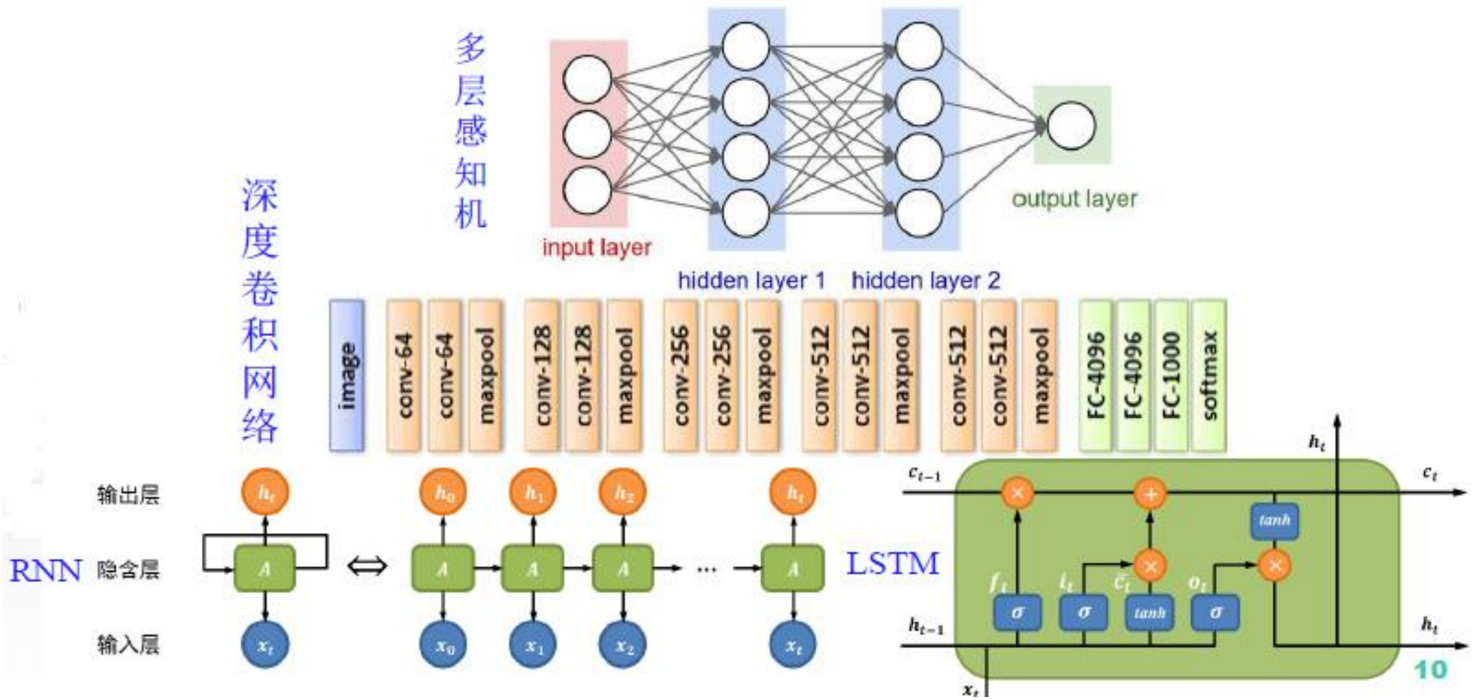
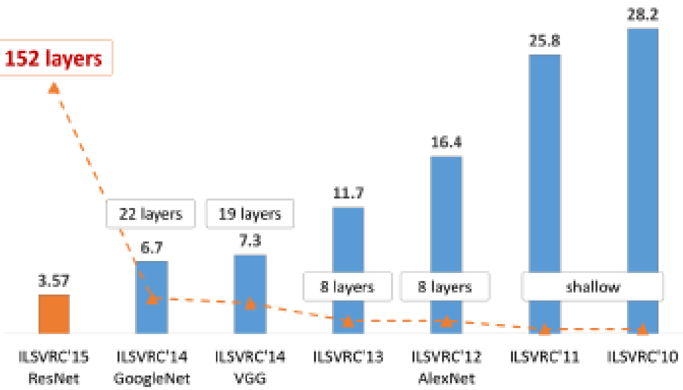
Brief Introduction of Deep Learning

- **深度学习是什么?**
  - 以不少于两个隐含层的神经网络对输入进行非线性变换或表示学习的技术
  - 包括多种结构：MLP，CNN，RNN/LSTM等

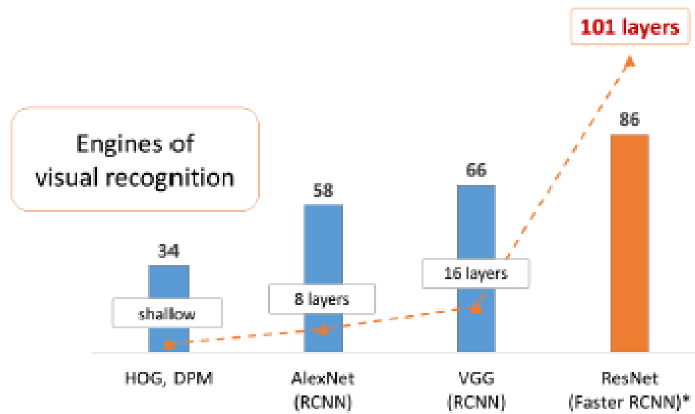


Brief Introduction of Deep Learning

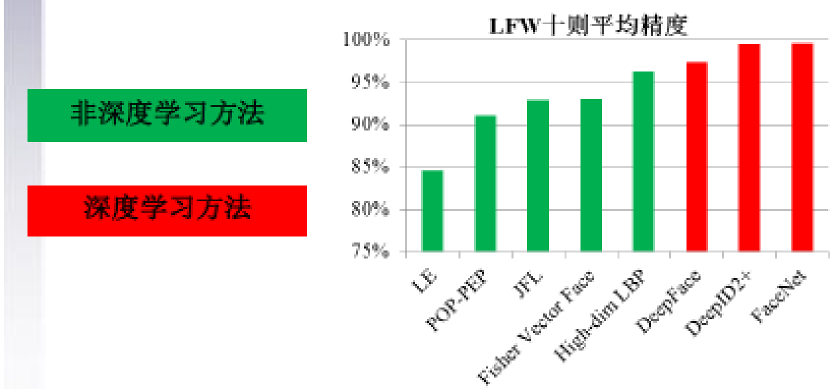
- 图像分类、人脸识别、物体检测领域的突破性进步

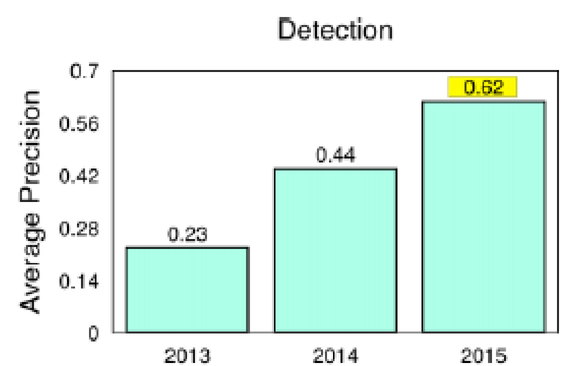


ImageNet图像分类Top-5错误率



Pascal VOC目标检测MAP

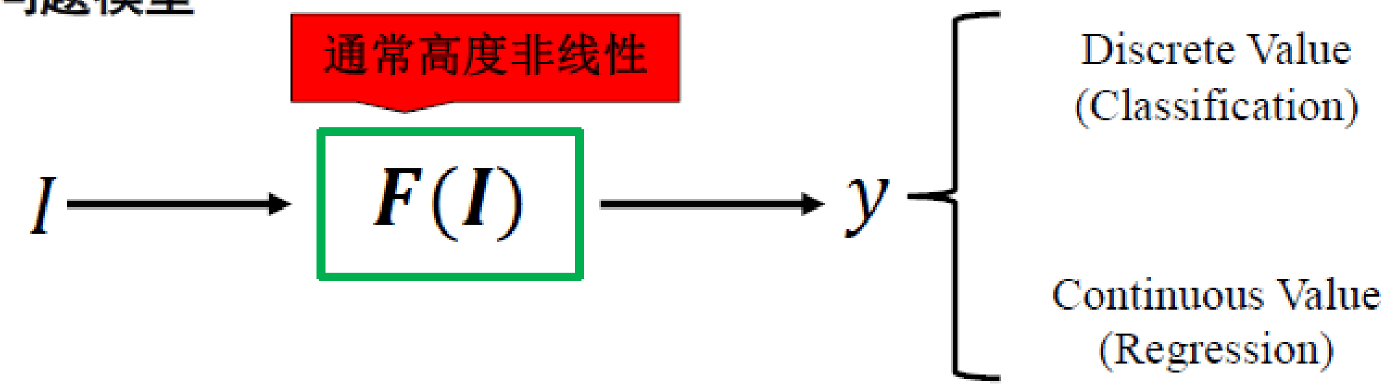




ImageNet目标检测MAP

Brief Introduction of Deep Learning

- 计算机视觉方法模型
  - 问题模型



  - 经典的两段式方法

图像表示：Gabor, SIFT, HOG, LBP, POEM, LGBP, LPQ
图像集表示：Manifold, GMM, Covariance

子空间学习&度量学习：PCA/LDA，Manifold, LMNN, NCA…..
词典学习&稀疏编码

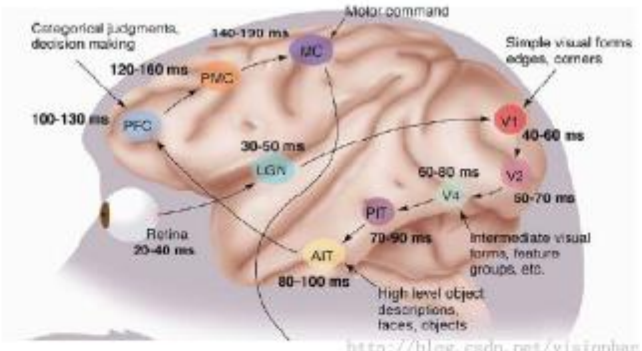
Brief Introduction of Deep Learning

- **计算机视觉方法模型**
  - 问题模型

通常高度非线性

$$I \longrightarrow \boxed{F(I)} \longrightarrow y \begin{cases} \text{Discrete Value (Classification)} \\ \text{Continuous Value (Regression)} \end{cases}$$

  - 深度学习带来的方法革命
    - 显示学习非线性映射$F(I)$
    - 分层非线性->逐层的语义抽象
    - 端到端学习(End to End)
    - 具有协同增效（synergy）优势



深度学习导引

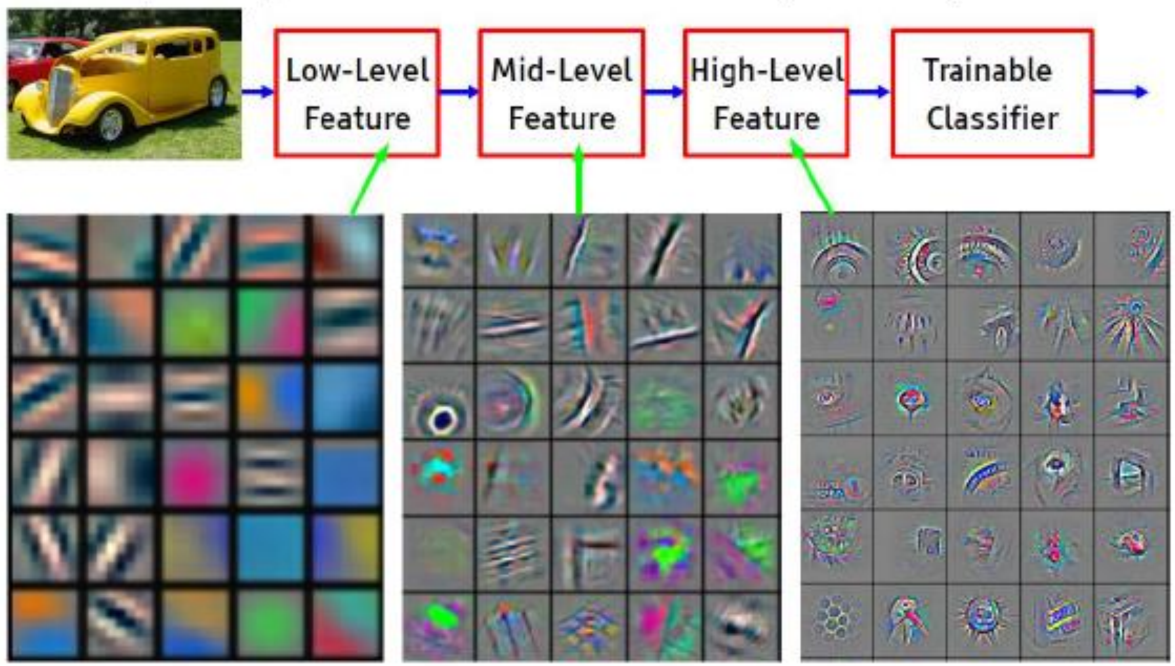
- **深度学习带来了什么样的方法变化？**
  - 经典的模式识别技术：手工设计的特征提取+分类器

  - 现代主流模式识别技术：无监督中层特征学习

  - 深度学习技术：端到端的层级特征学习

Brief Introduction of Deep Learning

- 深度学习学习了层级的特征
  - Image recognition: Pixel → edge → texton → motif → part → object
  - Text: Character → word → word group → clause → sentence → story
  - Speech: Sample → spectral band → sound → ... → phone → phoneme → word



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Brief Introduction of Deep Learning

- 深度网络具有语义抽象层次不断提高的感受野
  - 细节纹理到局部块再到特定物体的语义递进

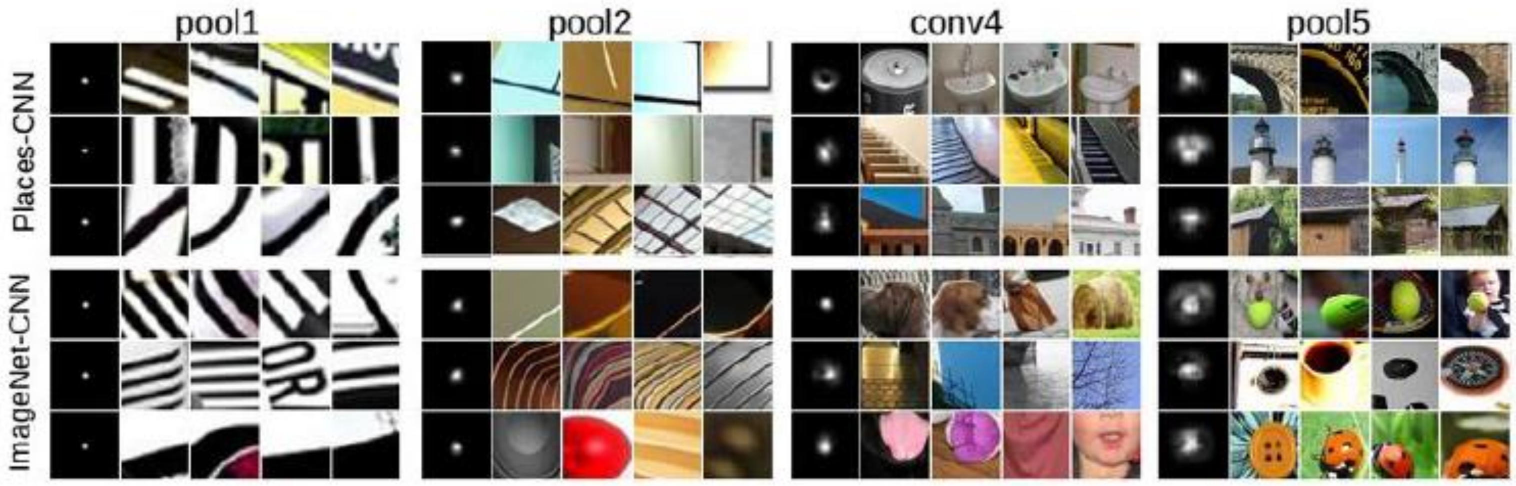


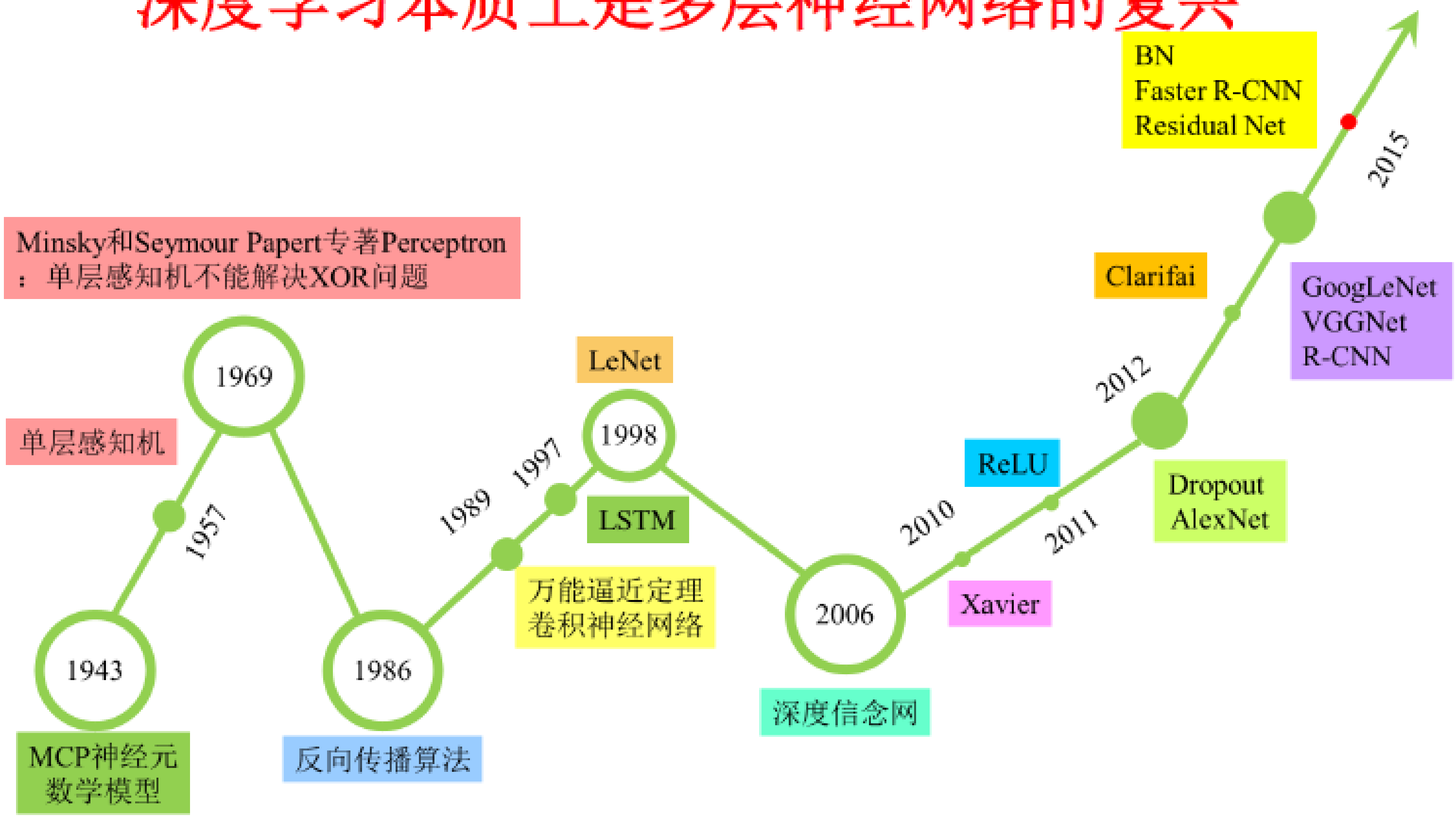
Figure 4: The RFs of 3 units of pool1, pool2, conv4, and pool5 layers respectively for ImageNet- and Places-CNNs, along with the image patches corresponding to the top activation regions inside the RFs.

感受野：一个感觉神经元的感受野是指这个位置里适当的刺激能够引起该神经元反应的区域

Brief Introduction of Deep Learning

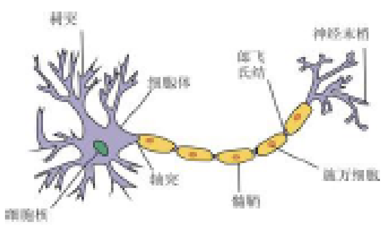
- 深度学习的曲折历史与光明未来

深度学习本质上是多层神经网络的复兴

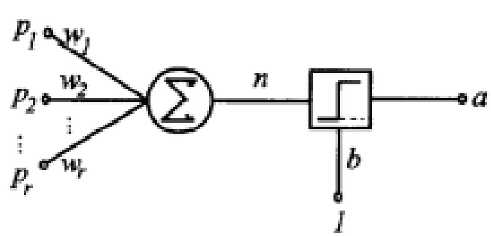


Brief Introduction of Deep Learning

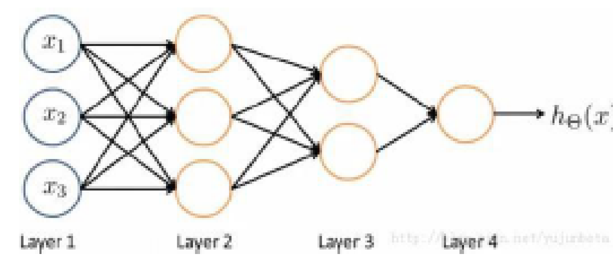
- 思想源于对人脑神经元的研究；
- 将神经的突触连接结构用抽象化的模型表达；
- 将单一神经元结构相互连接得到复杂的网络结构；



生物神经元模型



感知器
(Perceptron)

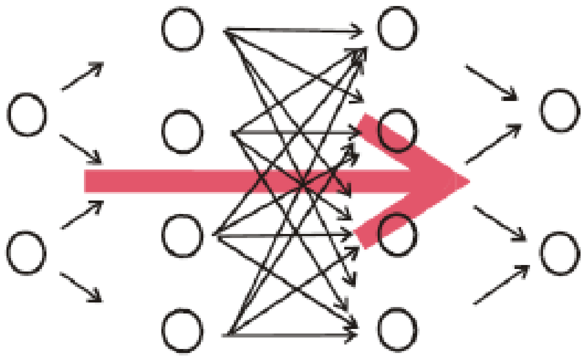


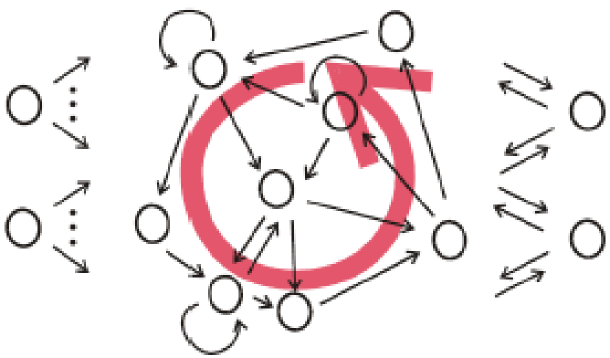
多层感知器
(MLP)

Brief Introduction of Deep Learning

经验
1. 传递方向
   CNN；RNN；
2. 神经元个数
   层级数+每层个数；神经元总数；
3. 激活函数
   Sigmoid；tanh；ReLu；

训练
4. 神经元间连接权重
5. 神经元偏差项(bias)

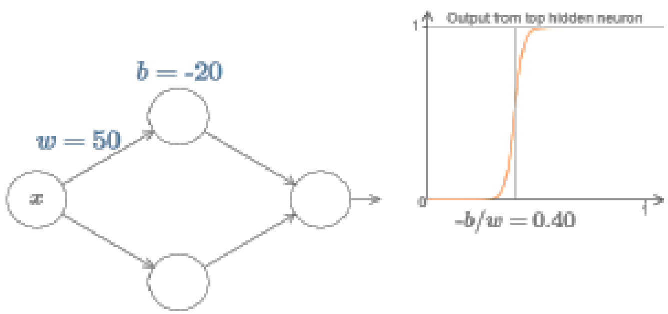


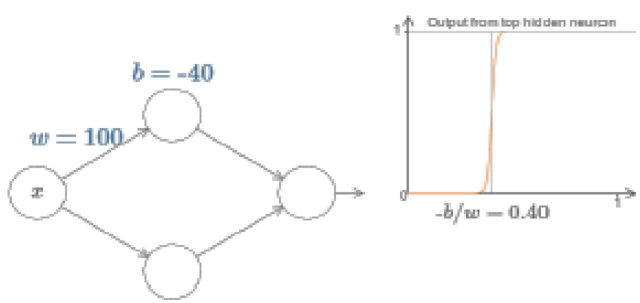


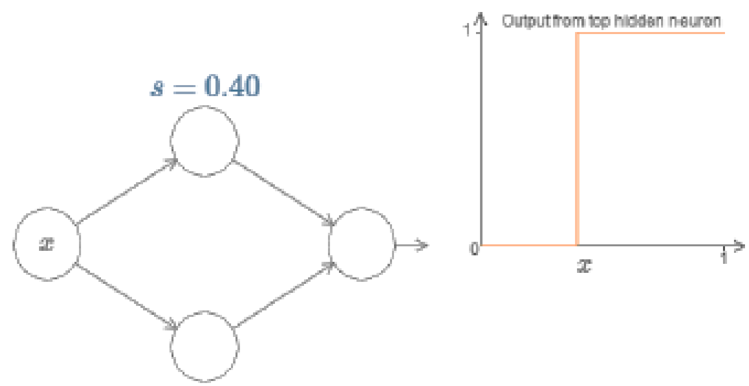
Brief Introduction of Deep Learning

- 普适逼近原理(Universal approximation theorem)：单隐藏层神经网络可以逼近任意函数[1]

$$y = sigmoid(wx + b) \qquad s = -b/w$$



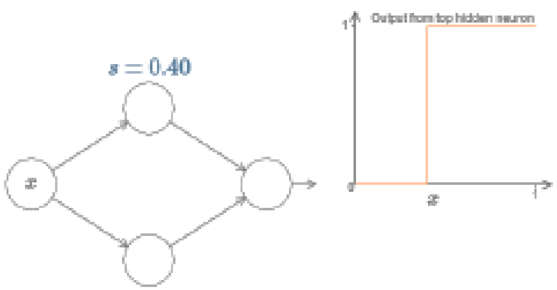


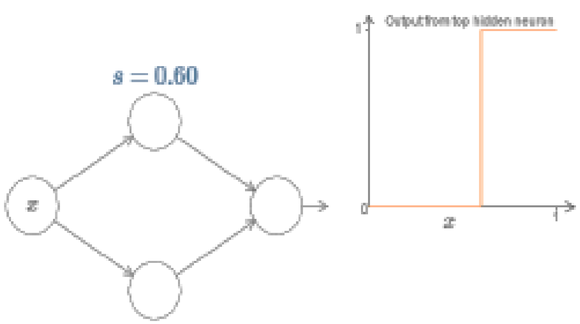


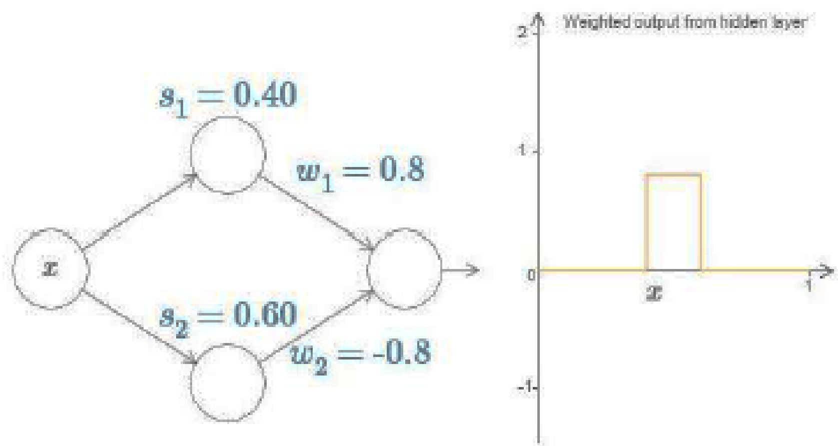
[1] Http://neuralnetworksanddeeplearning.com/chap4.html

Brief Introduction of Deep Learning

- 普适逼近原理(Universal approximation theorem)：单隐藏层神经网络可以逼近任意函数[1]



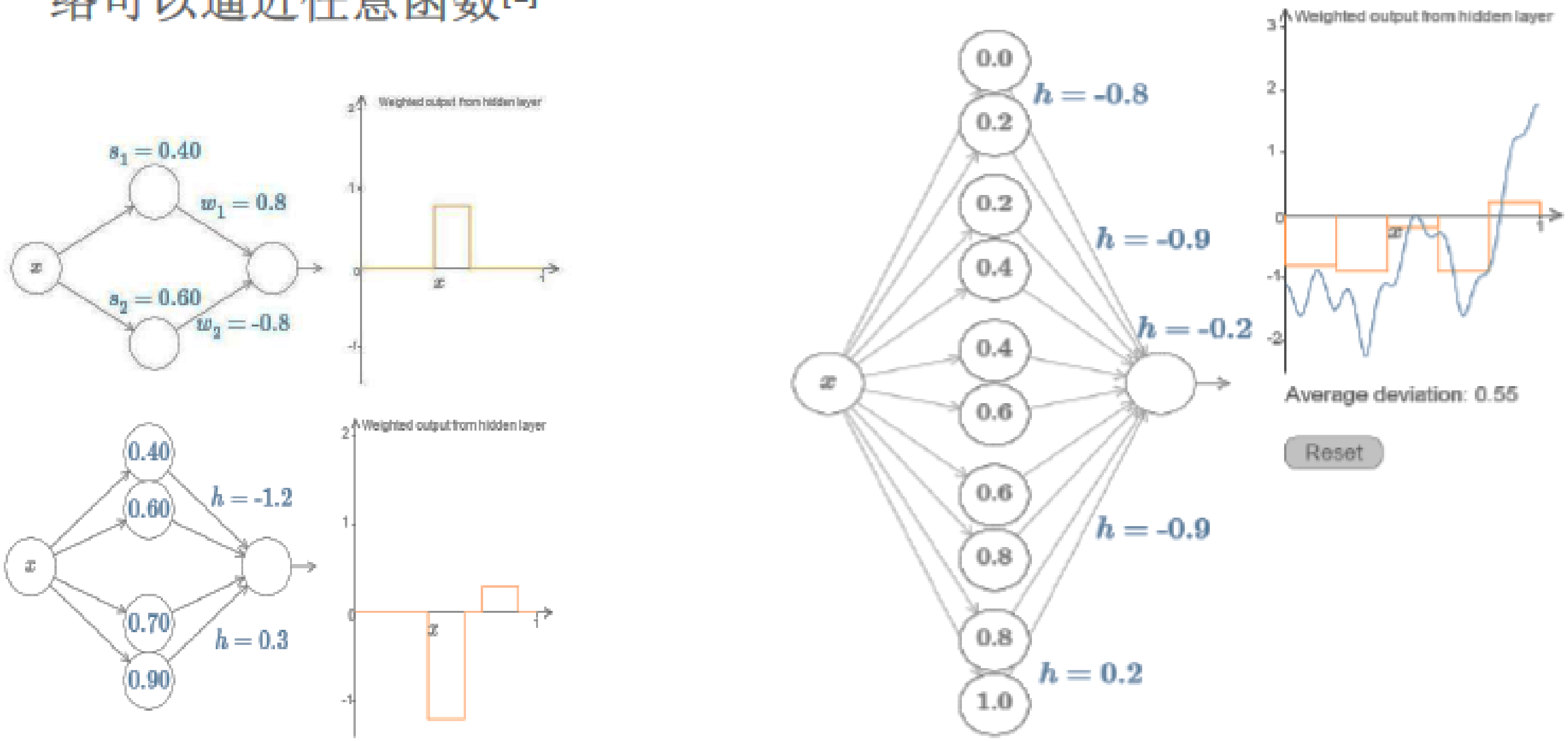




[1] Http://neuralnetworksanddeeplearning.com/chap4.html

Brief Introduction of Deep Learning

- 普适逼近原理(Universal approximation theorem)：单隐藏层神经网络可以逼近任意函数[1]



[1] Http://neuralnetworksanddeeplearning.com/chap4.html

Brief Introduction of Deep Learning

- 为什么要使用多隐藏层的结构？

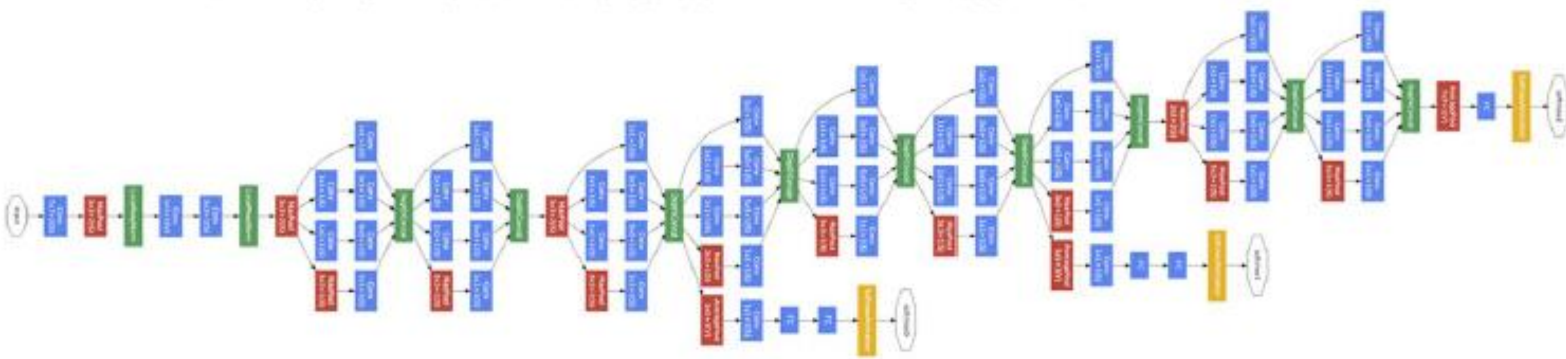
  1、适合多层知识体系学习；

  例：图像识别：边缘-几何形状-不变性特征-...

  2、泛化能力更强；

  3、隐藏层本质是一个特征探测器(feature detector)；

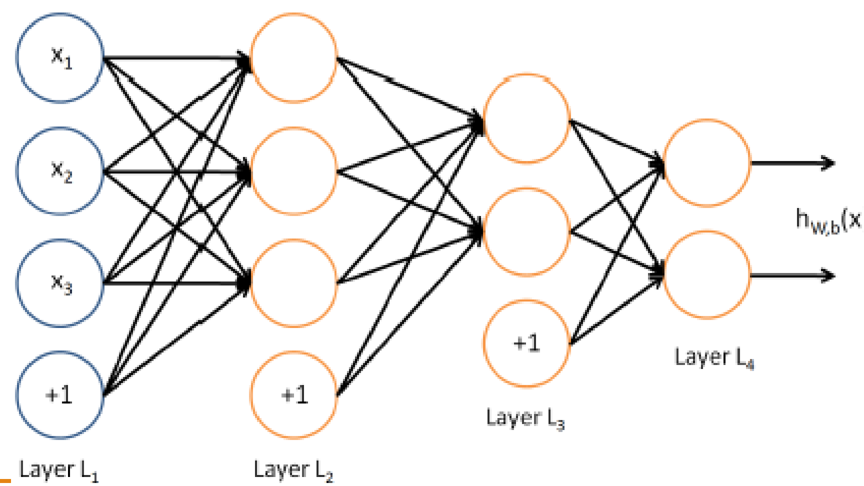
  学习数据的内在结构特征，不是映射关系；



Brief Introduction of Deep Learning

卷积神经网络

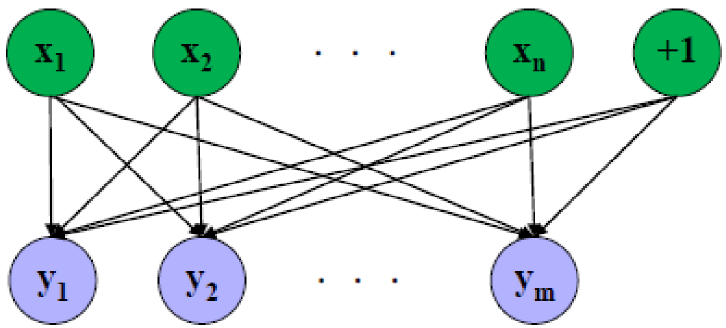
- 前向网络；
- 隐藏层有三类：卷积层(conv)，池化层(pooling)，全连接层(fc)；
- 输出层完成多分类任务，多为Softmax层；
- 网络训练算法：误差反向传播算法(BP)；



Brief Introduction of Deep Learning

卷积神经网络

- 全连接层
  - 相当于内积运算，图中“+1”表示偏置项b
  - 输出层的神经元和输入层的每个神经元都相连：得名“全”连接



  - Forward运算：$y = W^T x + b$，其中$y \in R^{m\times 1}, x \in R^{n\times 1}, W \in R^{n\times m}$
  - Backward运算：$\frac{\partial L}{\partial x} = W * \frac{\partial L}{\partial y}$，$\frac{\partial L}{\partial W} = x * \left(\frac{\partial L}{\partial y}\right)^T$

Brief Introduction of Deep Learning

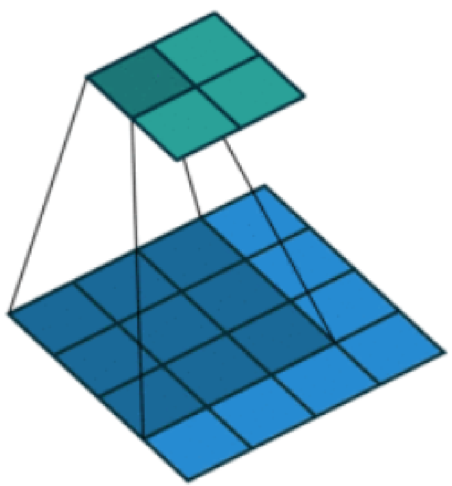
卷积神经网络

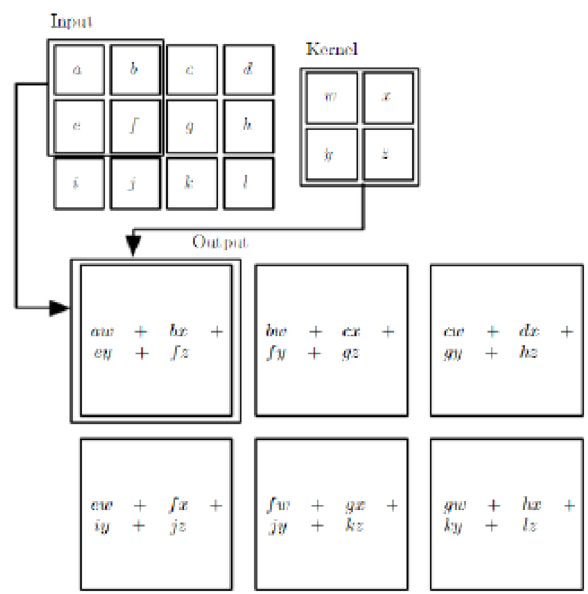
- **卷积层**
  - 2D卷积的数学形式

连续卷积：$h(x,y) = i * k(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} i(u,v)k(x-u,y-v)dudv$

离散卷积：$H(x,y) = I * K(x,y) = \sum_m \sum_n I(m,n)K(x-m,y-n)$

Caffe实现：$H(x,y) = I * K(x,y) = \sum_m \sum_n I(x+m,y+n)K(m,n)$

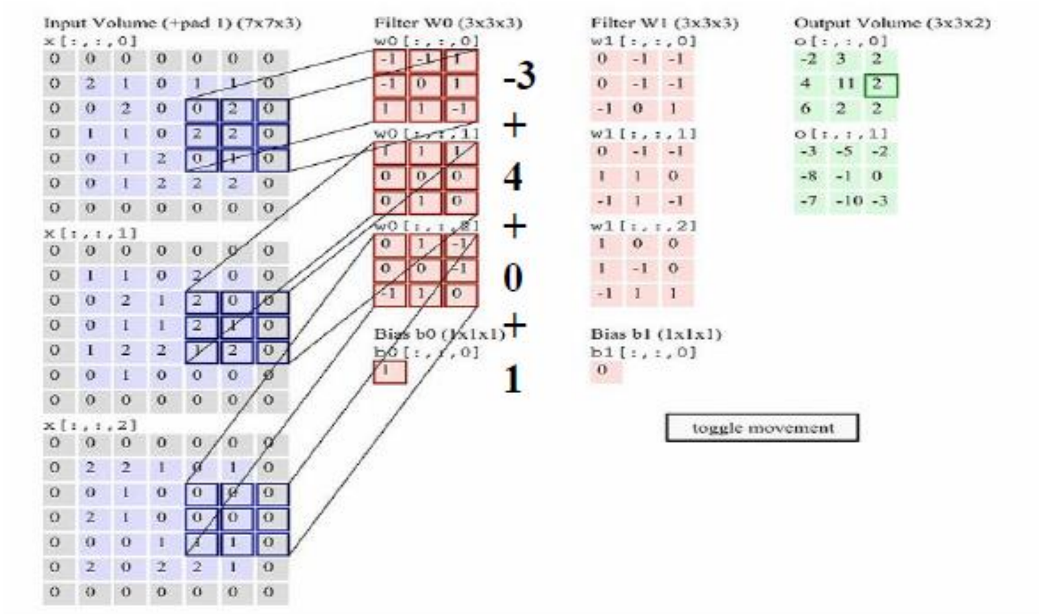




Brief Introduction of Deep Learning

卷积神经网络

- **卷积层**
  - 多个Feature Map的计算

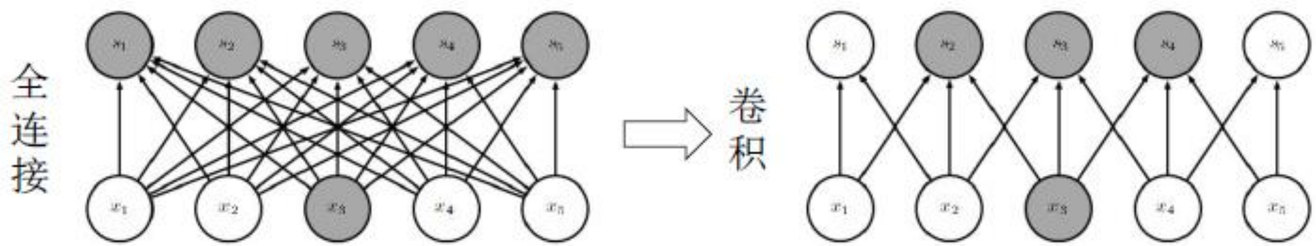


Animation: Andrej Karpathy http://cs231n.github.io/convolutional-networks/

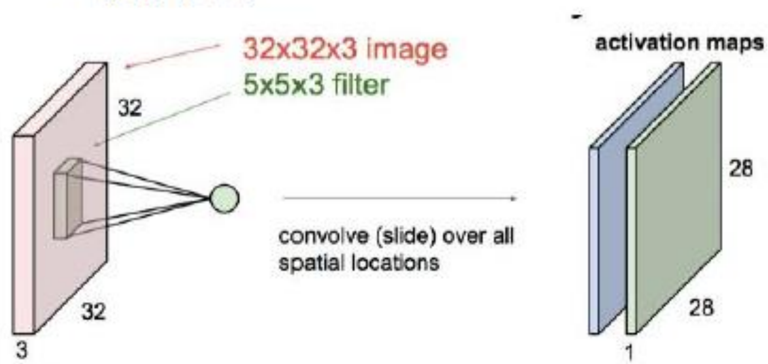
Brief Introduction of Deep Learning

卷积神经网络

- 卷积层
  - 稀疏连接：输出层神经元只和部分输入层神经元相连



  - 权值共享

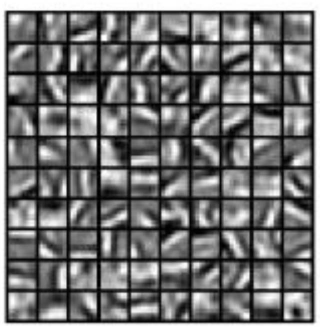


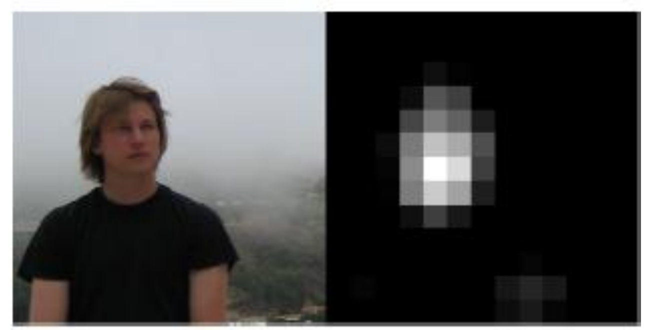
参数数量：5x5x3

Brief Introduction of Deep Learning

卷积层的作用

- 概念来自于信号与系统，其数学表达类似于信号互相关(Cross-correlation)；
- 卷积运算也可为一种提取特征的滤波过程；
- 在图像领域常表现为模板(mask)运算；
- 底层获得边缘信息，高层表达更鲁棒的特征[1]；



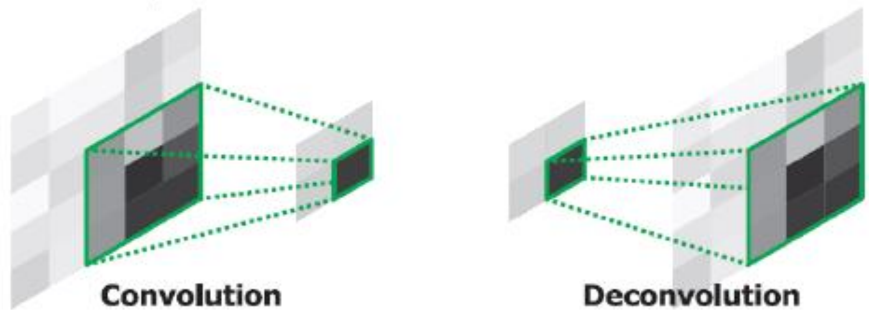


[1] Understanding Convolution in Deep Learning

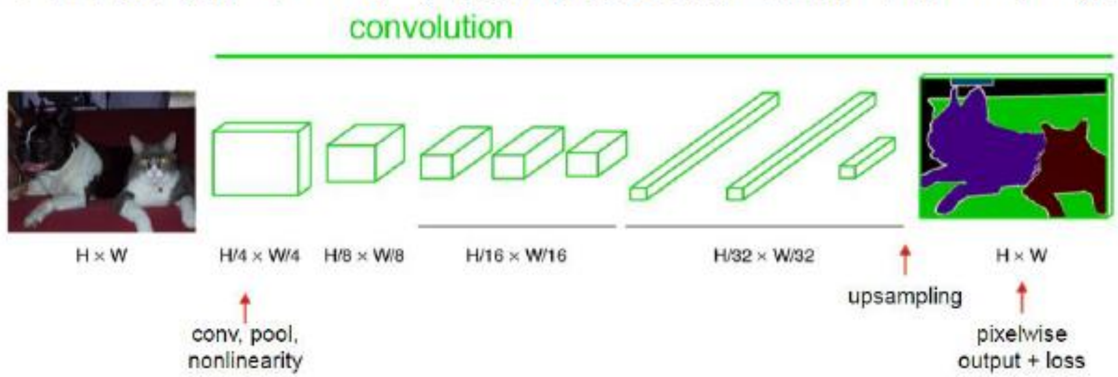
Brief Introduction of Deep Learning

卷积层的作用

- **反卷积层**
  - 卷积的逆过程，实现信号复原



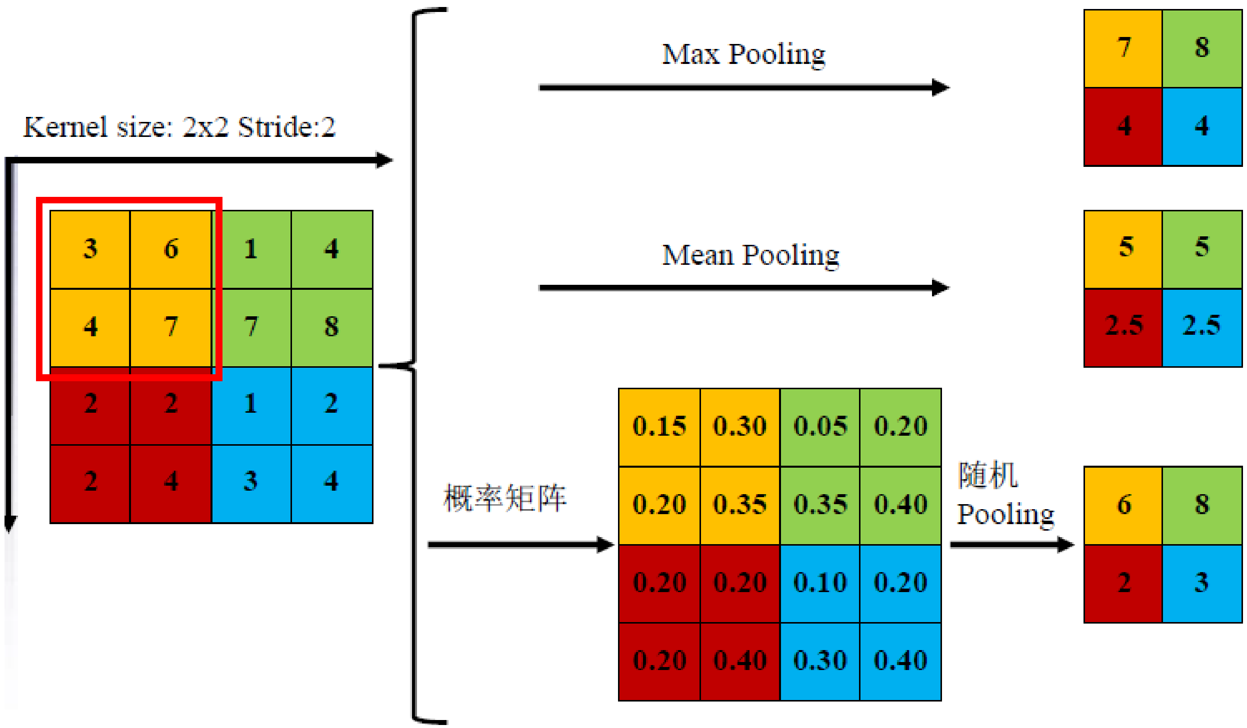
  - 全卷积网络（FCN），反卷积层实现上采样（upsampling）



Brief Introduction of Deep Learning

池化层的作用

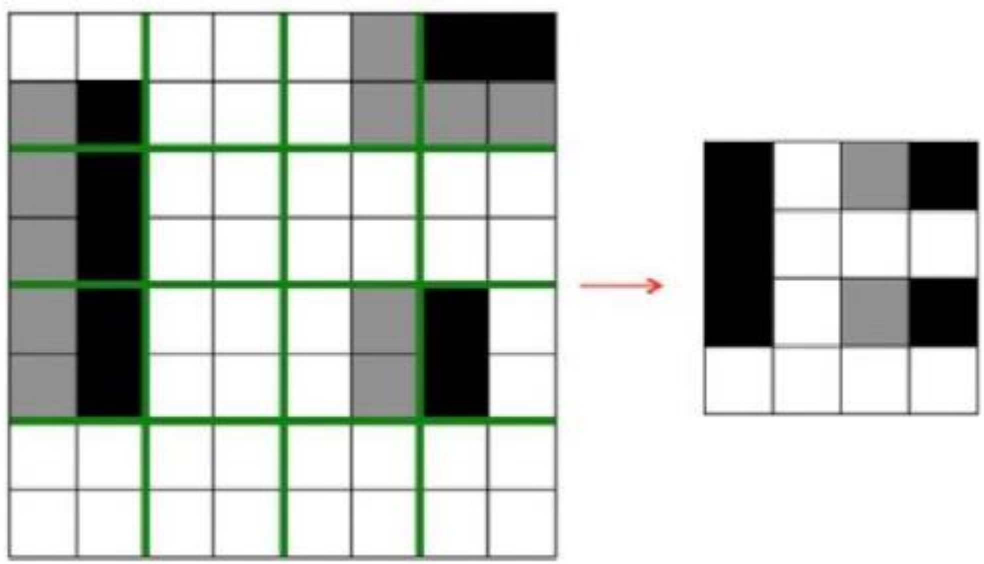
- **Pooling层**
  - 一般配合卷积层使用



Brief Introduction of Deep Learning

池化层的作用

- 池化层常用最大池化(max-pooling)，也有平均池化；
- 减少数据量，抑制过拟合，提高鲁棒性；
- Dropout(类似的正则化操作)：随机丢弃部分数据，抑制过拟合；



Brief Introduction of Deep Learning

- **激活函数**
  - 用于卷积层和全连接层之后
  - 网络非线性来源

Sigmoid $S(x)=1/(1+e^{-x})$

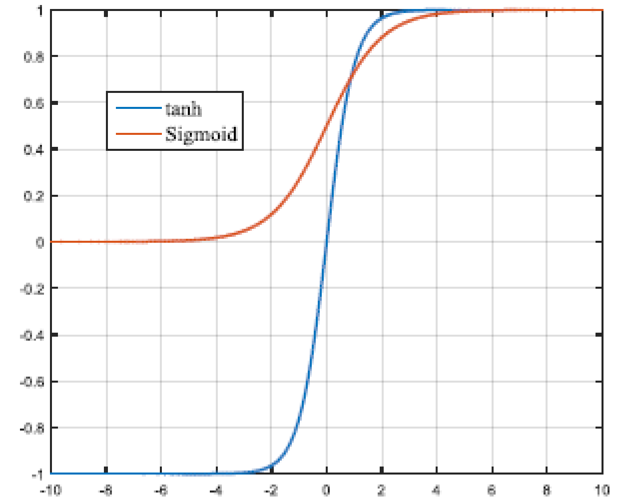
tanh $\tanh(x)$

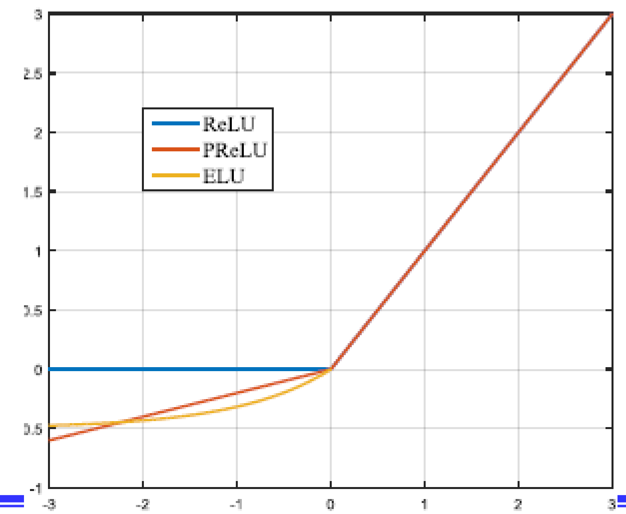
ReLU $\max(0,x)$

PReLU $\max(ax,x)$

ELU $y=\begin{cases} x \text{ if } x>0 \\ \alpha(e^{x}-1) \text{ if } x\le 0 \end{cases}$

Maxout $\max(w_1x_1+b_1, w_2x_2+b_2)$





Brief Introduction of Deep Learning

激活函数的作用

- 激活函数的作用
  - 增加网络的非线性性，拓展了网络的表达能力；
  - 使输出为连续值，便于网络训练；

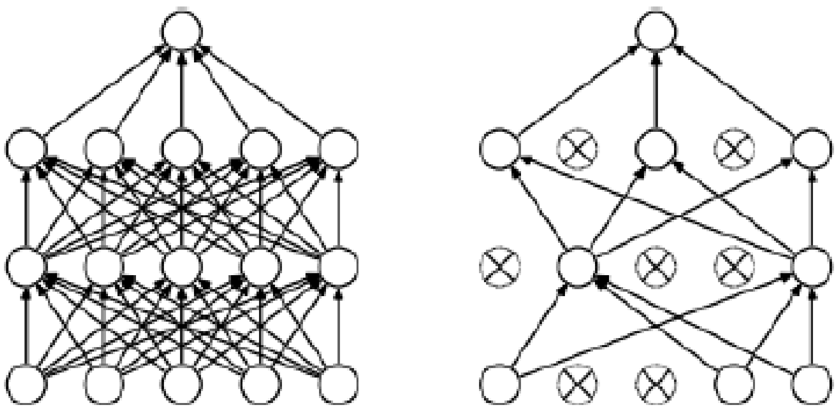
- 常用BP算法训练神经网络，故激活函数的求导性能影响关键；
  - Sigmoid函数：输入值过大或过小时，梯度值过小；
  - tanh函数：输入值过大或过小时，梯度值过小；
  - ReLu函数(使用广泛)：导数是分段线性的，出现稀疏化的结果；

- **Dropout（2012）**
  - 引入Bernoulli随机数u, p代表dropout ratio

$$y_{\text{train}} = \begin{cases} \frac{x}{1-p} & \text{if } u > p \\ 0 & \text{otherwise} \end{cases} \qquad \text{Where, } u \sim U(0,1)$$

$$E(y_{\text{train}}) = p \cdot 0 + (1-p)\frac{E(x)}{1-p} = E(x)$$

  - 测试阶段：Do Nothing
  - 正则化手段，提高泛化能力



(a) Standard Neural Net (b) After applying dropout.

卷积神经网络

- **Batch Normalization（2015）**
  - 逐层尺度归一，避免了梯度消失和梯度溢出
  - 加速收敛5x~20x, 同时作为一种正则化技术也提高了泛化能力

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\ldots m}\}$;
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.

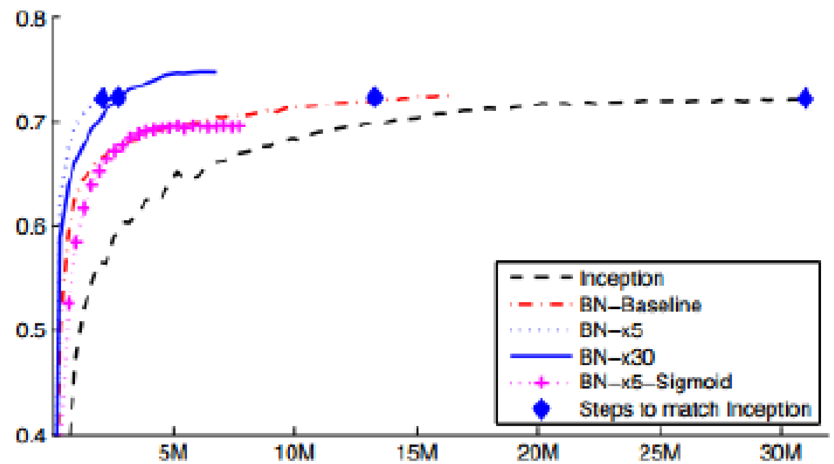


Figure 2: *Single crop validation accuracy of Inception and its batch-normalized variants, vs. the number of training steps.*

Conv → **BN** → **ReLU** → **FC** → **BN** → **ReLU**

卷积神经网络

■ 损失函数

- Softmax + Cross Entropy Loss

损失函数：$E = \frac{-1}{N}\sum_{n=1}^{N}\log(\hat{p}_{nl_n}), l_n \in [0,1,\ldots,K-1]$

其中 $\hat{p}_{nl_n}$ 由softmax函数计算 $\hat{p}_{nk} = \frac{e^{x_{nk}}}{\sum_{l=0}^{K-1} e^{x_{nl}}}$

适用场景：单标签分类问题

```
layer {
  name: "loss"
  type: "SoftmaxWithLoss"
  bottom: "fc8"
  bottom: "label"
  top: "loss"
}
```

一般是全连接层输出，结点数等于类别数

假设C为类别数，label的取值范围为0~C-1

卷积神经网络

Softmax层的作用

- 二分类问题(logistic回归)：

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$$

$$p(y \mid x;\theta) = (h_\theta(x))^y (1-h_\theta(x))^{1-y}$$

$$\begin{aligned}\ell(\theta) &= \log L(\theta) \\ &= \sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1-y^{(i)}) \log(1-h(x^{(i)}))\end{aligned}$$

- 多分类问题即将二分类问题拓展；
- Softmax损失函数为交叉熵损失函数(Cross Entropy Error Function)；

$$J = -\sum_{j=1}^{k} 1(y=j) \log \frac{e^{\theta_j^T x}}{\sum_{l=1}^{k} e^{\theta_l^T x}}$$

卷积神经网络

- **损失函数**
  - **Euclidean Loss**

损失函数：$E = \frac{1}{2N}\sum_{n=1}^{N}\|\hat{y}_n - y_n\|_2^2$

适用场景：实数值回归问题

注意事项：Caffe实现中，欧式损失没有除去标签维度！

```
layer {
  name: "loss"
  type: "EuclideanLoss"
  bottom: "loss3/classifiersigmoid"
  bottom: "label"
  top: "loss"
}
```

Tips1: 默认的，Data层和ImageData层均不支持多维标签，可以使用HDF5Layer

Tips2: 欧式损失前可以增加Sigmoid操作进行归一化，相应的输出标签也归一化到[0,1]

卷积神经网络

网络的预处理

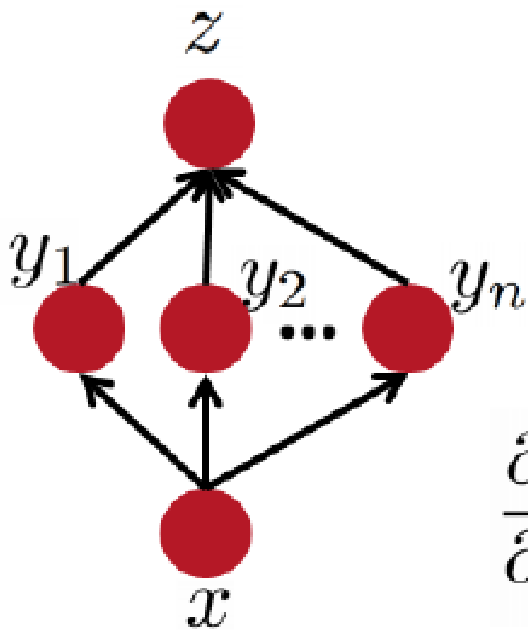
- 网络初始化
  - 权重值——经验公式：0.01*rand(D)/sqrt(D)；
  - 偏置量——全置0；
- 数据预处理
  - 数据归一化——简单缩放(彩色图像)；
    均值消减(灰度图像)；
  - 白化——使各维度分布一致，降低输入的冗余性；

卷积神经网络的训练方法

- **Back Propagation的数学基础**
  - 复合函数链式求导：

$$z = f(y) \quad y = g(x) \quad \frac{\partial z}{\partial x} = \frac{\partial z}{\partial y}\frac{\partial y}{\partial x}$$

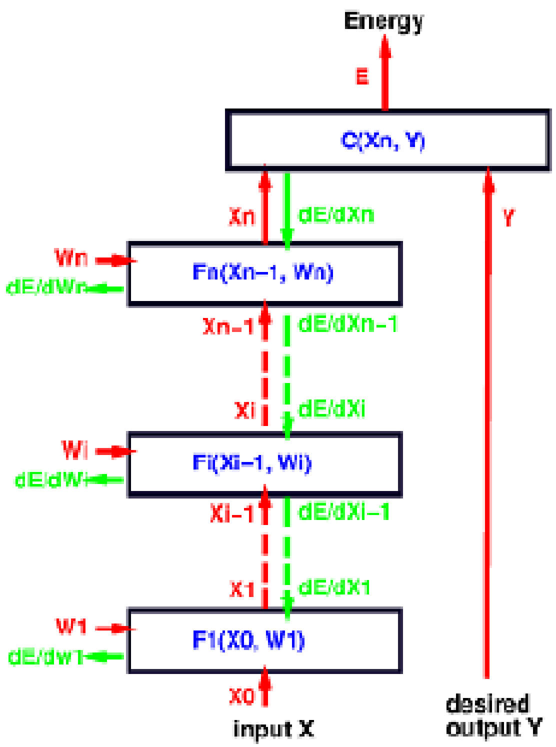
  - 对NAG的链式求导：



$$\frac{\partial z}{\partial x} = \sum_{i=1}^{n} \frac{\partial z}{\partial y_i}\frac{\partial y_i}{\partial x}$$

卷积神经网络的训练方法

- **Back Propagation**
  - 1974年Webos在博士论文中首次提出BP算法，但未引发关注
  - 目前广泛使用的BP算法诞生于1986年
  - 以全连接层为例：链式求导，梯度反向传导



- $\dfrac{\partial E}{\partial X_n} = \dfrac{\partial C(X_n, Y)}{\partial X_n}$
- $\dfrac{\partial E}{\partial X_{n-1}} = \dfrac{\partial E}{\partial X_n} \dfrac{\partial F_n(X_{n-1}, W_n)}{\partial X_{n-1}}$
- $\dfrac{\partial E}{\partial W_n} = \dfrac{\partial E}{\partial X_n} \dfrac{\partial F_n(X_{n-1}, W_n)}{\partial W_n}$
- $\dfrac{\partial E}{\partial X_{n-2}} = \dfrac{\partial E}{\partial X_{n-1}} \dfrac{\partial F_{n-1}(X_{n-2}, W_{n-1})}{\partial X_{n-2}}$
- $\dfrac{\partial E}{\partial W_{n-1}} = \dfrac{\partial E}{\partial X_{n-1}} \dfrac{\partial F_{n-1}(X_{n-2}, W_{n-1})}{\partial W_{n-1}}$
- ....etc, until we reach the first module.
- we now have all the $\frac{\partial E}{\partial W_k}$ for $k \in [1, n]$.

卷积神经网络的训练方法

Gradient Descent and its Variant

- Gradient Descend：利用所有样本计算梯度

$$w_{t+1} = w_t - \eta_t \nabla_w L(w)$$

速度慢！大数据内存不足！

- Stochastic Gradient Descend：随机选择单个样本

$$w_{t+1} = w_t - \eta_t \nabla_w L(w, x_i, y_i)$$

方差大！损失函数震荡严重！

- Mini-batch SGD: 对随机mini-batch计算梯度，进行参数更新

$$w_{t+1} = w_t - \frac{1}{N}\eta_t \sum_{i=1}^{N} \nabla_w L(w_t, x_i, y_i)$$

Mini-batch SGD:
for $i = 1$: Num_Iterations
1. 随机采样一个Mini-batch的样本
2. 前向操作：计算损失
3. 后向操作：通过BP算法计算梯度
4. 网络更新：利用步骤3）计算的梯度更新网络参数

end

卷积神经网络的训练方法

- **Mini-batch SGD**
  - Weight Decay
    - 避免过拟合，二范数较常用，一般 λ 设置的较小，例如 0.0005

$$\tilde{L}(w) = L(w) + \frac{\lambda}{2}\|w\|_2^2$$

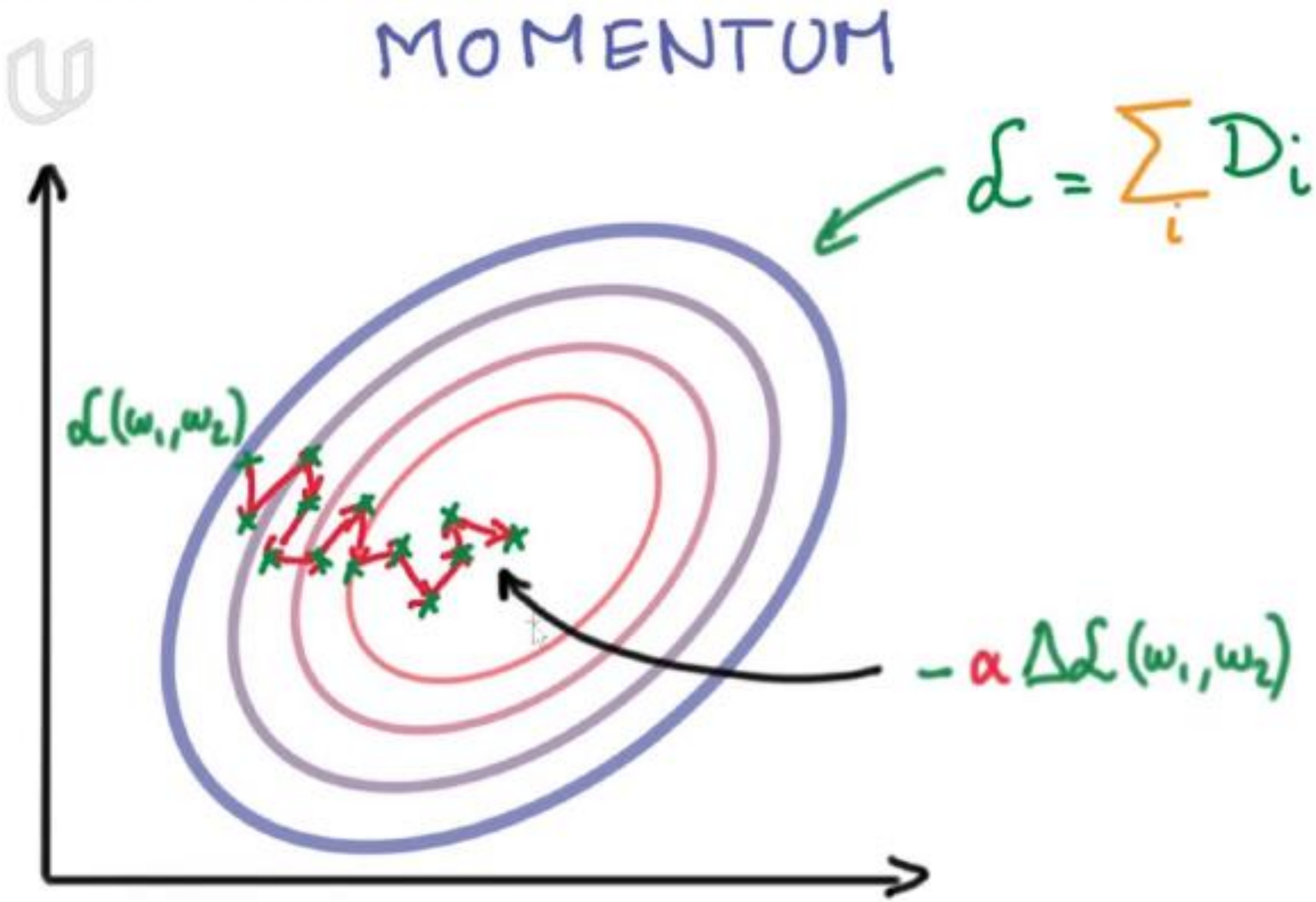
  - Momentum
    - 计算梯度时考虑历史梯度信息 $\mu = 0.9$

$$v_{t+1} = \mu v_t - \eta_t \nabla L(w_t)$$
$$w_{t+1} = w_t + v_{t+1}$$

    - 使随机梯度下降更容易跳出局部最优，加速收敛

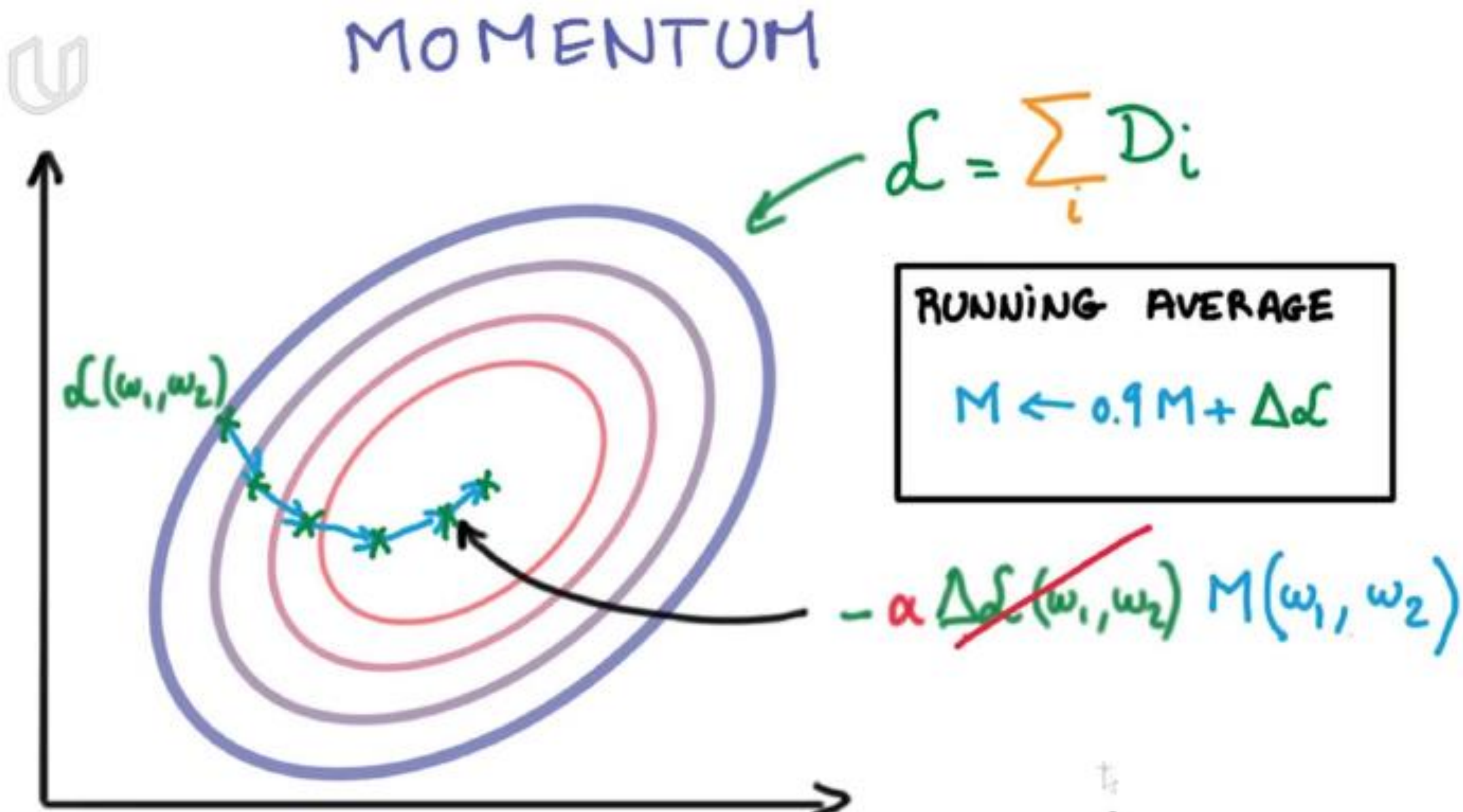
卷积神经网络的训练方法

- **Mini-batch SGD**
  - Without Momentum



卷积神经网络的训练方法

- **Mini-batch SGD**
  - With Momentum



卷积神经网络的训练方法

■ SGD的各种扩展

- AdaGrad

根据历史梯度信息决定当前batch的leaning rate

$$(W_{t+1})_i = (W_t)_i - \alpha \frac{(\nabla L(W_t))_i}{\sqrt{\sum_{t'=1}^{t} (\nabla L(W_{t'}))_i^2}}$$

- AdaDelta

$$(v_t)_i = \frac{\mathrm{RMS}((v_{t-1})_i)}{\mathrm{RMS}\,(\nabla L(W_t))_i} (\nabla L(W_{t'}))_i$$

$$\mathrm{RMS}\,(\nabla L(W_t))_i = \sqrt{E[g^2] + \varepsilon}$$

$$E[g^2]_t = \delta E[g^2]_{t-1} + (1-\delta) g_t^2$$

更新策略：

$$(W_{t+1})_i = (W_t)_i - \alpha (v_t)_i.$$

卷积神经网络的训练方法

- **Learning Rate**

$$\omega \leftarrow \omega - \eta \frac{\partial E}{\partial \omega}$$

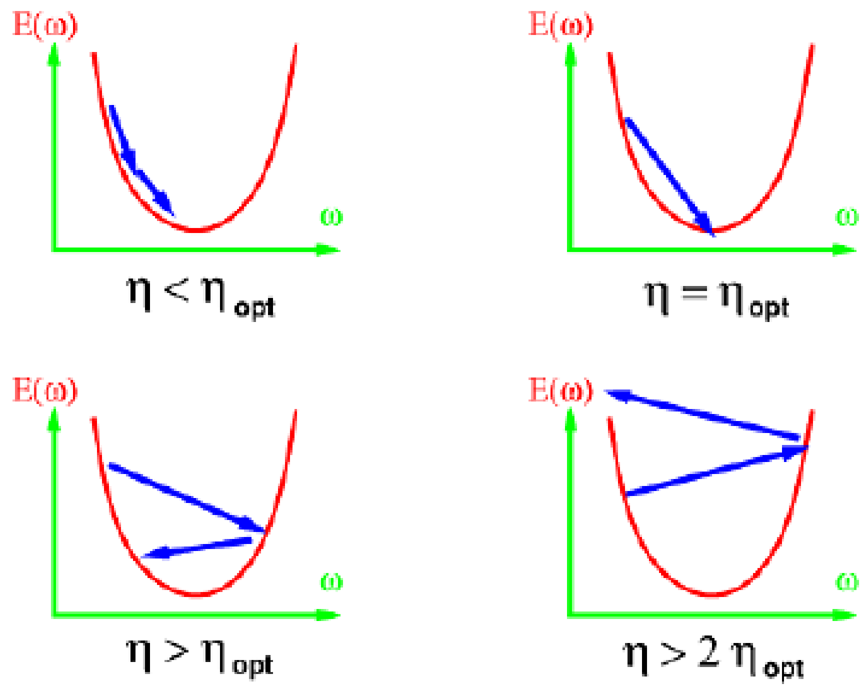
gradient of objective function

weight vector

learning rate

- Batch Gradient
- There is an optimal learning rate
- Equal to inverse 2nd derivative

$$\boxed{\eta_{opt} = \left(\frac{\partial^2 E}{\partial \omega^2} \right)^{-1}}$$



卷积神经网络的训练方法

- **Learning Rate Policy（以Caffe为例）**
  - Fixed
    - Learning Rate固定不变
  - Step
    - Learning Rate在每隔stepsize轮迭代后减少gamma倍
  - Polynomial
    - Learning Rate依多项式曲线下降

$$LR(t) = \text{base\_lr} \times (1 - \frac{t}{T})^{power}$$

  - Inv
    - Learning Rate随迭代次数增加而下降

$$LR(t) = \text{base\_lr} \times (1 + \text{gamma} * \text{iter})^{-power}$$

Solver.prototxt

```
base_lr: 0.01
lr_policy: "fixed"
```

```
base_lr: 0.01
lr_policy: "step"
gamma: 0.1
stepsize: 100000
```

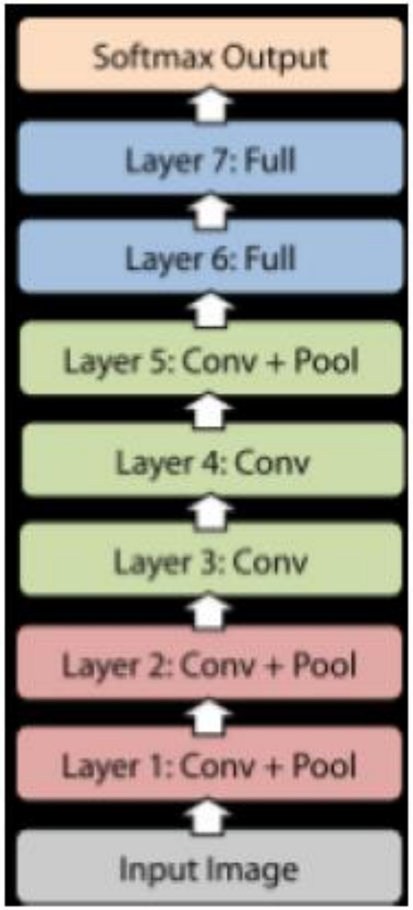
```
base_lr: 0.01
lr_policy: "poly"
power: 0.5
```

```
base_lr: 0.01
lr_policy: "inv"
gamma: 0.0001
power: 0.75
```

卷积神经网络的实例

Alexnet

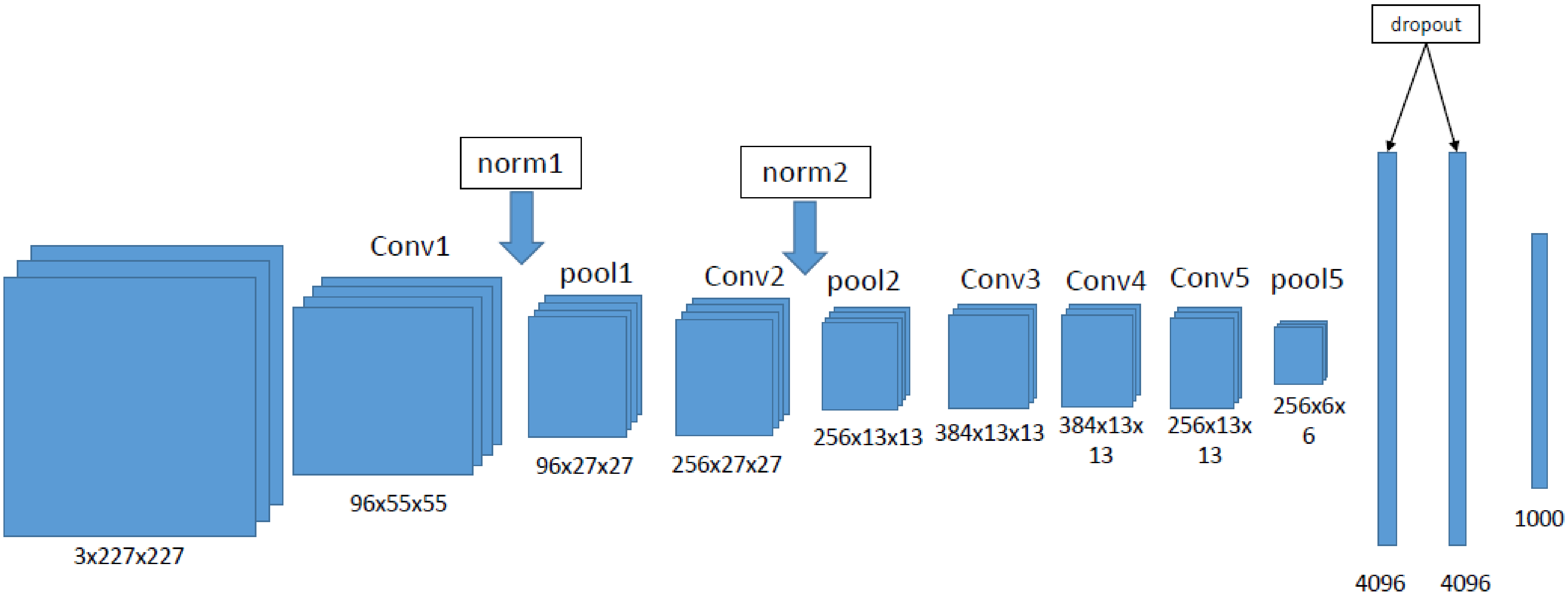
- 网络基本结构



[1] 2012-NIPS-ImageNet Classification with Deep ConvolutionalNeural Networks

卷积神经网络的实例

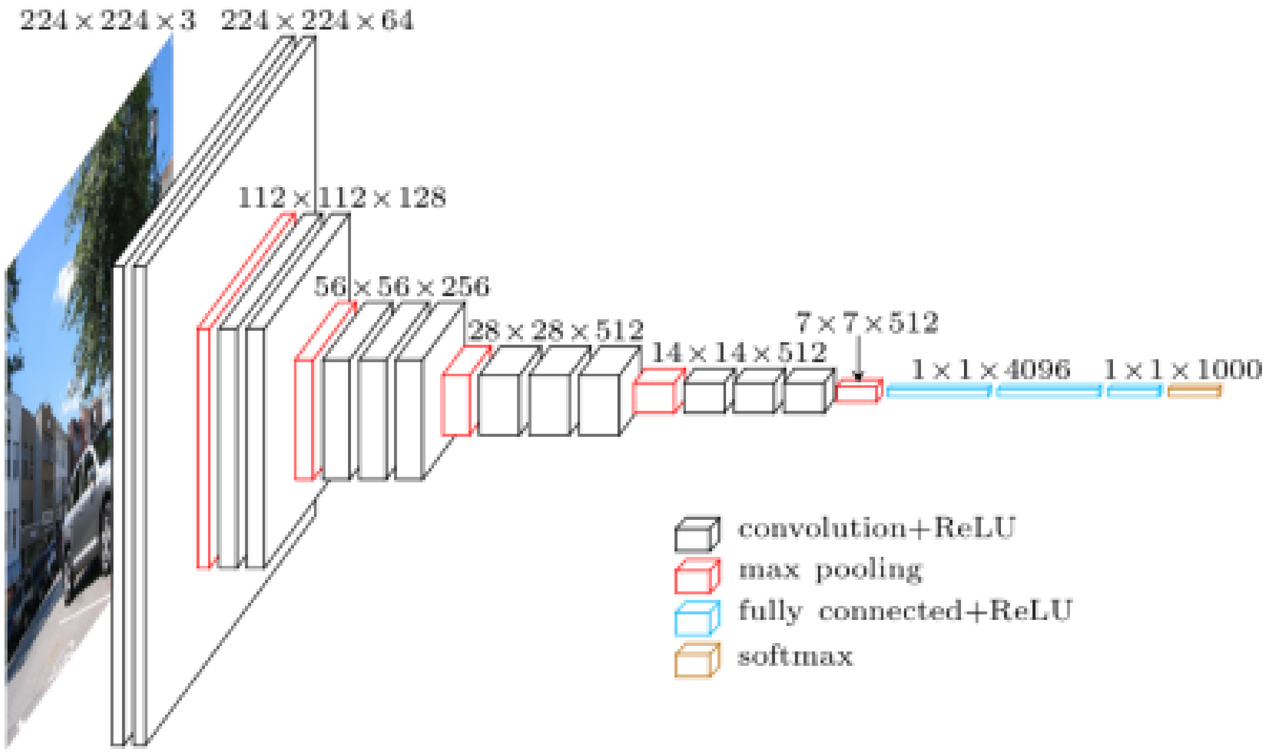
Alexnet



卷积神经网络的实例

VGG16

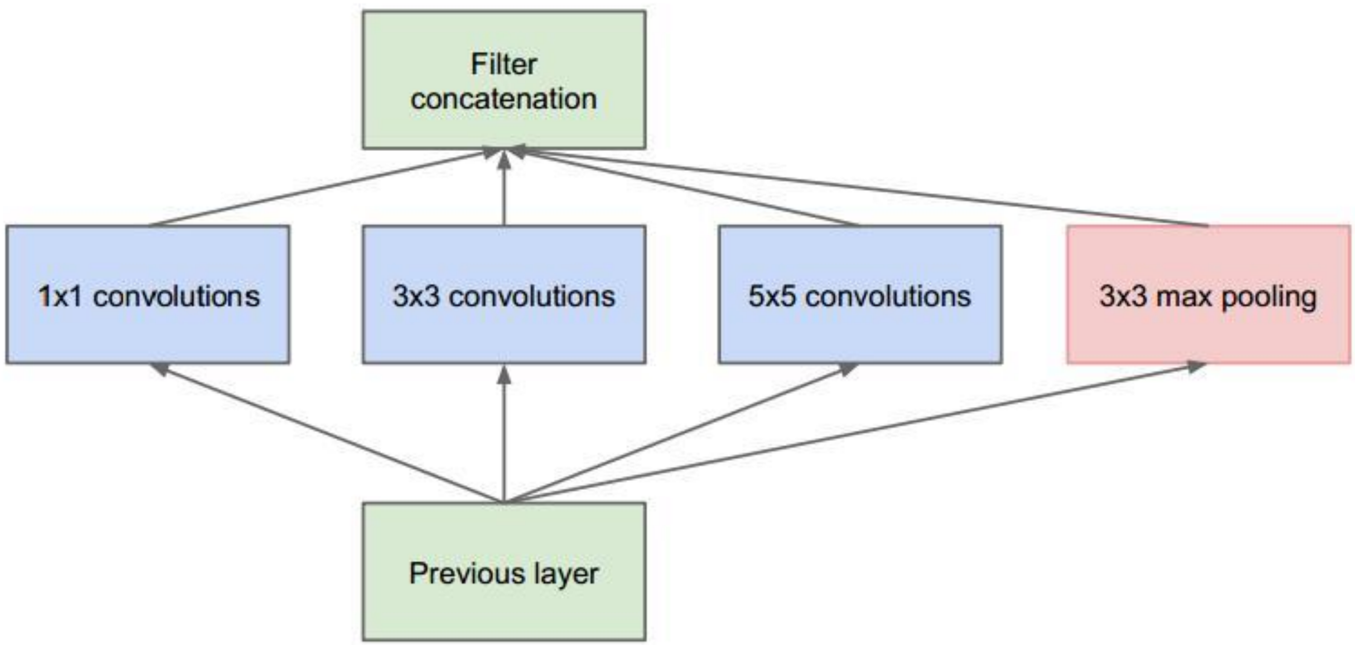
- 使用多个较小的卷积核代替较大的卷积
  - 减少参数
  - 增加非线性拟合能力



卷积神经网络的实例

Inception-GoogleNet

- 采用不同大小的卷积核意味着不同大小的感受野，最后拼接意味着不同尺度特征的融合减少参数。
- 网络越到后面，特征越抽象，而且每个特征所涉及的 … 卷

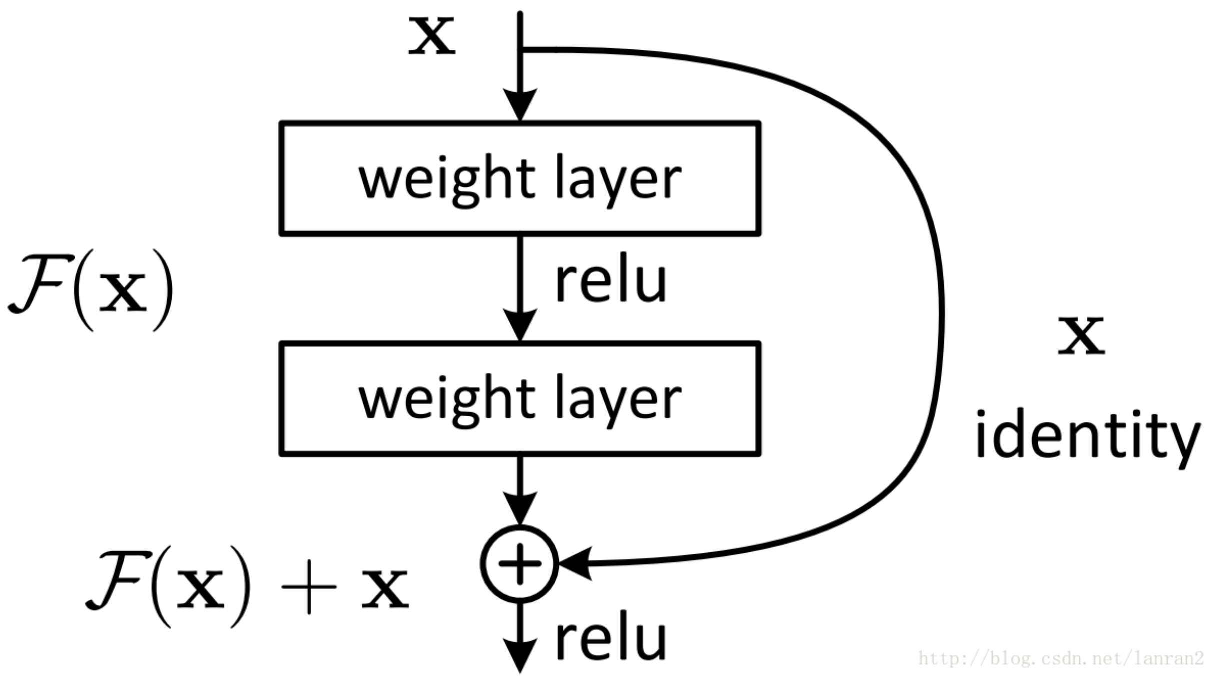


(a) Inception module, naïve version

卷积神经网络的实例

Skip-connection - ResNet

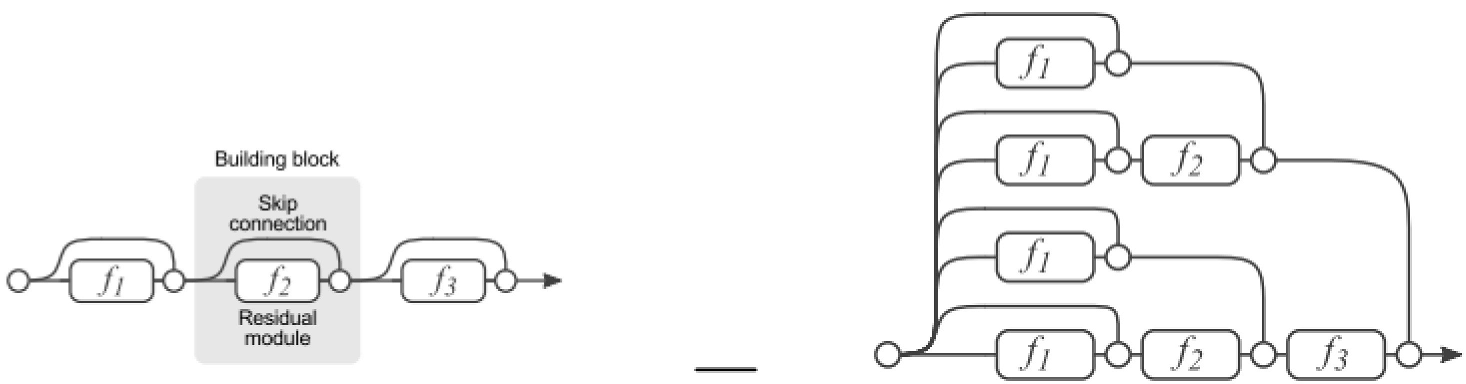
- 网络越深，梯度消失的现象就越来越明显，网络的训练效果也不会很好。



卷积神经网络的实例

Skip-connection - ResNet

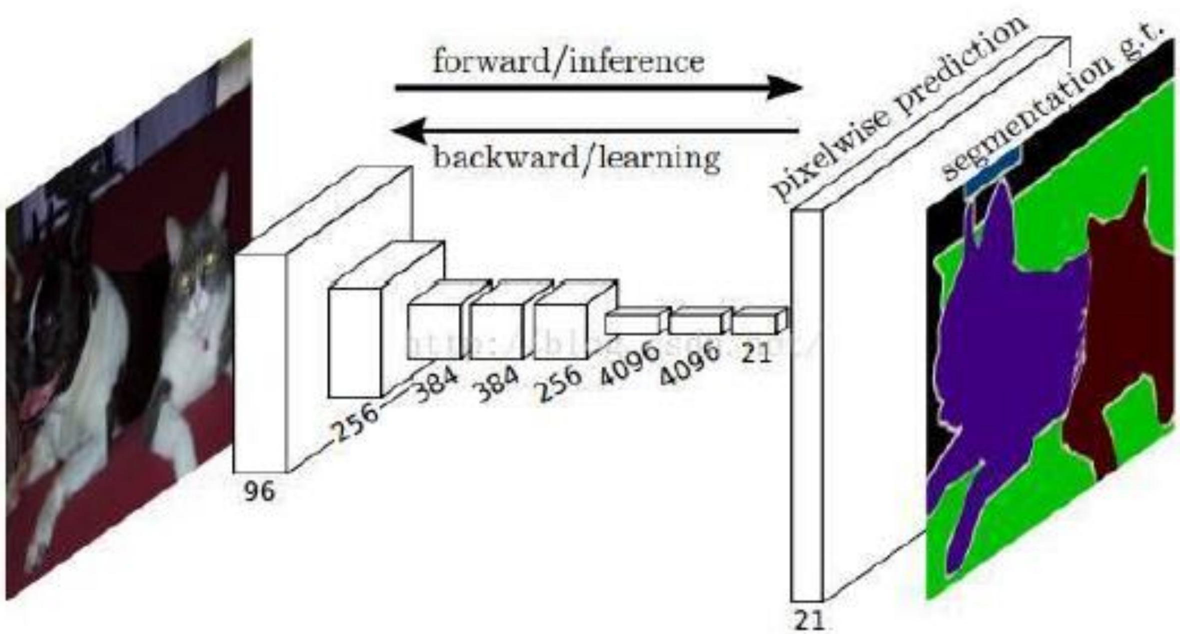
- 网络越深，梯度消失的现象就越来越明显，网络的训练效果也不会很好。
- 残差网络可以理解成由多种路径组合的一个网络，是很多并行子网络的组合。整个残差网络其实相当于一个多人投票系统（Ensemble System）。



卷积神经网络的实例

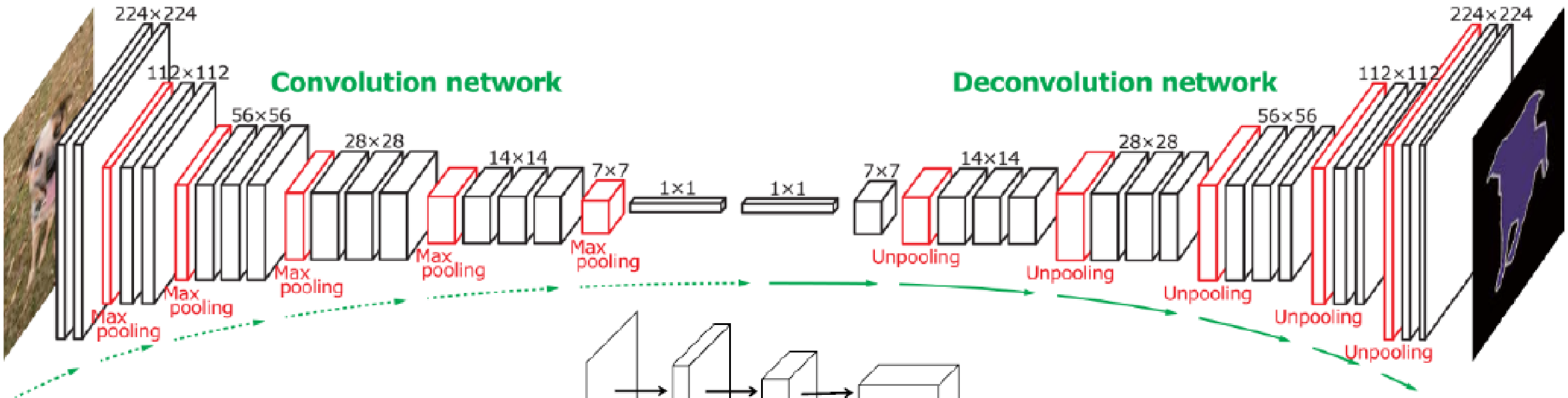
FCN网络

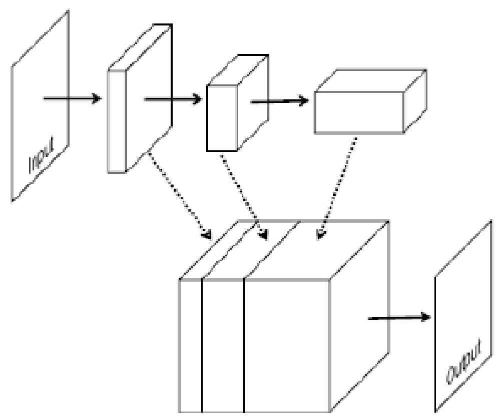
- 网络基本架构
  - 全连接层->卷积层
  - 1000维向量->原图大小



卷积神经网络的实例

FCN网络

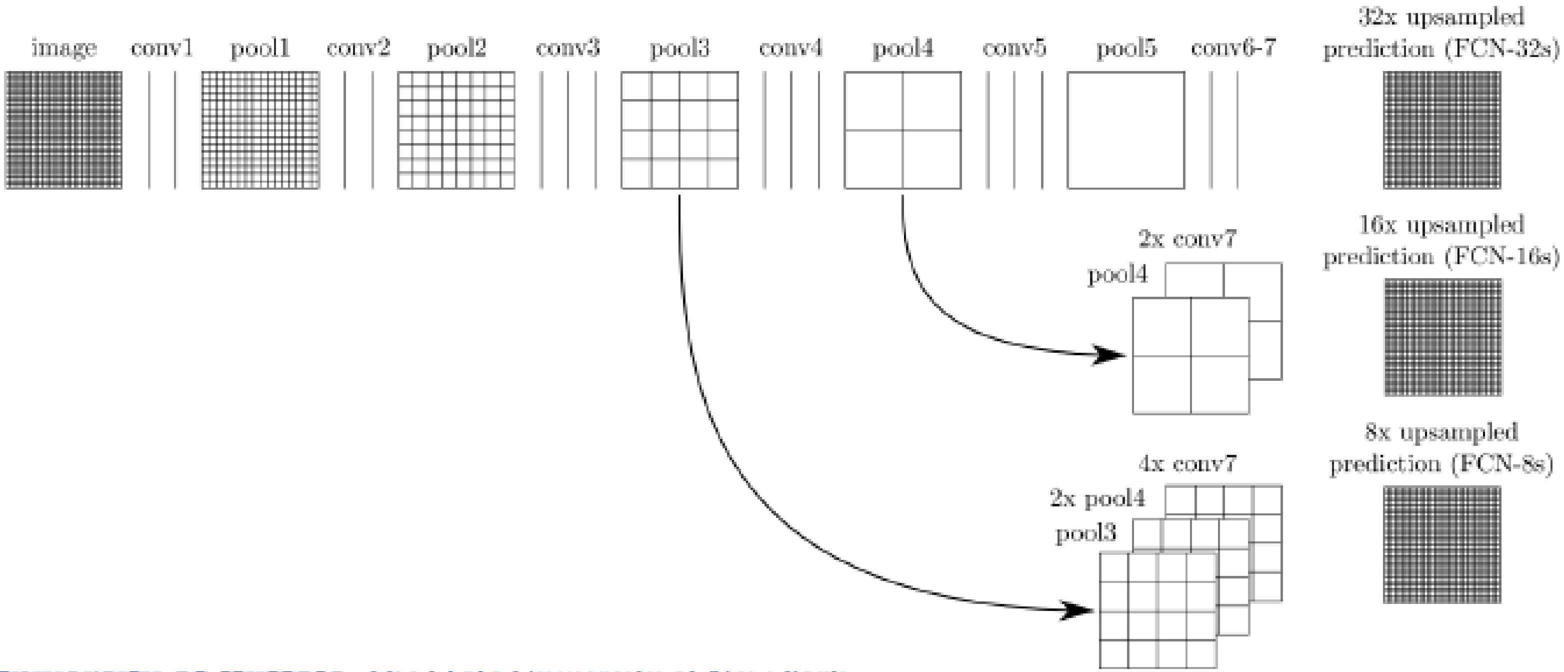




卷积神经网络的实例

FCN网络

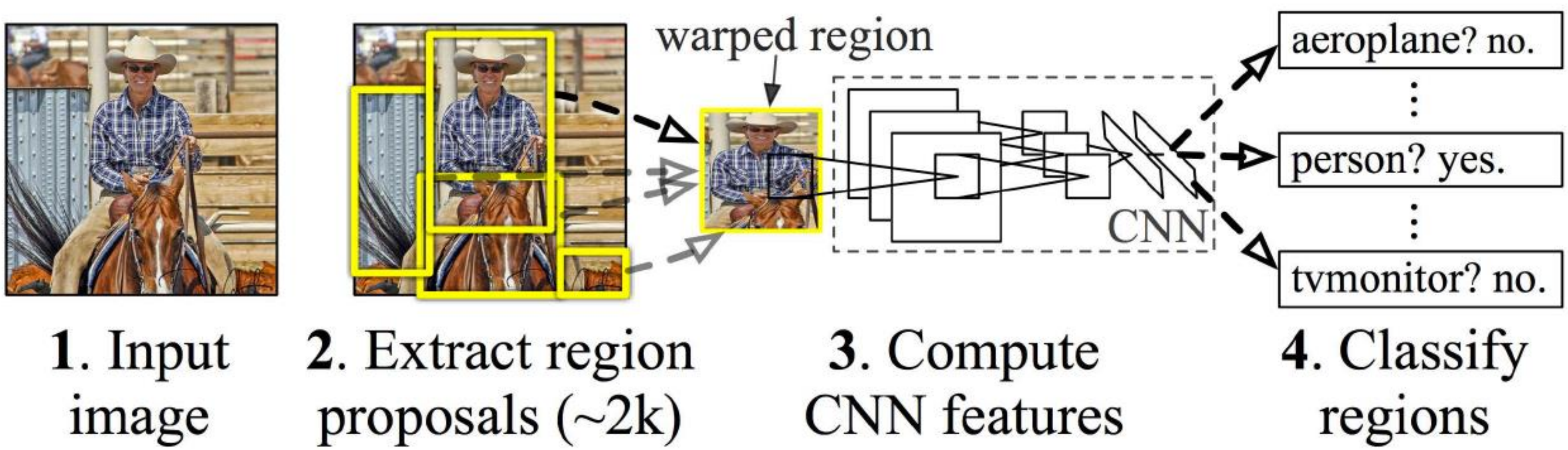
- Deconv
  - 简单上采样结果很差
  - 可以看成引入了高层信息
  - 局部信息有所丢失



卷积神经网络的实例

R-CNN网络

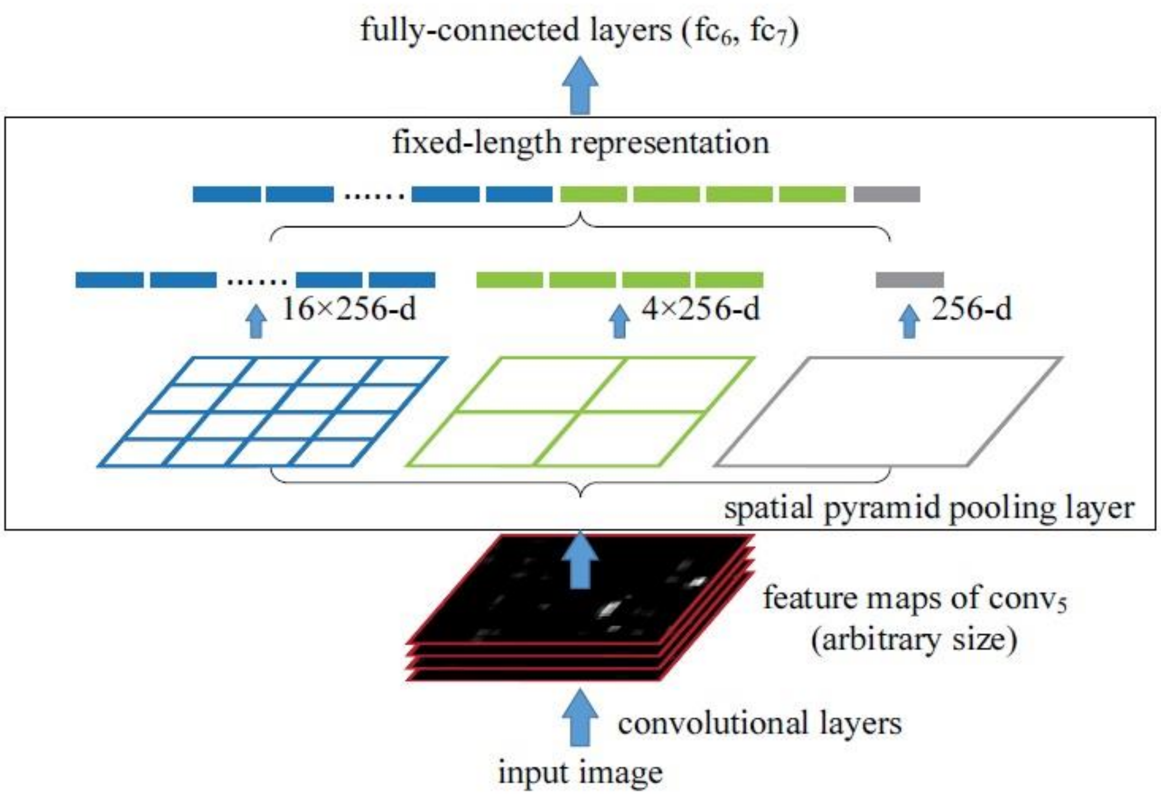
- 对于输入图片，运用Selective Search 提取大约2000 个候选区域；
- 对这些候选区域分别用预训练的AlexNet 或VGG16 模型提取特征；
- 将提取到的特征输入SVM 分类器进行分类和边框回归。



卷积神经网络的实例

SPP-Net

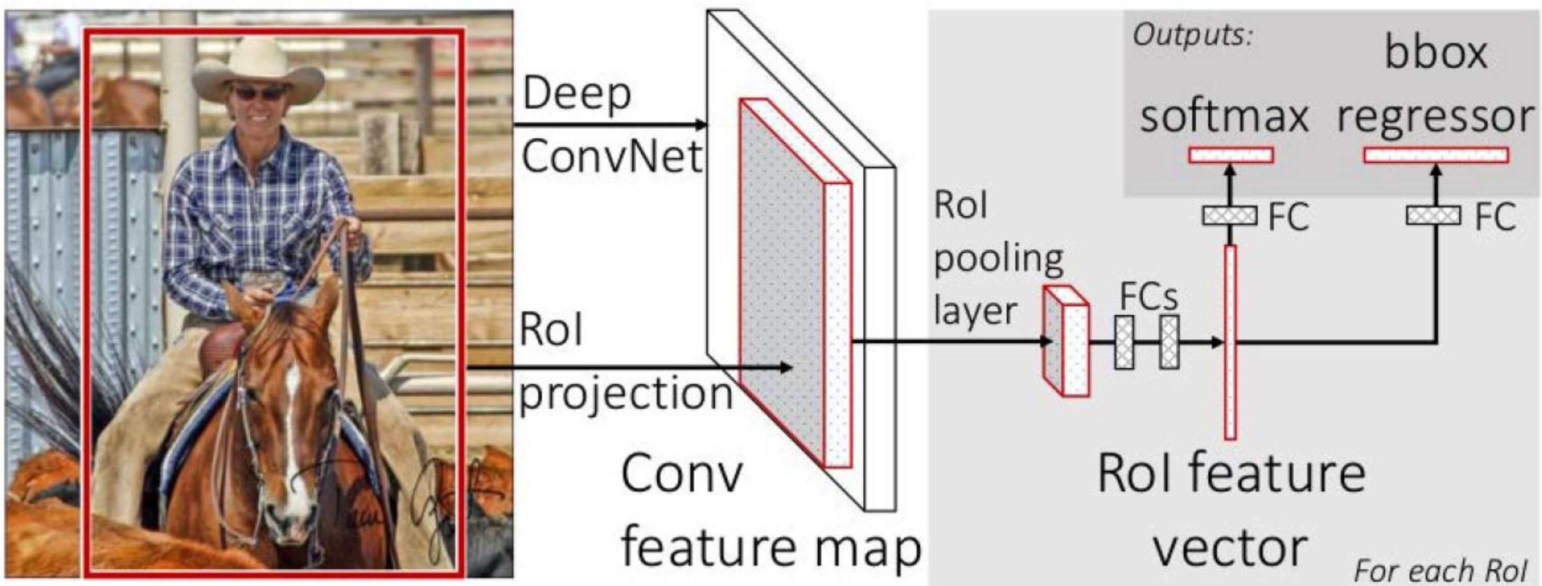
- 对于输入图片，运用Selective Search 提取大约2000 个候选区域；
- 对这些候选区域分别用预训练的AlexNet 或VGG16 模型提取特征；



卷积神经网络的实例

Fast R-CNN网络

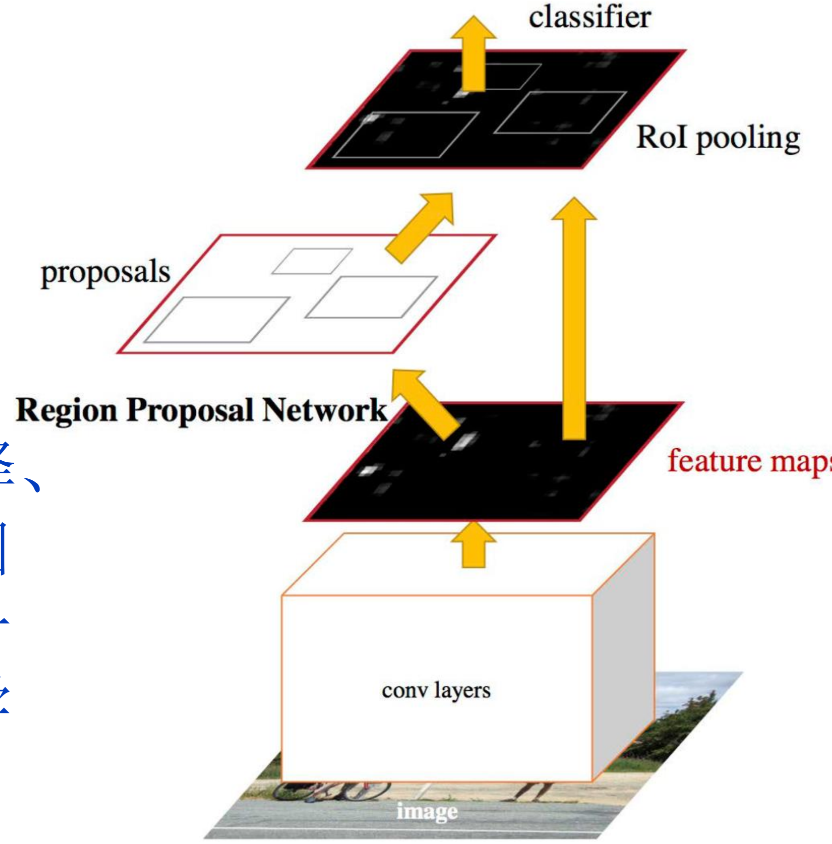
- End to End，用softmax层取代了SVM分类器
- 多任务学习框架，同时完成bounding box边界回归任务和分类任务



卷积神经网络的实例

Faster R-CNN网络

- RPN + Fast R-CNN
- 两个任务共享网络前端的部分卷积层来进行特征提取
- “锚点”机制（anchor 机制）

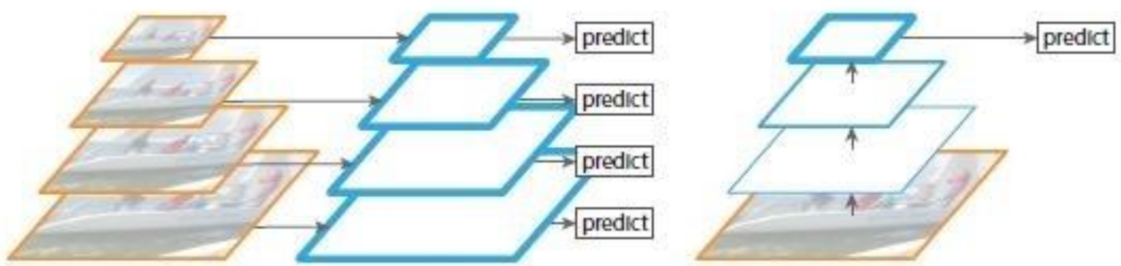


Faster R-CNN 将候选区域的选择、特征提取、分类器分类和边框回归都整合到了一个框架中，是一个真正意义上的端到端的深度学习目标检测框架。

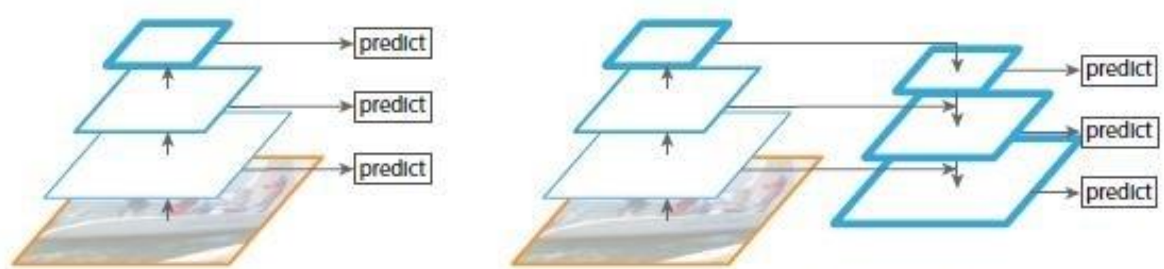
卷积神经网络的实例

FPN网络

- 特征金字塔结构
- 高层feature叠加到多个featureMap上进行预测
- 低层的思想，Top-Down and Bottom-Up

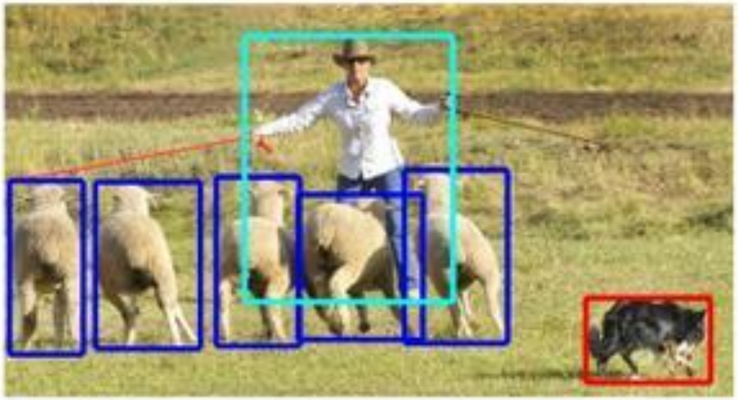


(a) Featurized image pyramid　　(b) Single feature map



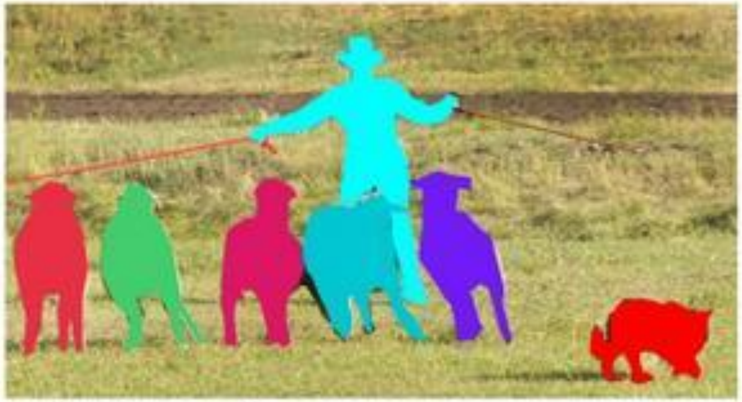
卷积神经网络的实例

(a) Image classification



(b) Object localization

(c) Semantic segmentation



(d) This work segment individual object instances

卷积神经网络的实例

Mask R-CNN网络

- FPN + Res-Net，检测+分割
- RoIAlign 层的加入，对 feature map 的插值
- softmax的多项式交叉熵替换成sigmod二值交叉熵