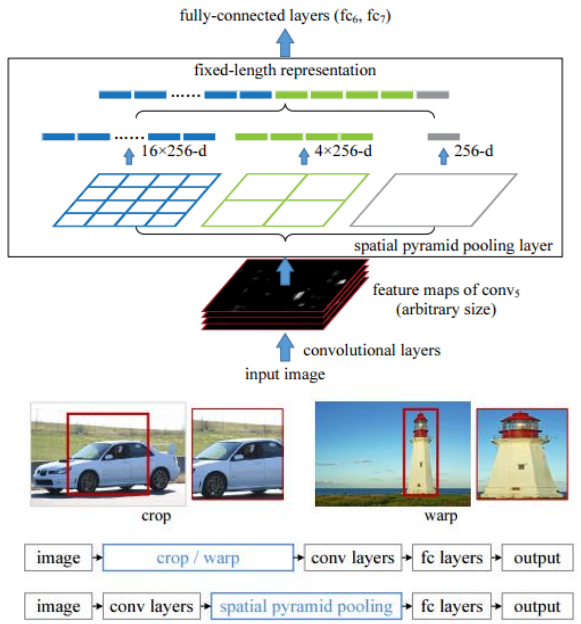


CNN发展新方向

- 不同应用中的CNN结构
  - 全卷积网络[29]：没有全连接层的CNN网络，在一系列卷积后使用反卷积还原回原始输入尺寸，用在像素级别预测，如图像分割



  - SPPNet [30]：支持任意大小输入的CNN网络，最后的卷积层后使用空间金字塔池化把任意宽高的特征图变成定长的向量



总结

- CNN结构是深度学习高速发展的源动力：AlexNet (14k+ citations), ResNet (3.3k+ citations)
- 好的CNN模型具有强大的迁移能力，在新数据集上微调后在很多计算机视觉领域，如物体检测、图像分割、视频跟踪、视频理解都取得了好的效果
- 人工经验目前还是CNN设计的主流，目前研究也遇到了瓶颈，ResNet之后没有全新的突破，新的突破或许将来自于机器自主学习

解答

- Q1：conv和pooling层在计算输出尺寸时有什么区别？
  - Conv窗口在滑动时不能超出图片，如果部分窗口越界，则舍弃该窗口，pooling层允许部分划窗越界,未越界部分也可以进行池化操作

解答

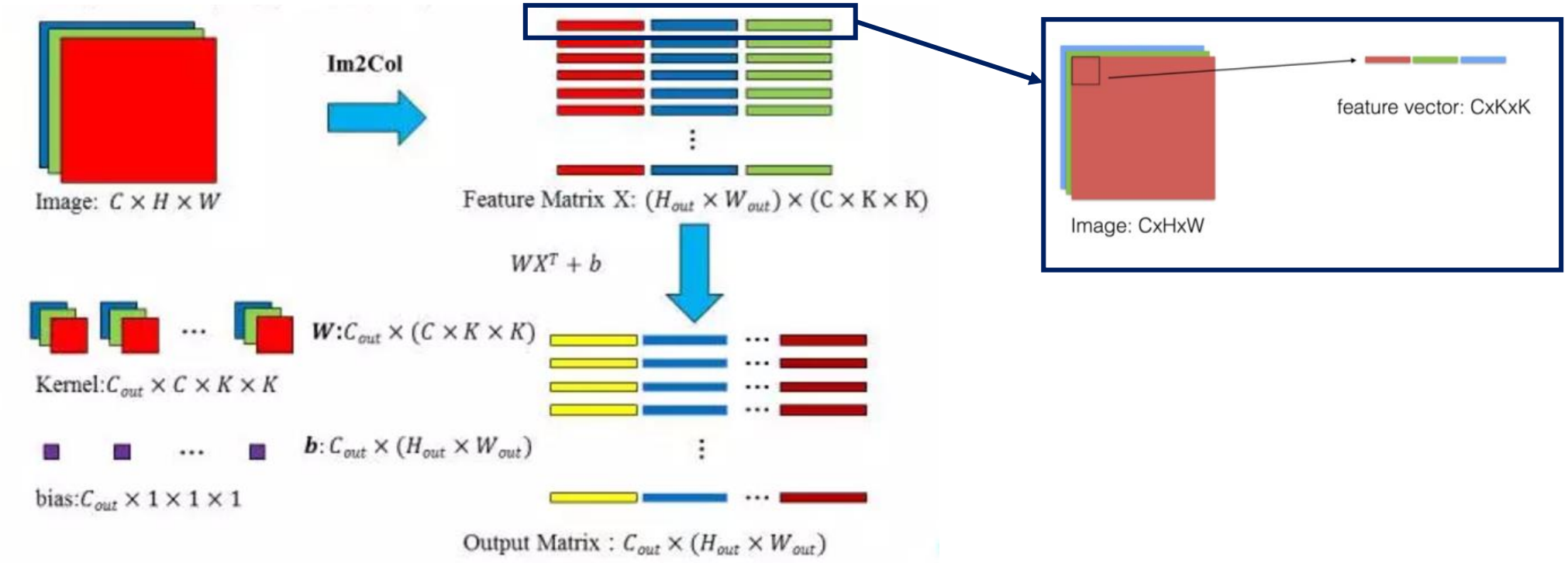
- Q1：conv和pooling层在计算输出尺寸时有什么区别？
  - Conv窗口在滑动时不能超出图片，如果部分窗口越界，则舍弃该窗口，pooling层允许部分划窗越界,未越界部分也可以进行池化操作
- Q2：sigmoid和tanh函数之间在数学上有什么关联？
  - tanh(x)= 2sigmoid(2x)-1

解答

- Q1：conv和pooling层在计算输出尺寸时有什么区别？
  - Conv窗口在滑动时不能超出图片，如果部分窗口越界，则舍弃该窗口，pooling层允许部分划窗越界,未越界部分也可以进行池化操作
- Q2：sigmoid和tanh函数之间在数学上有什么关联？
  - tanh(x)= 2sigmoid(2x)-1
- Q3：卷积和全连接层在形式上有什么关联？
  - There is no such thing as “fully-connected layers”. There are only convolution layers with 1x1 convolution kernels. ---- Yann LeCun

解答

- Q4: 两者可否复用一套代码实现?
  - 可以把卷积层转变成全连接层的W^Tx+b的形式(Caffe中的实现)。好处是可以把循环结构变成基本的矩阵运算，可以用现有的运算库如openblas，或cuda下的cublas高效实现



相关资料

[1]. A. Krizhevsky et al, ImageNet classification with deep convolutional neural networks, NIPS, 2012

[2]. M. Lin et al, Network in network, ICLR, 2013

[3]. K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, NIPS, 2014

[4]. C. Szegedy et al, Going deeper with convolutions, NIPS, 2015

[5]. S. Loffe and C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, ICML, 2015

[6]. C. Szegedy et al, Rethinking the inception architecture for computer vision, CVPR, 2016

[7]. C. Szegedy et al, Inception-v4, inception-resnet and the impact of residual connections on learning, CVPR, 2016

[8]. F. Chollet, Xception: Deep Learning with Depthwise Separable Convolutions, arXiv, 2016

[9]. K. He et al. Deep residual learning for image recognition, CVPR, 2016

[10]. K. He et al, Identity Mappings in Deep Residual Networks, ECCV, 2016

[11]. S. Xie et al, Aggregated residual transformations for deep neural networks, CVPR, 2017

[12]. S. Zagoruyko et al, Wide residual networks, BMVC, 2016

相关资料

[13]. G. Huang et al, Densely connected convolutional networks. CVPR, 2017

[14]. H. Song et al, Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman coding, ICLR, 2016

[15]. G. Hinton et al, Distilling the knowledge in a neural network, NIPS, 2015

[16]. I. Forrest et al, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size, CVPR. 2016

[17]. A. Howard et al, Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv, 2017

[18]. X. Zhang et al, ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices, arXiv, 2017

[19]. R. Mohammad et al, Xnor-net: Imagenet classification using binary convolutional neural networks, ECCV, 2016

[20]. M. Courbariaux et al, Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1}, arXiv, 2016

[21]. C. Zhu et al, Trained ternary quantization, arXiv, 2016

[22]. W. Meng et al, Two-bit networks for deep learning on resource-constrained embedded devices, arXiv, 2017

[23]. Z. Barret et al, Neural architecture search with reinforcement learning, ICLR, 2017

[24]. G. Huang et al, Deep Networks with Stochastic Depth, arXiv, 2016

[25]. A. Veit et al, Residual Networks Behave Like Ensembles of Relatively Shallow Networks. arXiv, 2016

相关资料

[26]. M. Zeiler et al, Visualizing and understanding convolutional networks, ECCV, 2014

[27]. F. Yu et al, Multi-scale context aggregation by dilated convolutions, arXiv, 2015

[28]. J. Dai et al, Deformable Convolutional Networks, CVPR, 2017

[29]. J. Long et al, Fully convolutional networks for semantic segmentation, CVPR, 2015

[30]. K. He et al, Spatial pyramid pooling in deep convolutional networks for visual recognition, ECCV, 2014

CAMALAB
计算机动画与多媒体分析实验室

Q&A

Thanks !