人工神经网络（深度学习）

王 瑞 轩

http://isee.sysu.edu.cn/~wangruixuan/

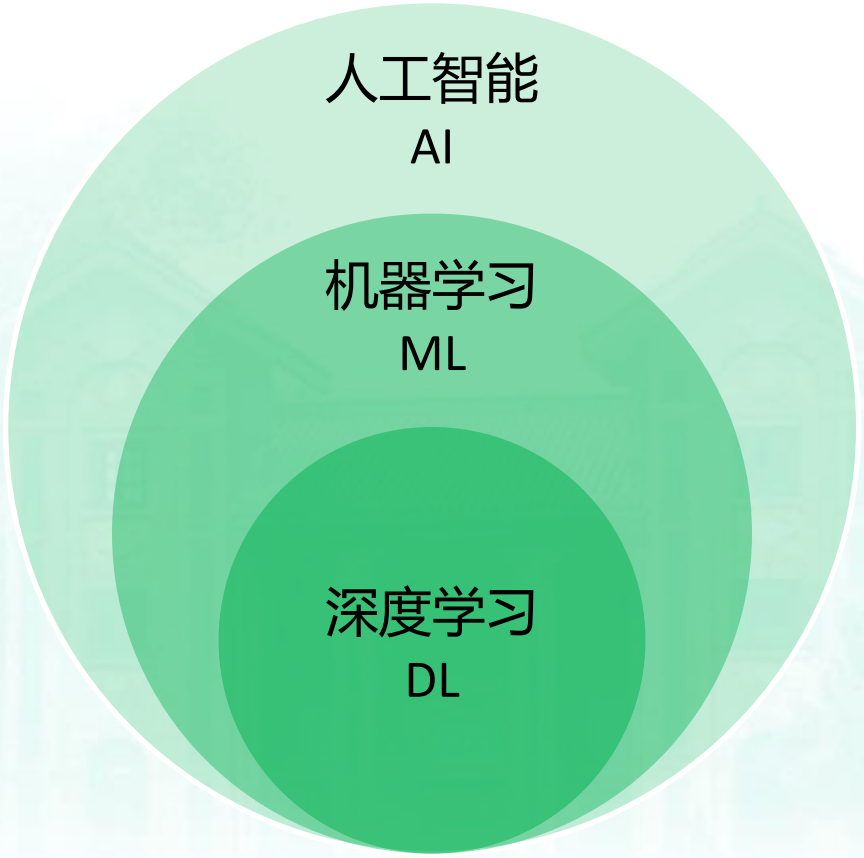
SUN YAT-SEN University

机器智能与先进计算
教育部重点实验室

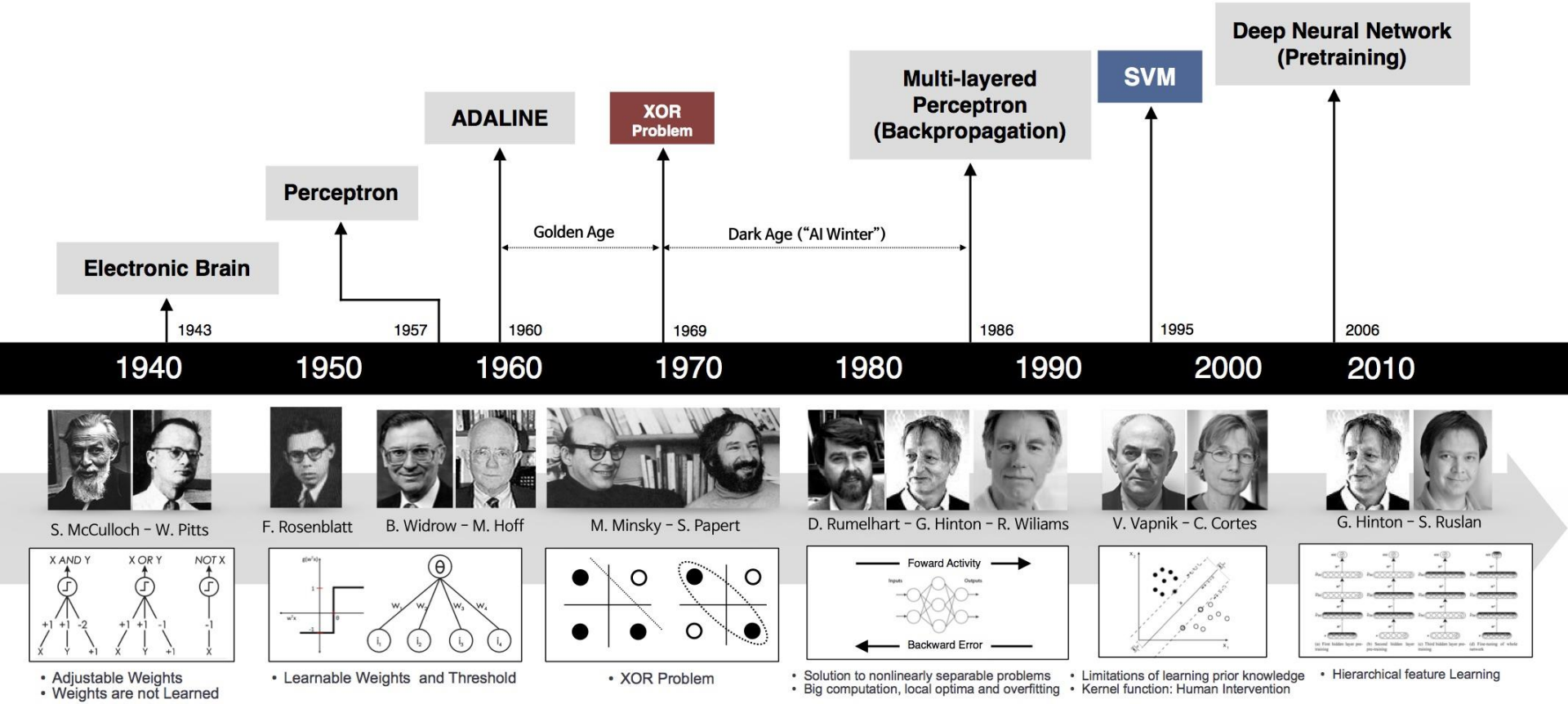
深度学习概览

- What: 揭神秘面纱
- Wow: 赏群模乱舞
- Why: 寻万能之源
- Where: 追研究前沿
- Whoops: 探未解之谜
- While: 评大众观点

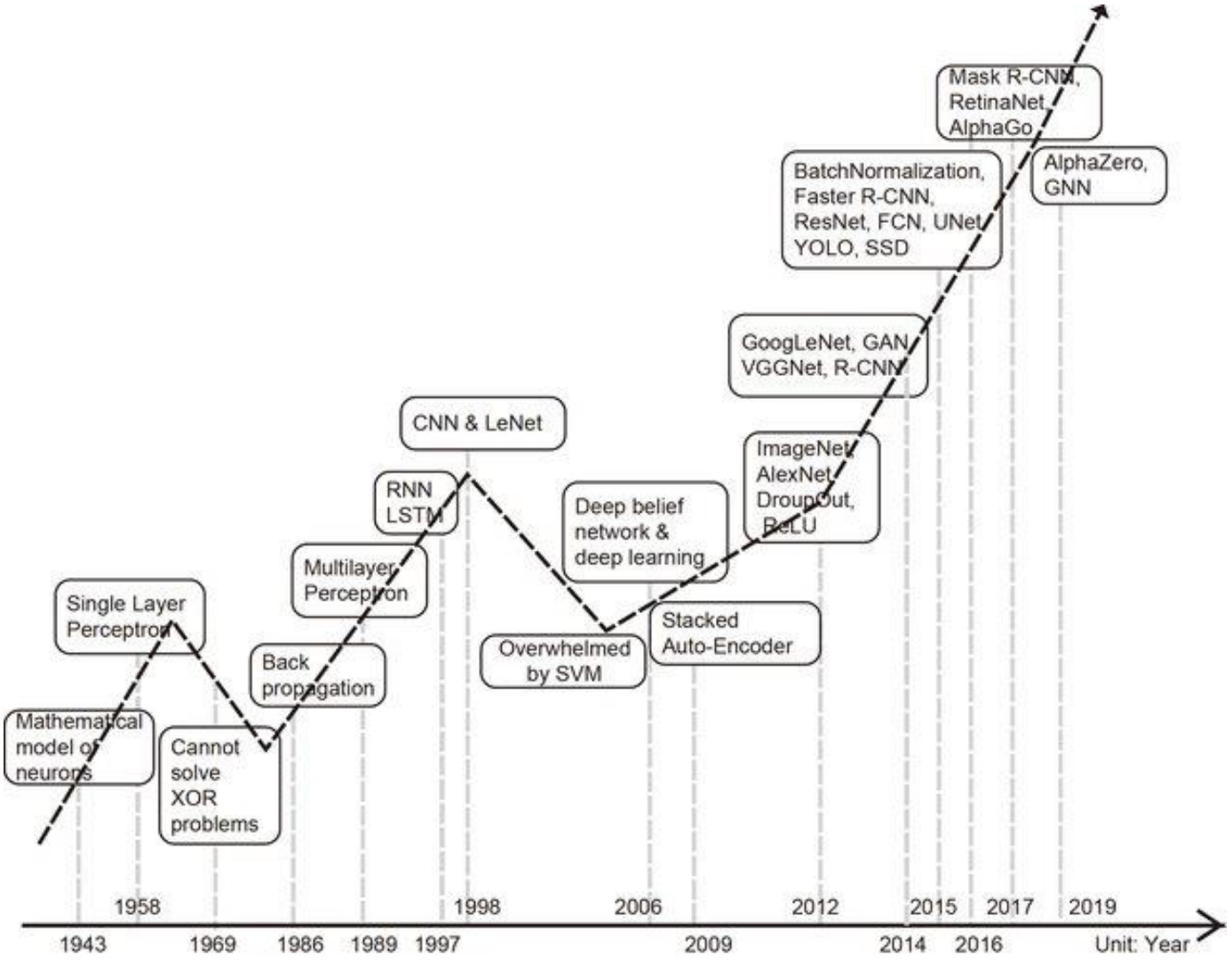
揭神秘面纱:深度学习与AI的关系



揭神秘面纱:深度学习发展历史



揭神秘面纱:深度学习发展历史



揭神秘面纱:深度学习研究的坚守者

- 选定人生方向，坚持走下去!
- 要尊重他人的付出与成果!

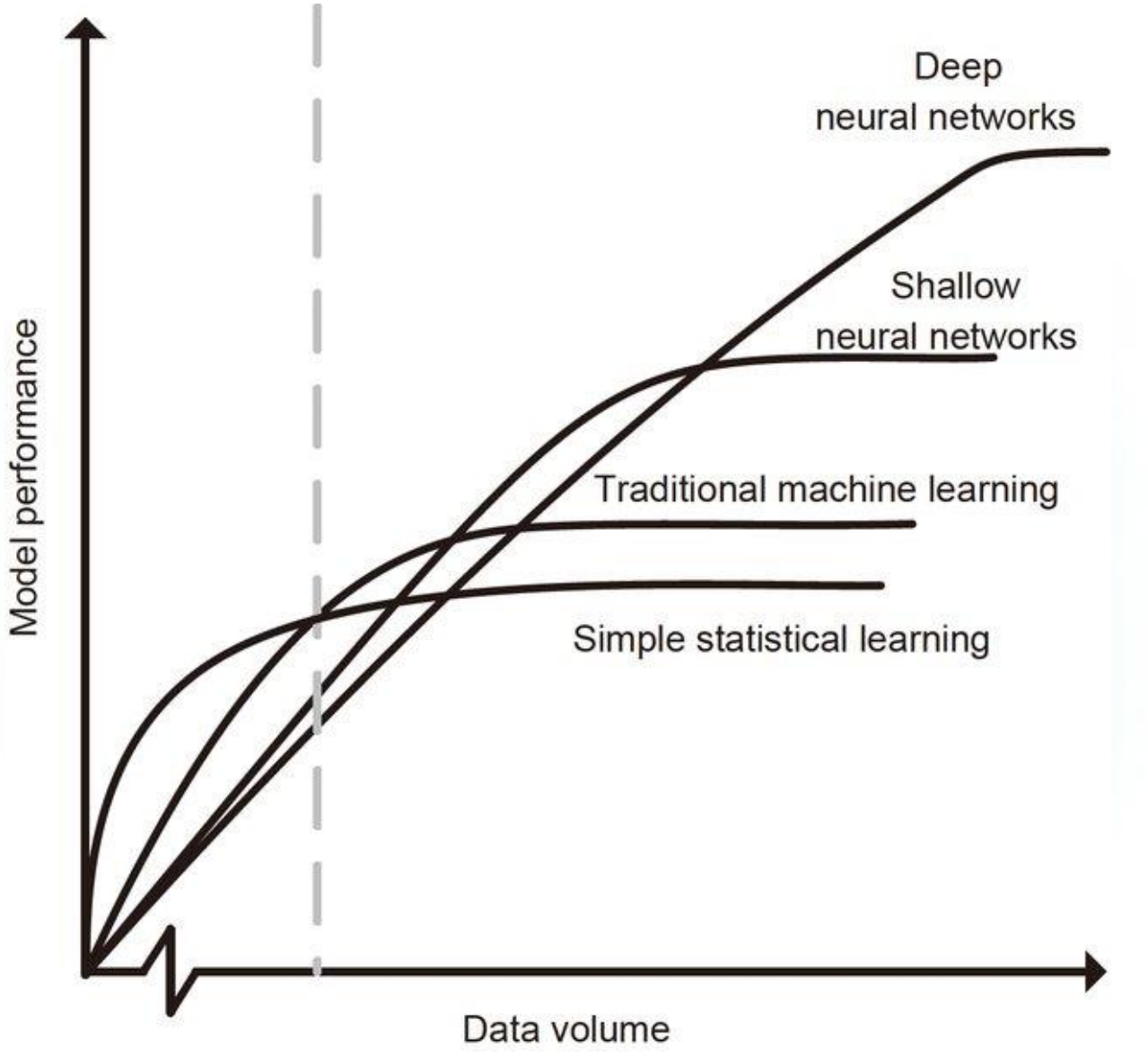
Yoshua Bengio　Geoffrey Hinton　Yann LeCun

Turing Award 2019

Jürgen Schmidhuber

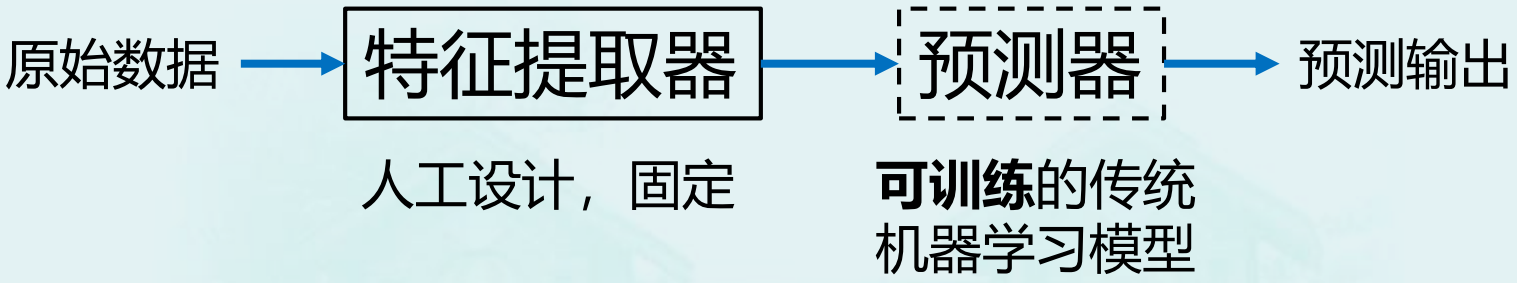
LSTM, 1997

揭神秘面纱:与传统方法的比较

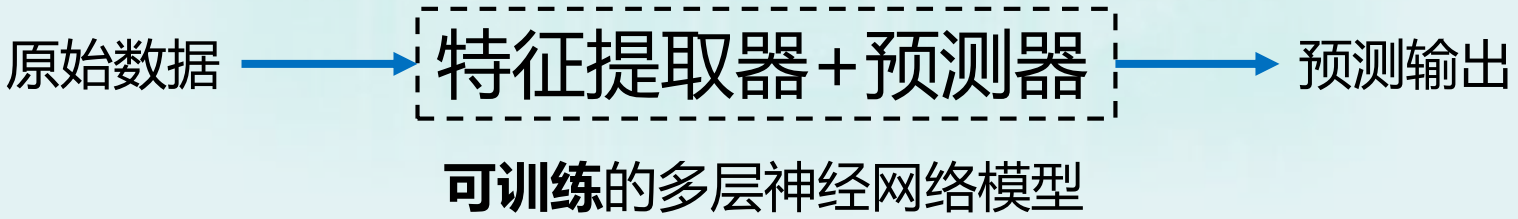


揭神秘面纱：与传统方法的比较

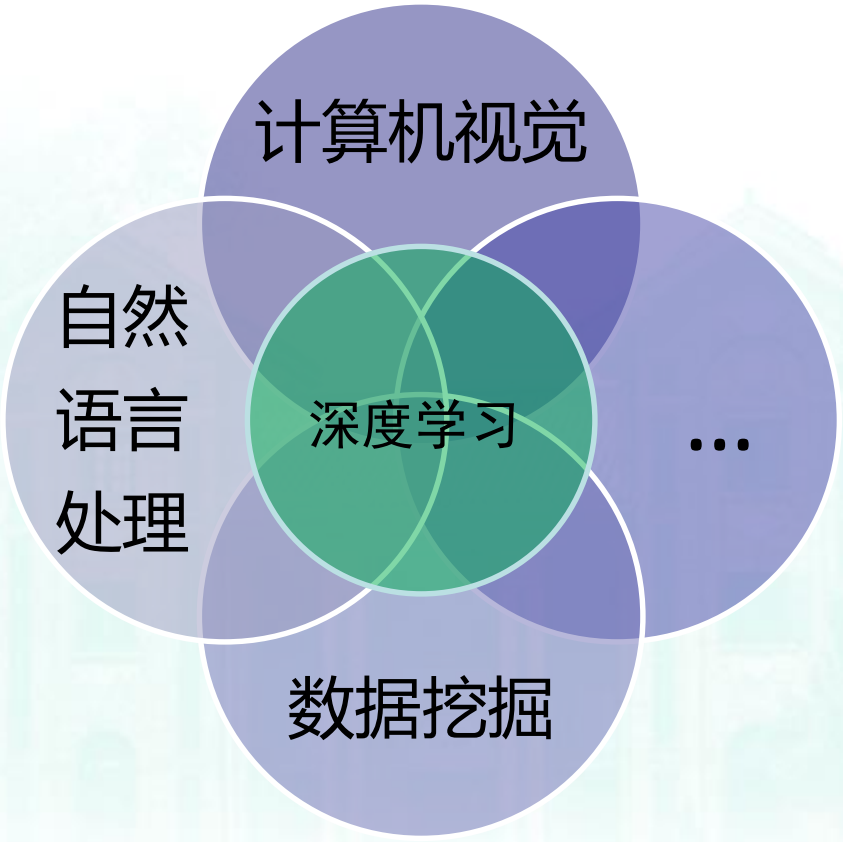
传统机器学习



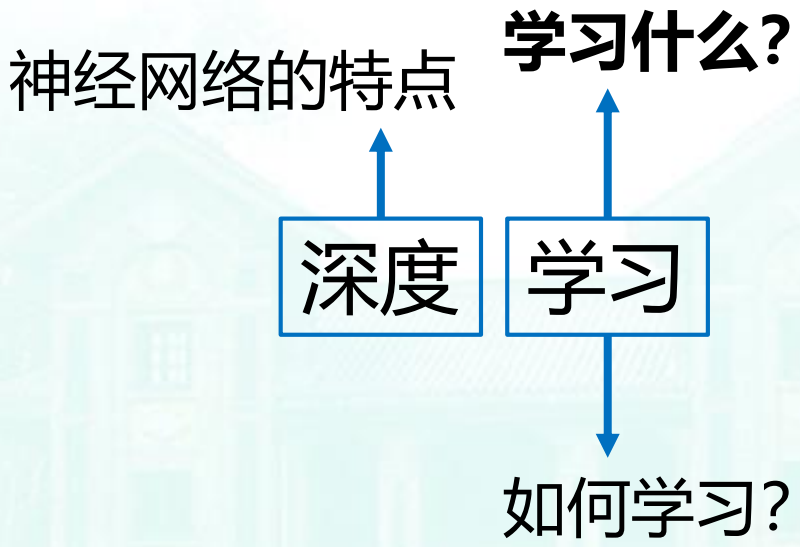
深度学习



揭神秘面纱:深度学习的应用



揭神秘面纱



揭神秘面纱:要解决之任务（示例）

- 回归(Regression)任务：“通过身高预测体重”

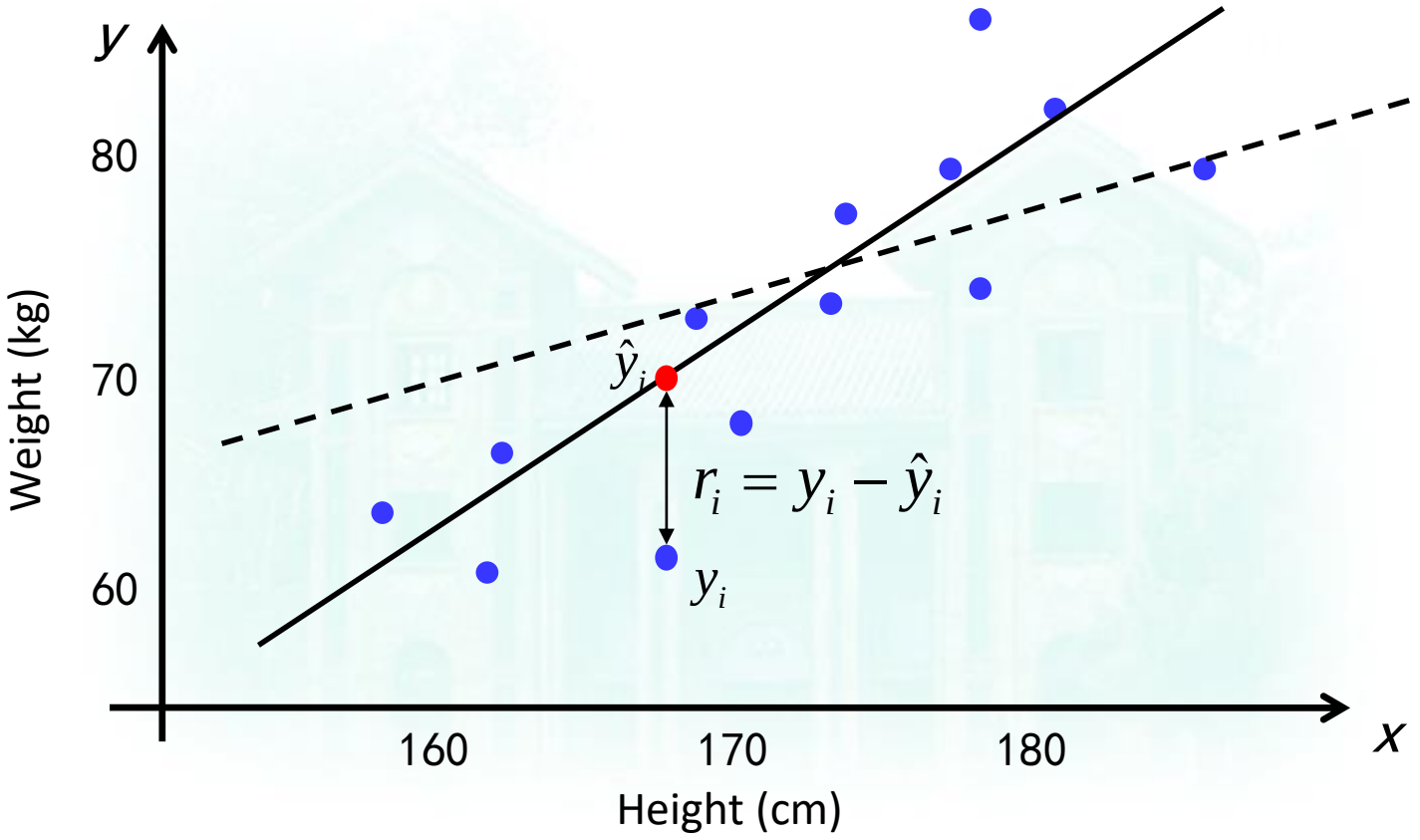
第一步：建模 $y = f(x) = wx + b$

第二步：收集数据 $D = \{(x_i, y_i)\}$

第三步：利用D寻找模型最佳参数 $\boldsymbol{\theta}^* = \{w^*, b^*\}$

揭神秘面纱:要解决之任务（示例）

- 回归(Regression)任务：“通过身高预测体重”



原则：希望预测的体重与观测的体重越接近越好！

揭神秘面纱:要解决之任务（示例）

- 换句话说：希望模型（函数）的预测误差尽量的小

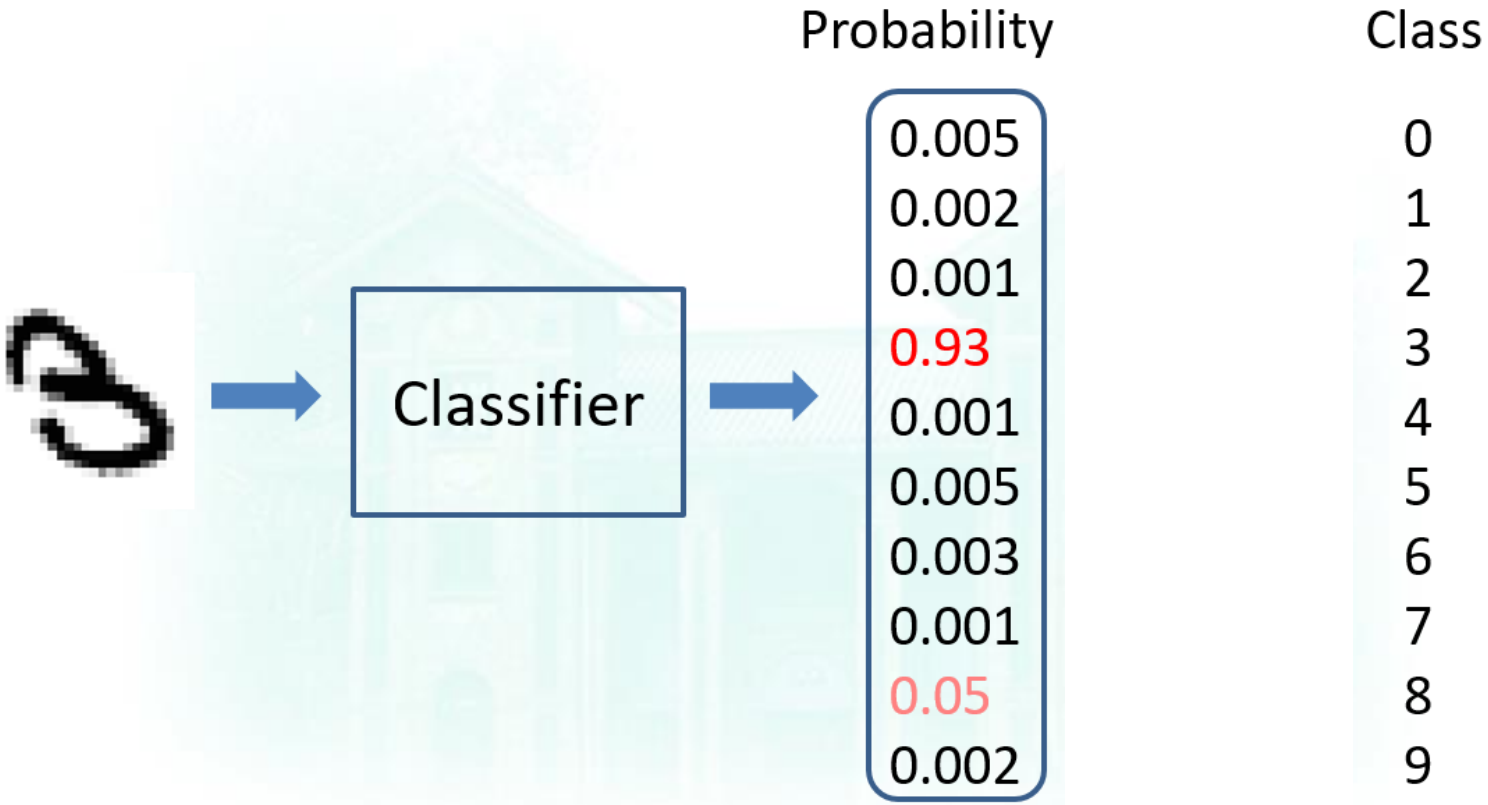
$$\begin{aligned}\min_{\boldsymbol{\theta}} \quad L(\boldsymbol{\theta}) &= \frac{1}{N}\sum_{i=1}^{N} r_i^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}(y_i - wx_i - b)^2\end{aligned}$$

寻找模型（函数）最佳参数 = 最小化损失函数$L(\boldsymbol{\theta})$

学习什么：模型的最佳参数！而最佳模型表示的是...
身高（**输入**）与体重（**输出**）之间的准确**关系**！

揭神秘面纱:要解决之任务（示例）

- 分类（Classification）任务：“手写体数字识别”



给一个输入数据（的表示），模型预测该数据属于每一类的概率

希望模型的输出与理想输出越接近越好！

揭神秘面纱:要解决之任务（示例）

- 模型的输出与理想输出各是一个离散概率分布

模型输出：$\hat{\mathbf{y}}_i = \mathbf{f}(\mathbf{x}_i; \boldsymbol{\theta}) = (\hat{y}_{i1}, \hat{y}_{i2}, \ldots, \hat{y}_{iK})$

理想输出：$\mathbf{y}_i = (y_{i1}, \ldots, y_{iK}) = (0, \ldots, 1, \ldots, 0)$

- 如何测量两个概率分布的差别？Cross-entropy loss!

$$
\begin{aligned}
l(\mathbf{y}_i, \mathbf{f}(\mathbf{x}_i; \boldsymbol{\theta})) &= \sum_{k=1}^{K} y_{ik} \log \frac{1}{\hat{y}_{ik}} \\
&= -\sum_{k=1}^{K} y_{ik} \log \hat{y}_{ik}
\end{aligned}
$$

揭神秘面纱:要解决之任务（示例）

- 与回归任务的目标类似：

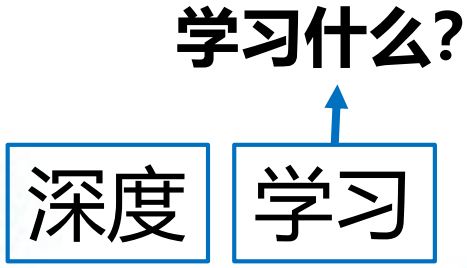
$$\min_{\boldsymbol{\theta}} \quad L(\boldsymbol{\theta}) \quad = \quad \frac{1}{N}\sum_{i=1}^{N} l(\mathbf{y}_i, \mathbf{f}(\mathbf{x}_i; \boldsymbol{\theta}))$$

$$= \quad -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik} \log \hat{y}_{ik}$$

寻找模型（函数）最佳参数 = 最小化损失函数$L(\boldsymbol{\theta})$

学习什么：模型的最佳参数！而最佳模型表示的是...
图象（**输入**）与数字类别（**输出**）之间的准确**关系**！

揭神秘面纱：学习什么



学习的是系统输入与输出的关系！

如果系统由一个数学函数表示，
学习的是函数（数学模型）的最佳参数！

揭神秘面纱：模型/函数

如何设计模型（函数）？

$$\min_{\boldsymbol{\theta}} \quad L(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} l(\mathbf{y}_i, \mathbf{f}(\mathbf{x}_i;\boldsymbol{\theta}))$$

人工神经网络！

$$= -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik}\log \hat{y}_{ik}$$

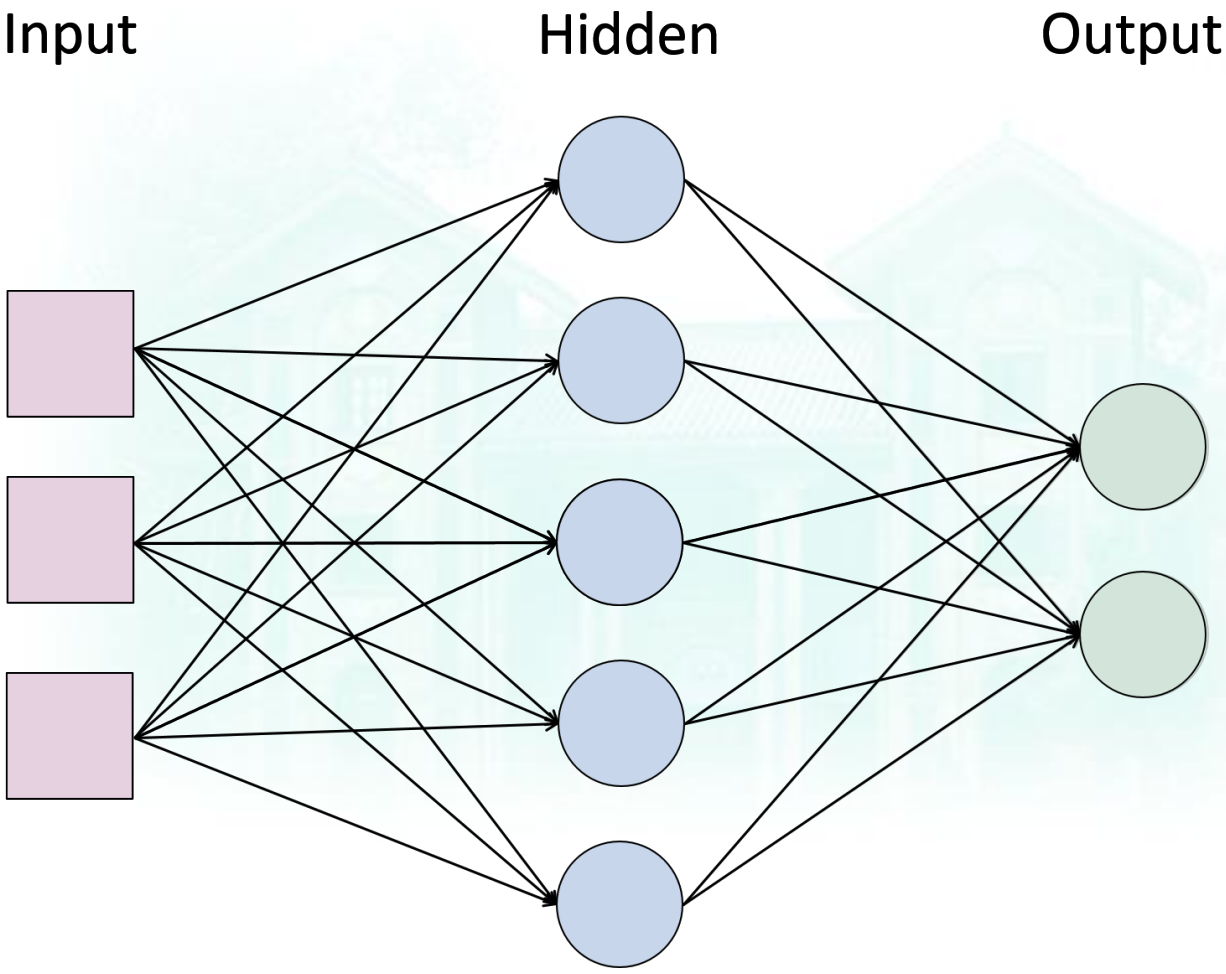
揭神秘面纱：为什么用神经网络模型？

万能近似定理（universal approximation theorem）(Hornik et al., 1989;Cybenko, 1989)：一个两层人工神经网络如果具有足够多的隐藏单元，它可以以任意的精度来近似任何一个函数（即可以表示任意的复杂输入-输出关系）。

更多层的神经网络比两层神经网络能够以更少的神经元表示具有同样复杂度的函数，并且性能更好！

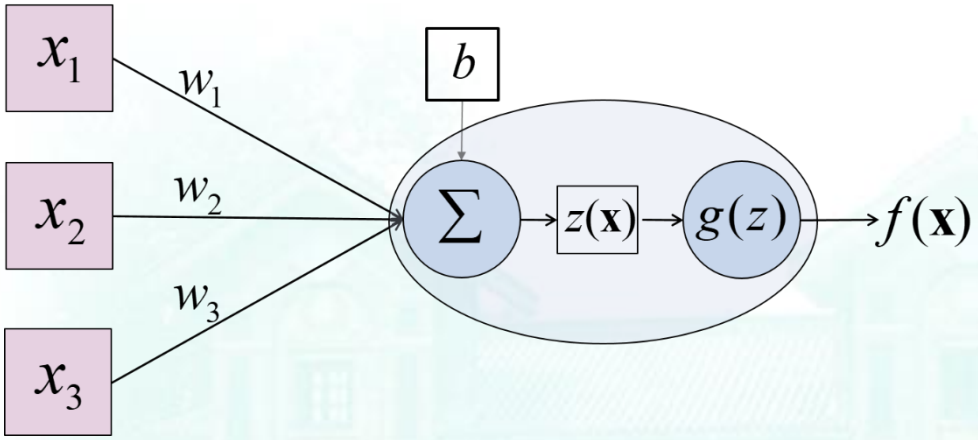
揭神秘面纱:传统全连接神经网络

- 两层全连接神经网络结构



揭神秘面纱:传统全连接神经网络

- 全连接神经网络中单个神经元:



$$z(\mathbf{x}) = \sum_i w_i x_i + b = \mathbf{w}^T\mathbf{x} + b$$

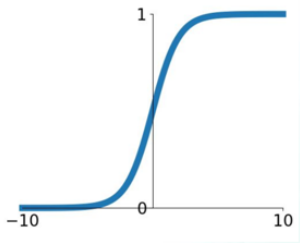
$$f(\mathbf{x}) = g(z) = g(\mathbf{w}^T\mathbf{x} + b)$$

$g(z)$: 激活函数

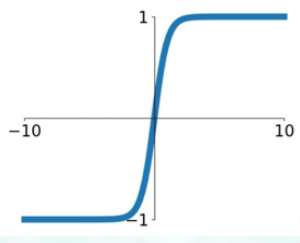
$\boldsymbol{\theta} = \{\mathbf{w}, b\}$: 模型参数

揭神秘面纱:传统全连接神经网络

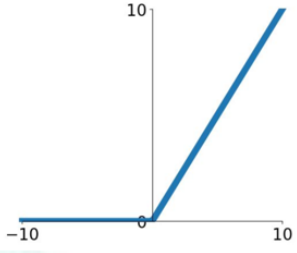
- 常见激活函数



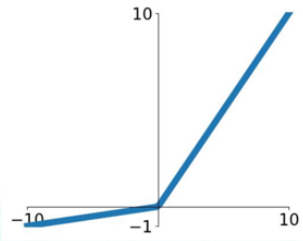
Sigmoid



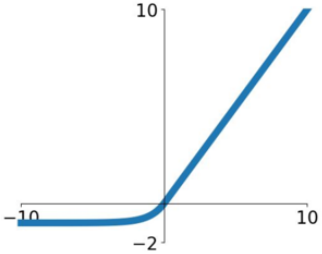
tanh



ReLU



Leaky ReLU



ELU

$$g(z)=\frac{1}{1+e^{-z}}$$

$$\tanh(z)$$

$$\max(0,z)$$

$$\max(\alpha z,z)$$

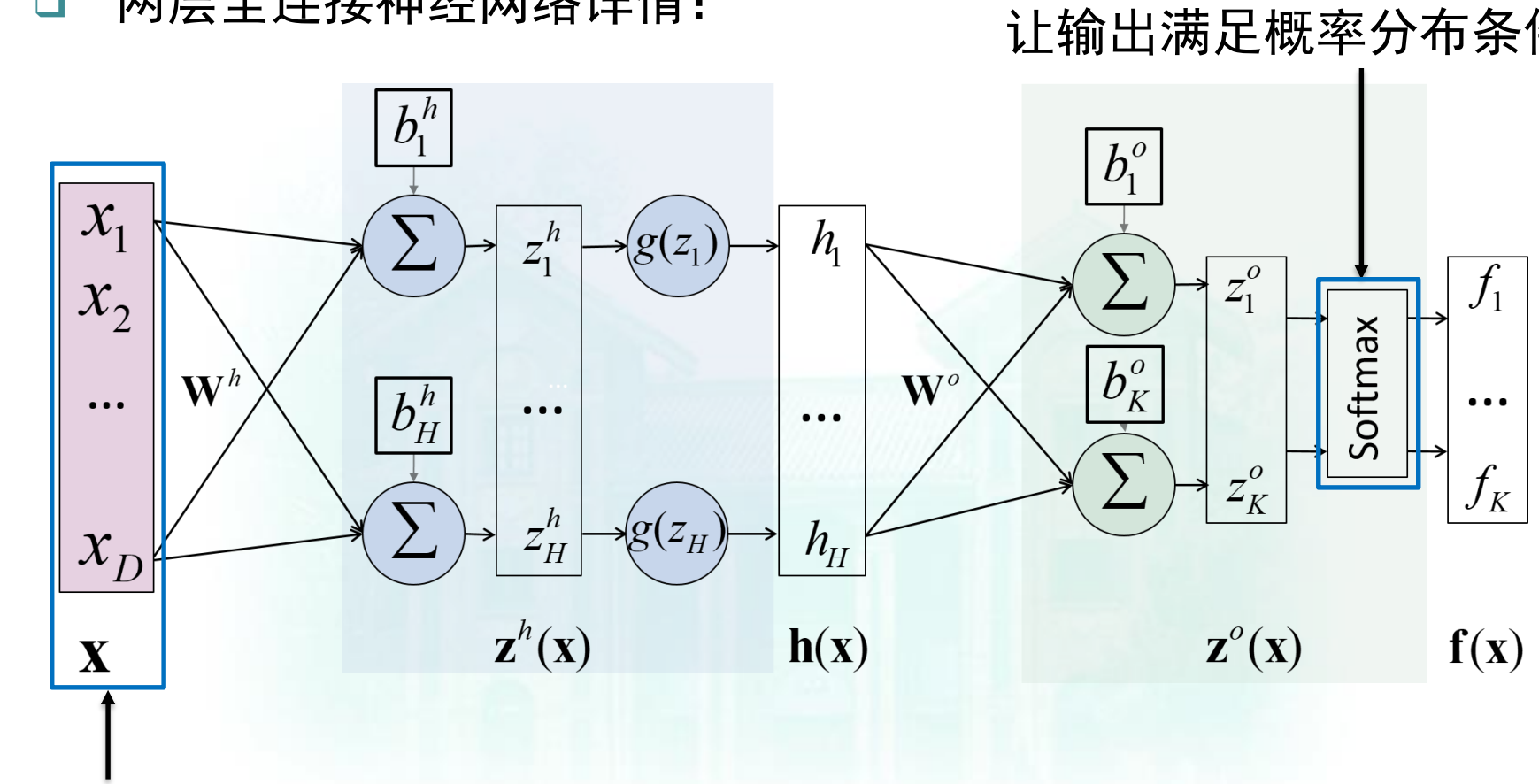
$$\begin{cases} z & z\geq 0 \\ \alpha(e^{z}-1) & z<0 \end{cases}$$

为什么需要激活函数？

让神经网络成为非线性函数，进而可以
表示输入-输出间的潜在复杂非线性关系！

揭神秘面纱：传统全连接神经网络

- 两层全连接神经网络详情：



数据表示为向量

上图中模型的参数包括？

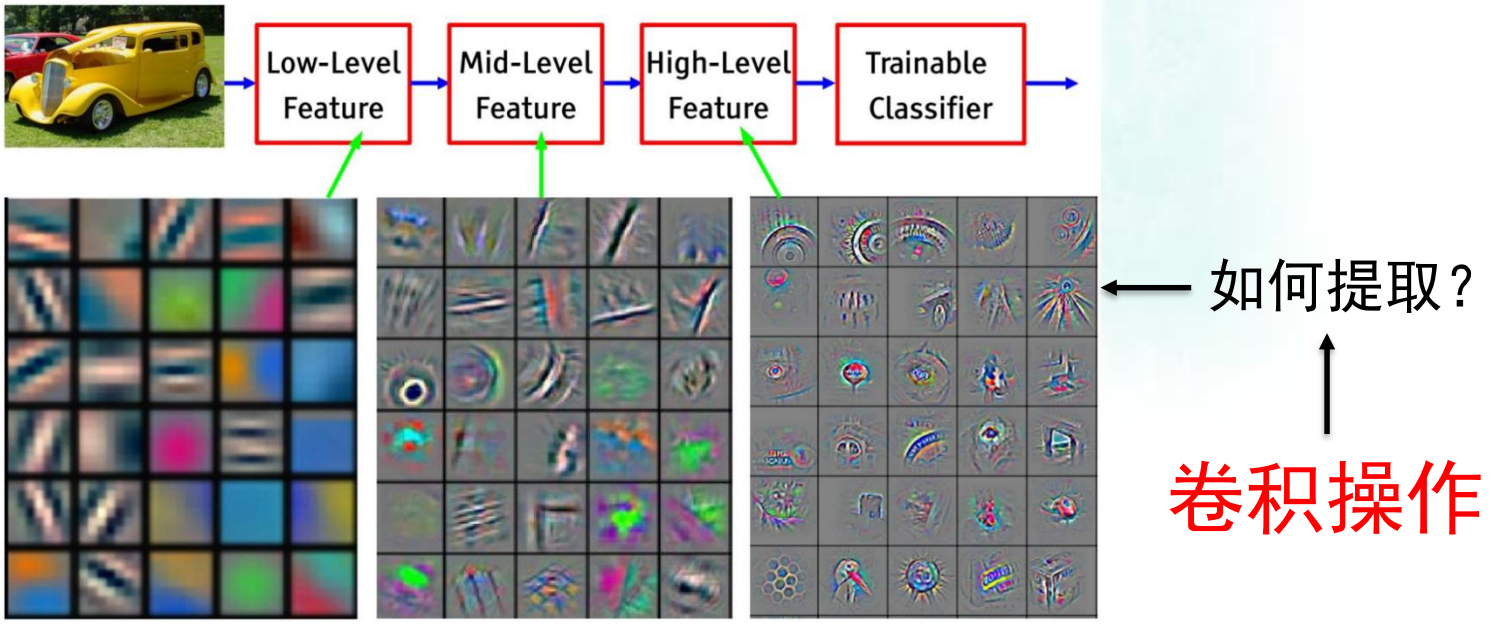
揭神秘面纱:数据的表示

- 如何将数据表示为向量，尤其是图像/视频/文本等数据？

- 以前：设计算法从数据中提取特征，比如SIFT+Bag of Words

  缺点–人工设计、与具体任务无关、可能漏掉有用特征

- 是否可以自动学习抽取与特定任务相关的特征？



揭神秘面纱:卷积

卷积是函数与函数之间的一种操作，结果为另一个函数

$$(f * g)(t) \equiv \int_{-\infty}^{\infty} f(\tau) g(t-\tau) d\tau$$

离散函数之间的卷积：

$$(f * g)[i] \equiv \sum_{m=-\infty}^{\infty} f[m]\, g[i-m] = \sum_{m=-\infty}^{\infty} f[i-m]\, g[m]$$

当函数$g(m)$只在$m \in [-M, M]$时非零：

$$(f * g)[i] = \sum_{m=-M}^{M} f[i-m]\, g[m]$$

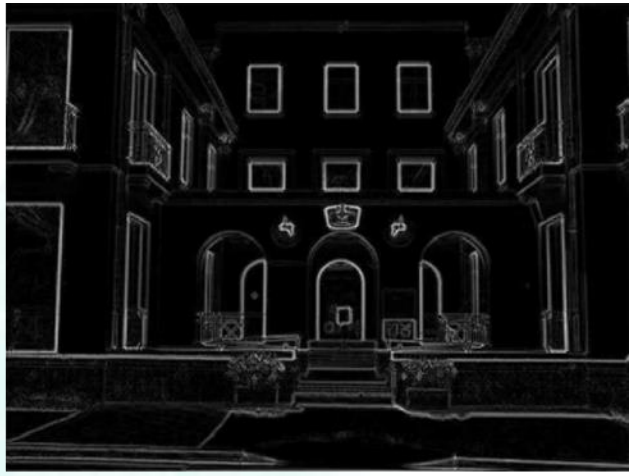
卷积可以扩展到二维函数：

$$(f * g)[i,j] = \sum_{m=-M}^{M} \sum_{n=-N}^{N} f[i-m, j-n]\, g[m,n]$$

揭神秘面纱:卷积

- 传统的边缘提取就是一个具体的卷积操作

$$* \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} =$$



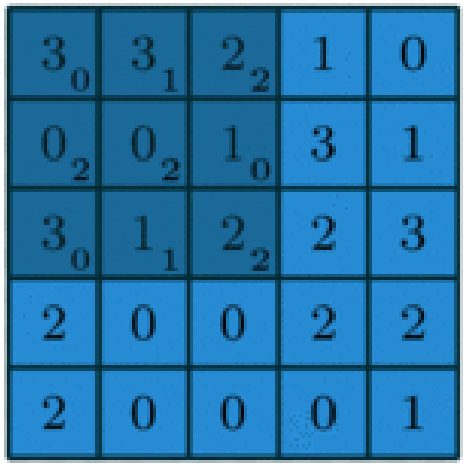
$f[i,j]$ ↑ 输入

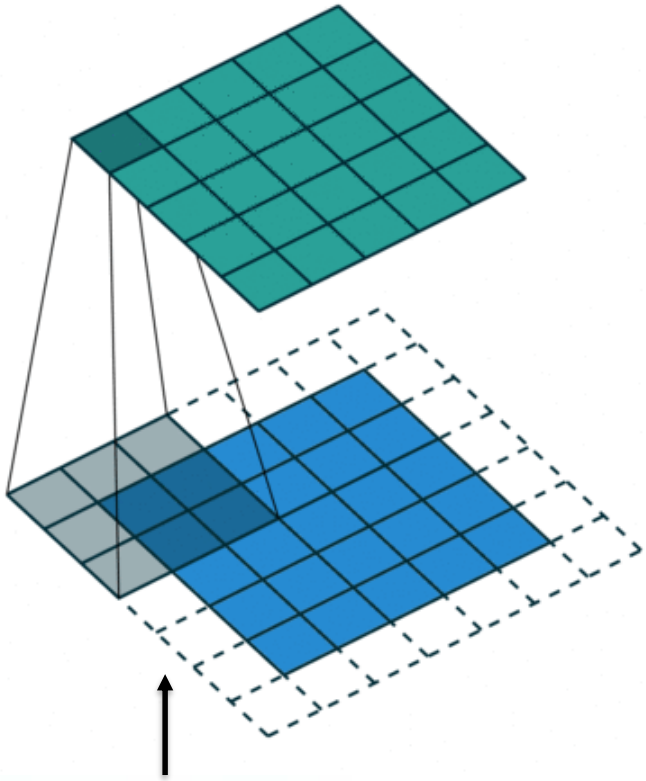
$g[m,n]$ ↑ 卷积核(Kernel)

$(f*g)[i,j]$ ↑ 特征图(Feature map)

揭神秘面纱:卷积

- 卷积具体操作（例子）：大小为5x5的输入、3x3的卷积核



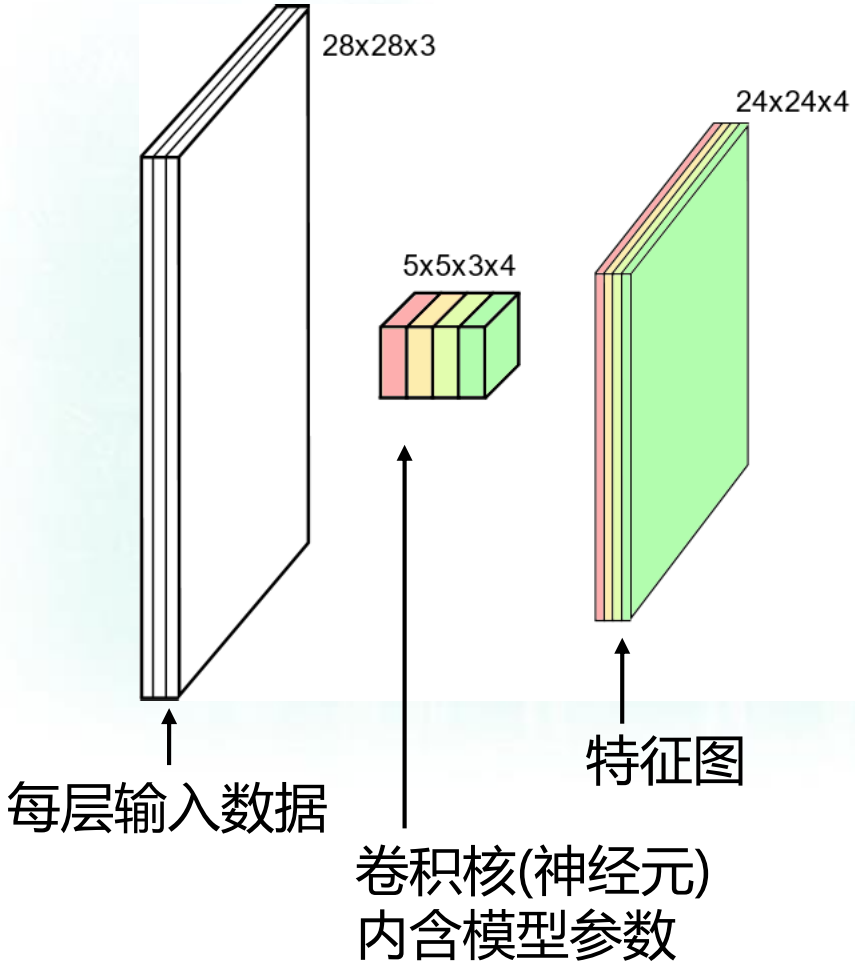


输出（特征图）小于输入的空间尺寸

Padding:对输入边缘补零，使卷积结果与输入尺寸一样

揭神秘面纱：卷积层

- 一个卷积层包含多个卷积操作，输出多个特征图
- 卷积层输出（经过激活函数等）作为下一卷积层的输入



Questions:

每个卷积核的维度？

卷积核通道数与输入通道数（Channels）的关系？

揭神秘面纱:卷积提取高层特征

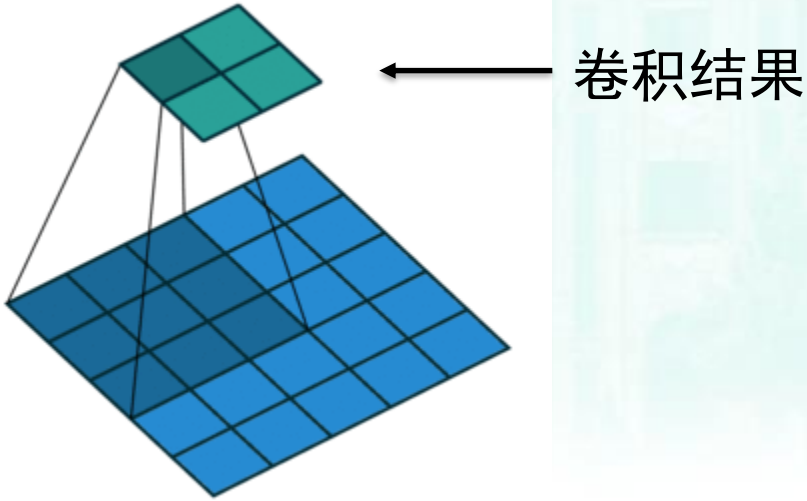
- 如何有效提取高层语义特征?
  - 语义特征对应更大图像区域（大尺寸卷积核？）
  - 语义特征需要忽略细节（大尺寸卷积核一般对细节敏感）

- 如何在不增加卷积核尺寸情况下提取语义特征?

揭神秘面纱：卷积中Stride

- 为了提取语义特征，需要减小（低层）特征图尺寸，然后与（高层）卷积核做卷积操作

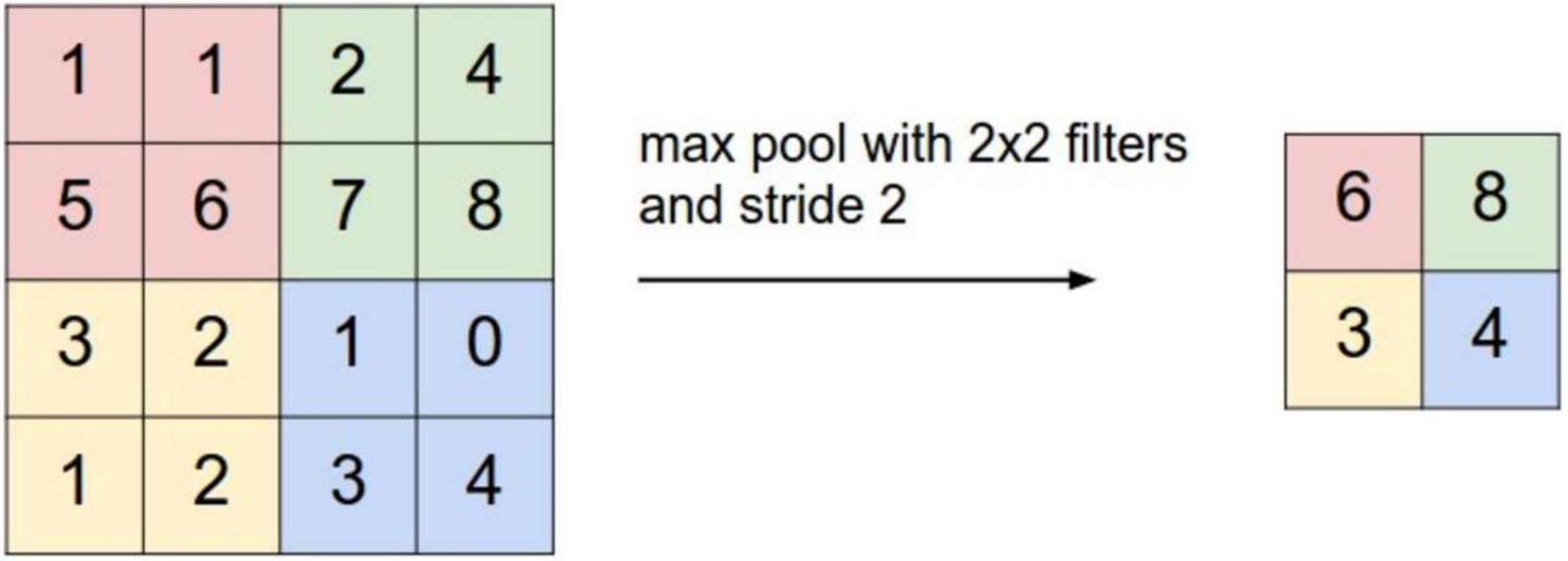
- Stride：卷积操作中卷积核相对于特征图每次移动距离
- 可有效减小特征图的空间尺寸



Stride=2 时的卷积操作

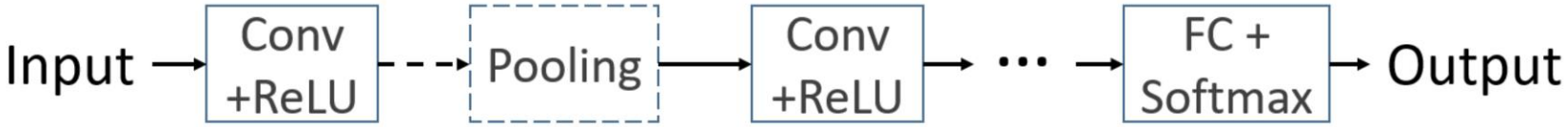
揭神秘面纱:卷积后池化

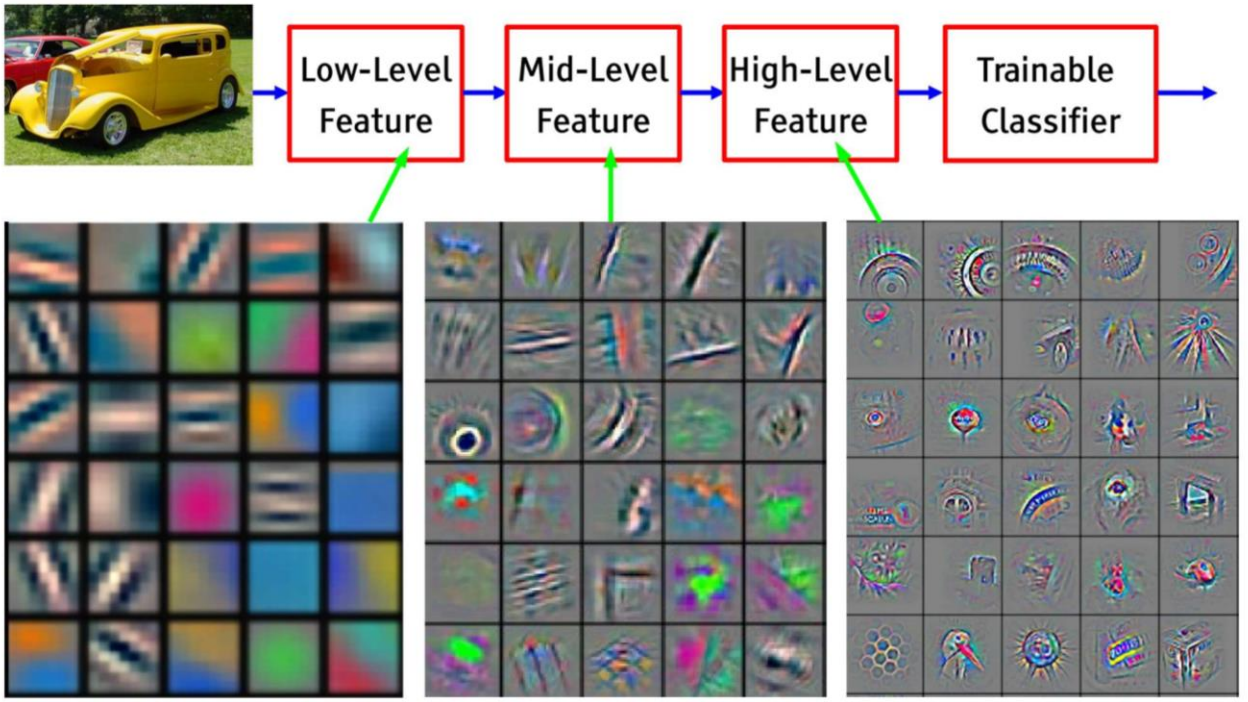
- 池化(pooling)：将每个特征图划分为多个（可重叠）局部区域，每个局部区域求均值或极大值输出
- 另外一种减少特征图尺寸的方法



揭神秘面纱:卷积神经网络CNN

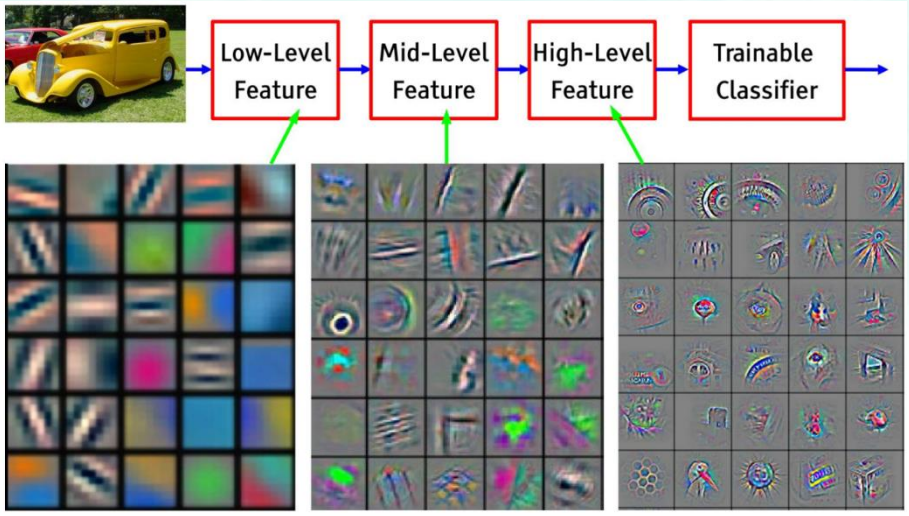
- CNN：多个（卷积层+激活函数）+多次池化+全连接层



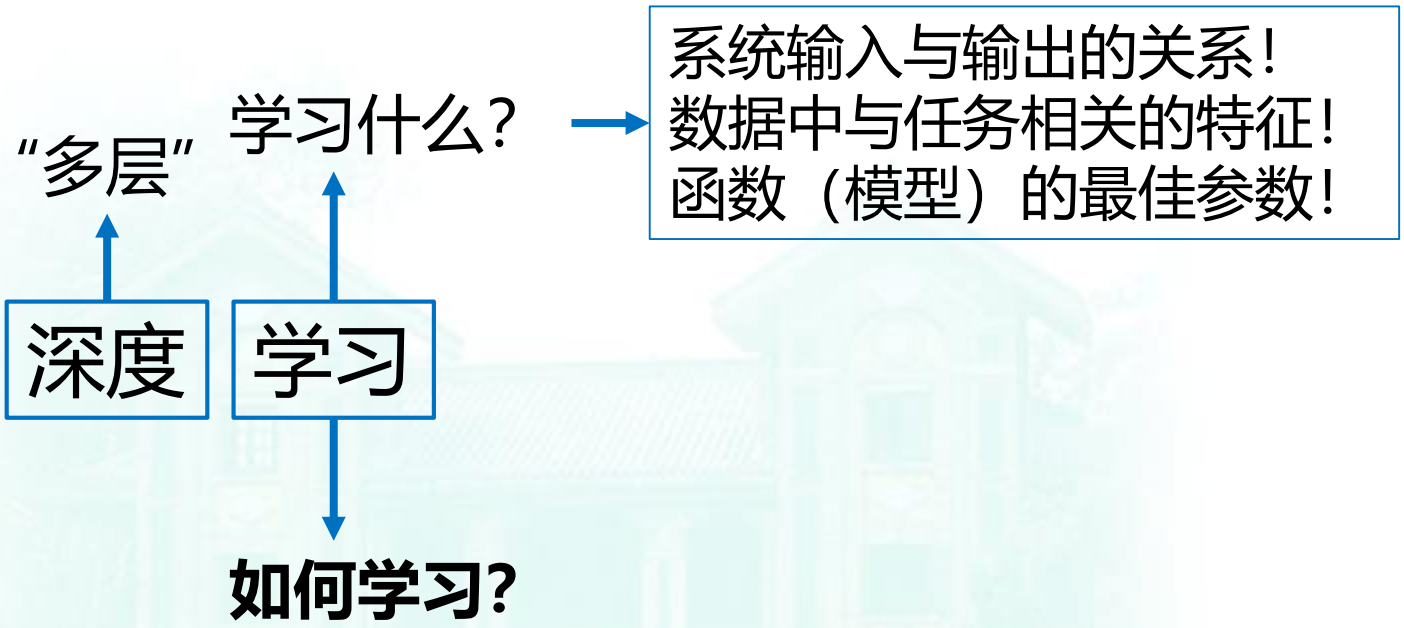


揭神秘面纱:深度学习(Deep Learning)

- “深度”：网络层数多
- “学习”：利用数据训练网络，找到每个卷积核的最佳参数，实现自动提取与任务相关的特征，即“特征学习”
- 注：每个卷积核（参数）不是人为设计的，而是自动学习的！
- 输入端是原始数据，输出端是预测结果，中间过程全部自动化（不用人为设计特征提取算法），所以叫“端到端学习”！
- 最终：学习到输入-输出关系



揭神秘面纱



揭神秘面纱：如何学习

- 比如针对分类任务

训练数据　模型参数

$$\min_{\boldsymbol{\theta}} \quad L(\boldsymbol{\theta}) \quad = \quad \frac{1}{N}\sum_{i=1}^{N} l(\mathbf{y}_i, \mathbf{f}(\mathbf{x}_i; \boldsymbol{\theta}))$$

$$= \quad -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik} \log \hat{y}_{ik}$$

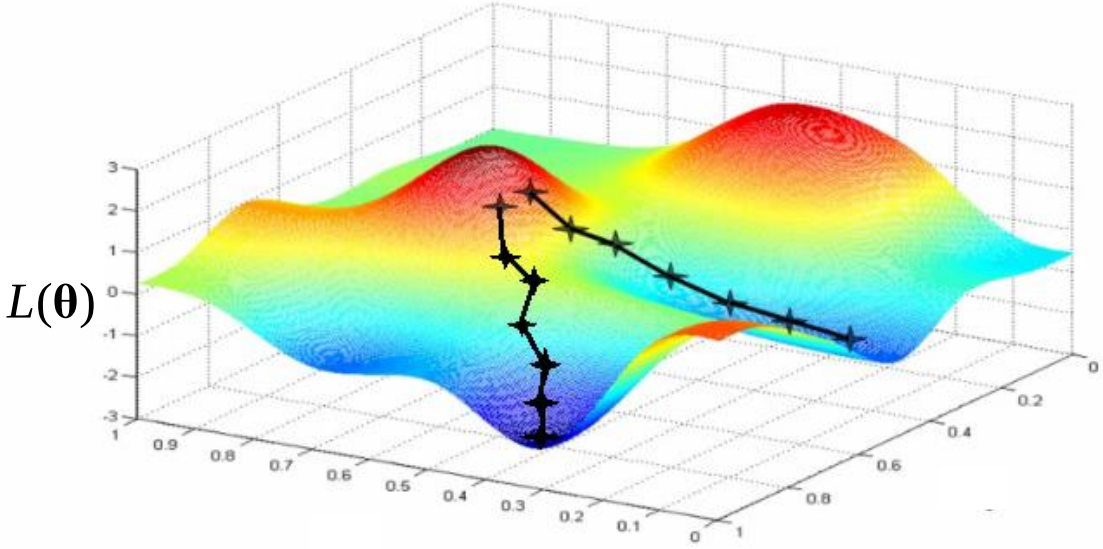
寻找模型（函数）最佳参数 = 最小化损失函数$L(\boldsymbol{\theta})$

梯度下降法！

基于训练数据，通过梯度下降法最小化损失函数，以此找到最佳的模型参数！

揭神秘面纱:如何学习好

- 梯度下降法不是只能找到局部最优解么?



局部最优解和全局最优解对应的模型性能常常类似!
大数据+随机梯度下降法让模型学到更准确输入输出关系!
各种训练策略(正则化,Dropout等)提升模型的预测性能!

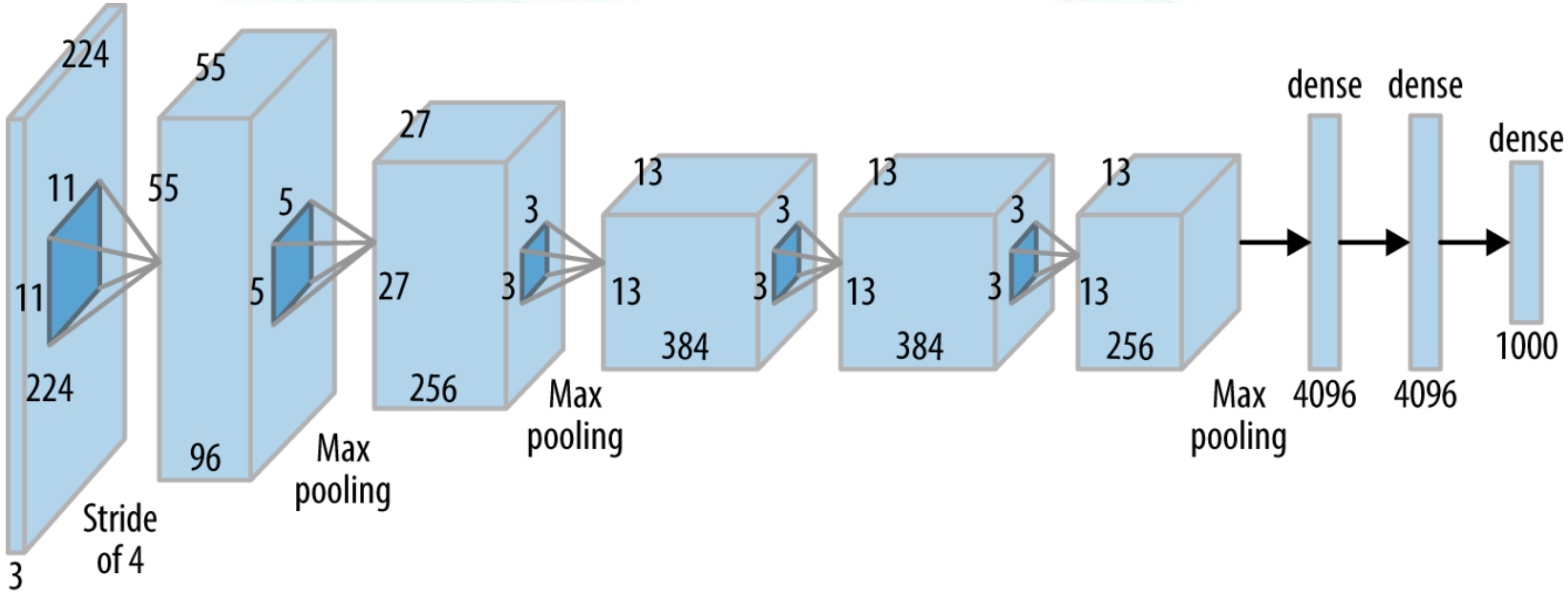
还没有定论,还在研究探索中...

赏群模乱舞

深度学习模型结构和应用场景多种多样！

群模乱舞之图像分类:AlexNet （2012）

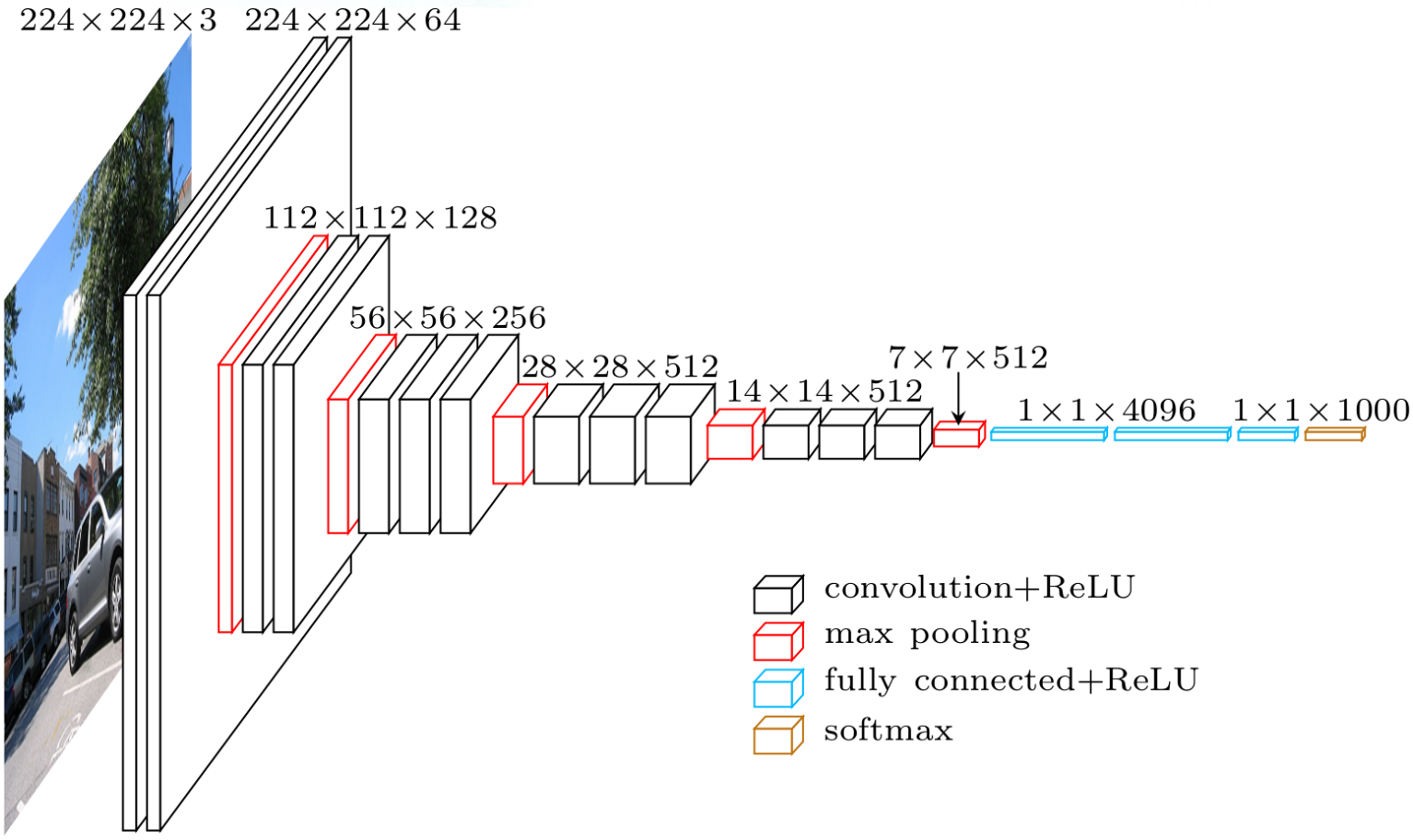
- AlexNet: CNN，5卷积层+3全连接层，1000类图像分类器
- 分类性能远超人工设计的特征提取器+训练的分类器
- 从此深度学习开始进入科研人员视野！



注：每个卷积层输出之后都接有ReLU激活函数

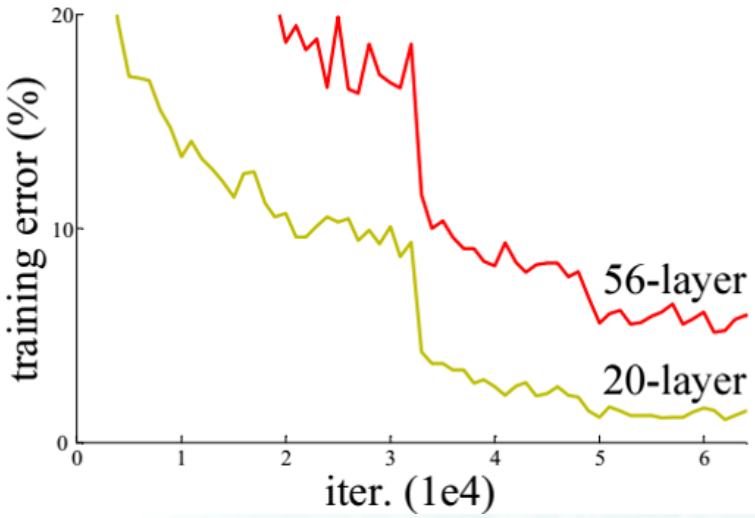
群模乱舞之图像分类:VggNet （2014）

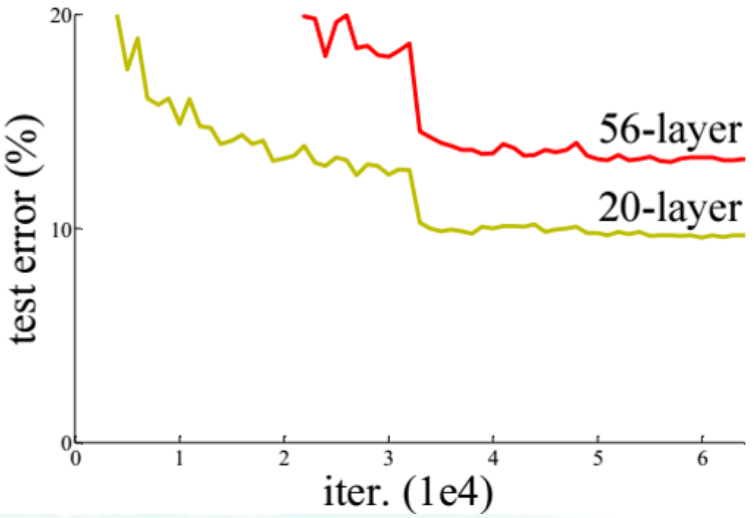
- 更多卷积层，每个卷积核大小为3x3
- 层数越多，越能表示更复杂的输入-输出关系



群模乱舞之图像分类:ResNet (2015)

- 奇怪：56层CNN分类器在训练集上分类误差大于20层CNN分类器！
- 说明：网络太深（层数太多），难于训练（被优化）！





群模乱舞之图像分类:ResNet (2015)

- 残差神经网络 (Residual Network)
- 创新点：Skip connection

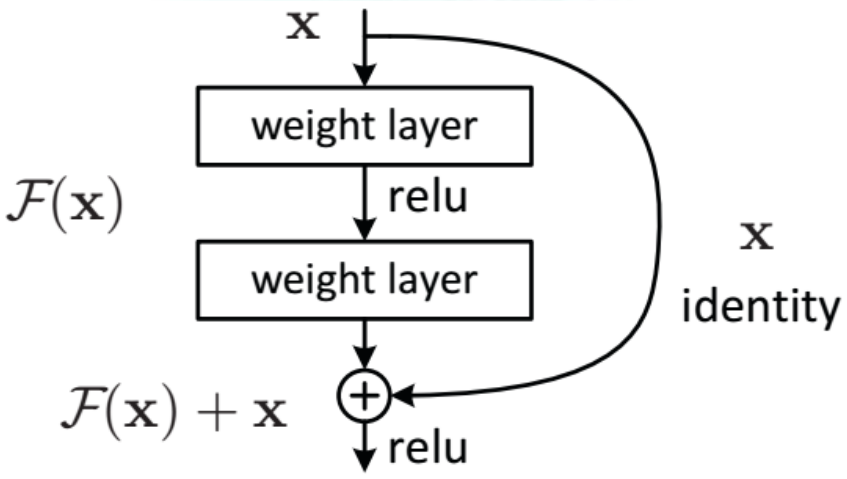
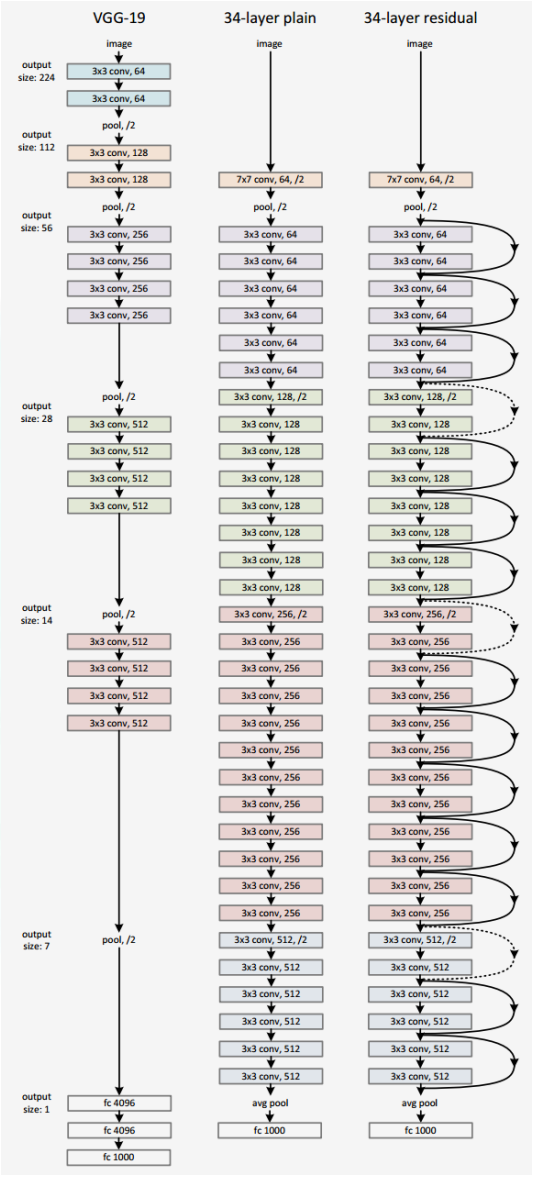


Figure 2. Residual learning: a building block.

$$\mathcal{H}(\mathbf{x}) = \mathcal{F}(\mathbf{x}) + \mathbf{x}$$

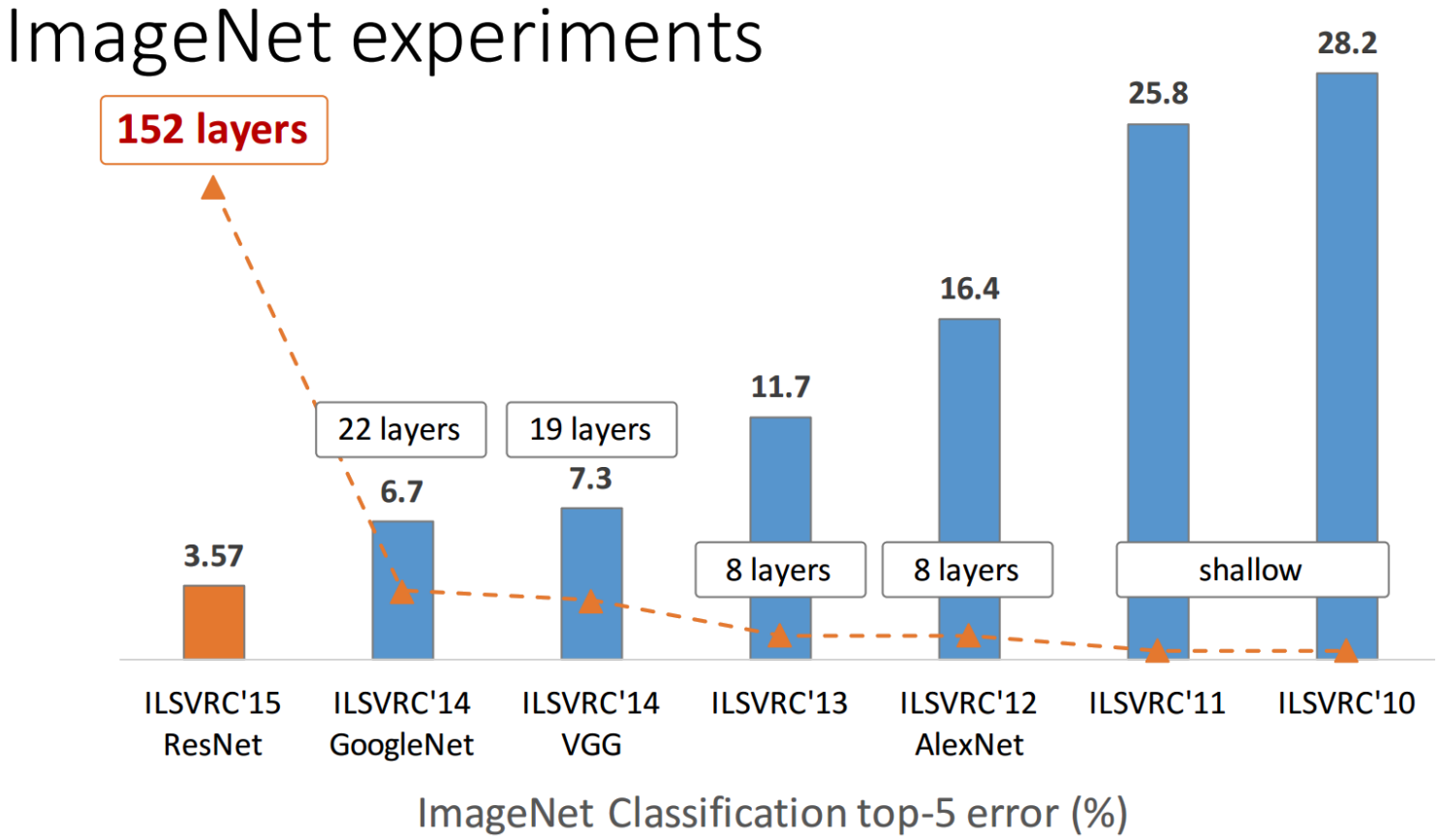
$$\mathcal{F}(\mathbf{x}) = \boxed{\mathcal{H}(\mathbf{x}) - \mathbf{x}}$$

Residual



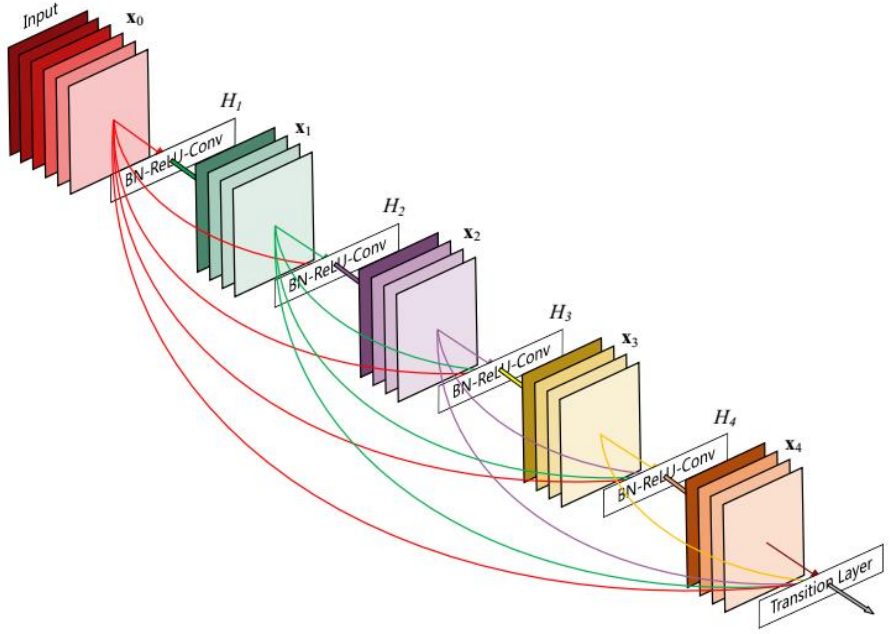
群模乱舞之图像分类:ResNet (2015-2016)

- 在1000类图像分类上首次超过人类表现!

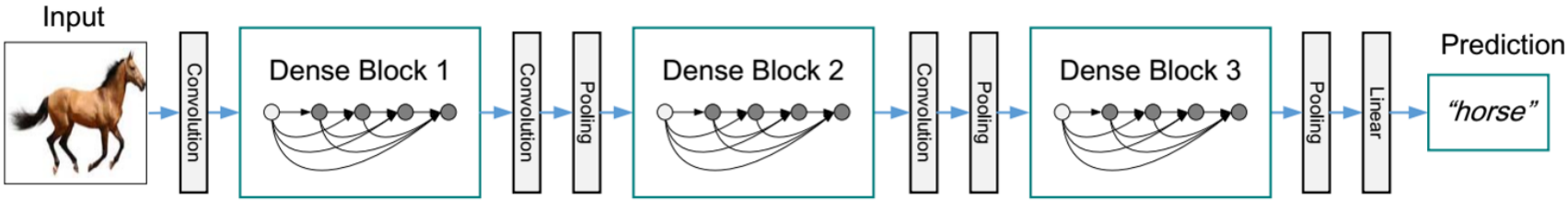


群模乱舞之图像分类:DenseNet (2017)

- 更多卷积层之间加入skip connections



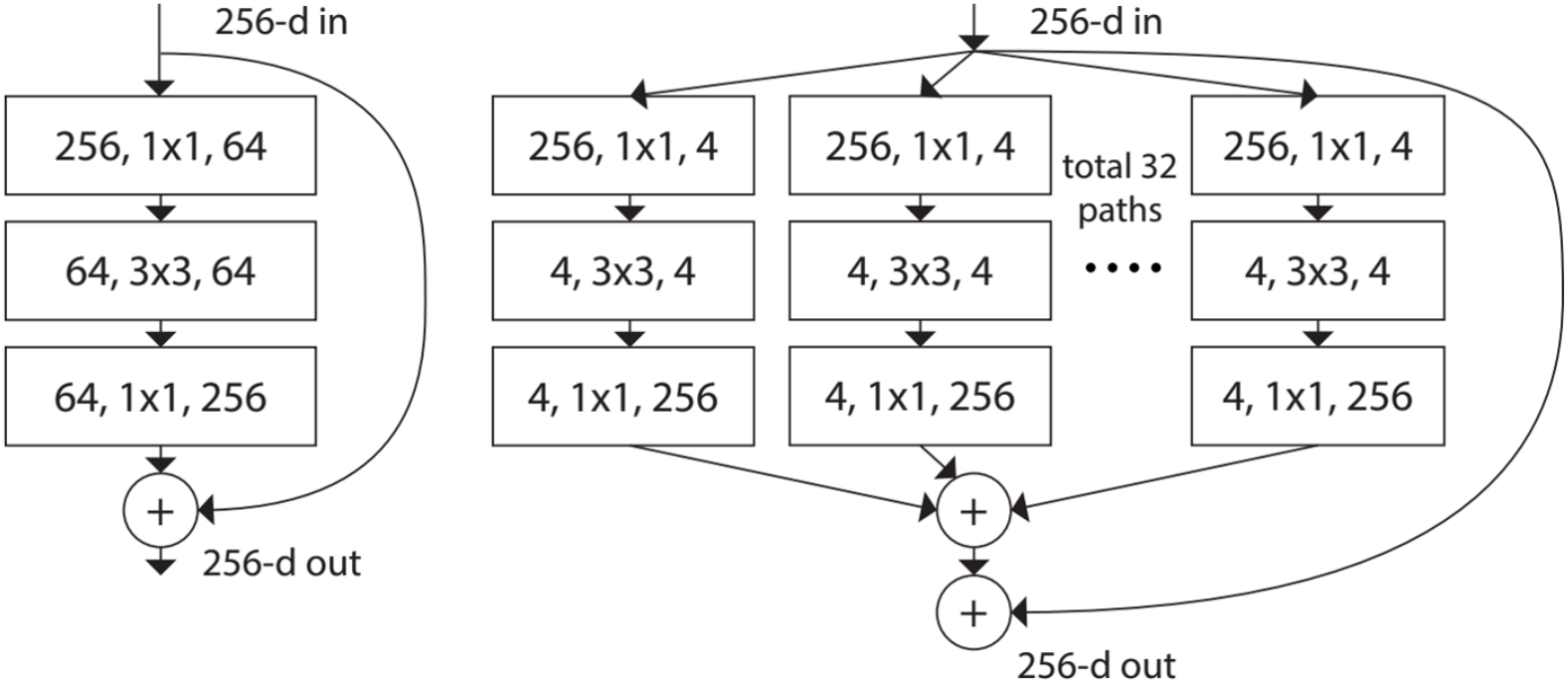
Dense block



DenseNet一般不需要太多卷积层

群模乱舞之图像分类:ResNeXt（2017）

- Split-Transform-Merge:分组卷积，再将结果合起来
- 同样多的模型参数，比ResNet分类性能更好

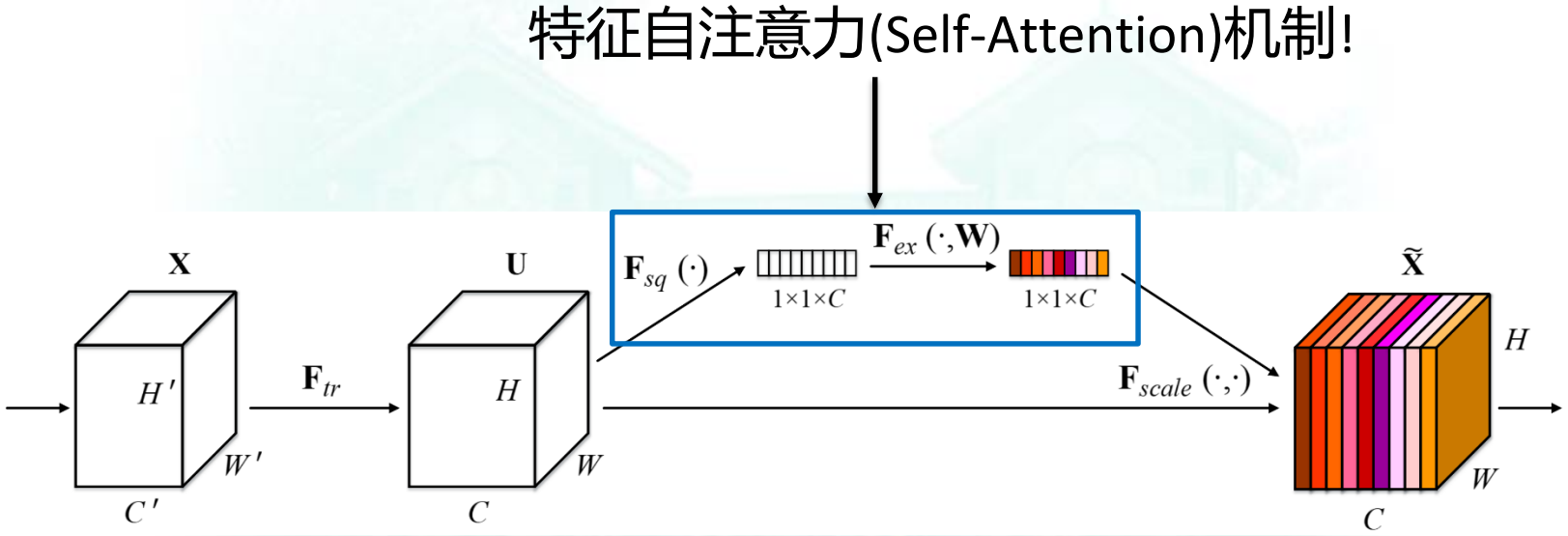


ResNet block

ResNeXt block

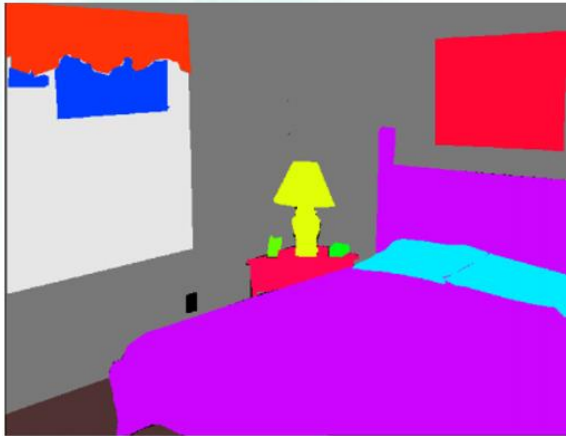
群模乱舞之图像分类:SENet (2018)

- Squeeze-and-Excitation Network:每个卷积层输出端加入SE模块
- 自动估计每个特征图的重要性，得到加权的特征图



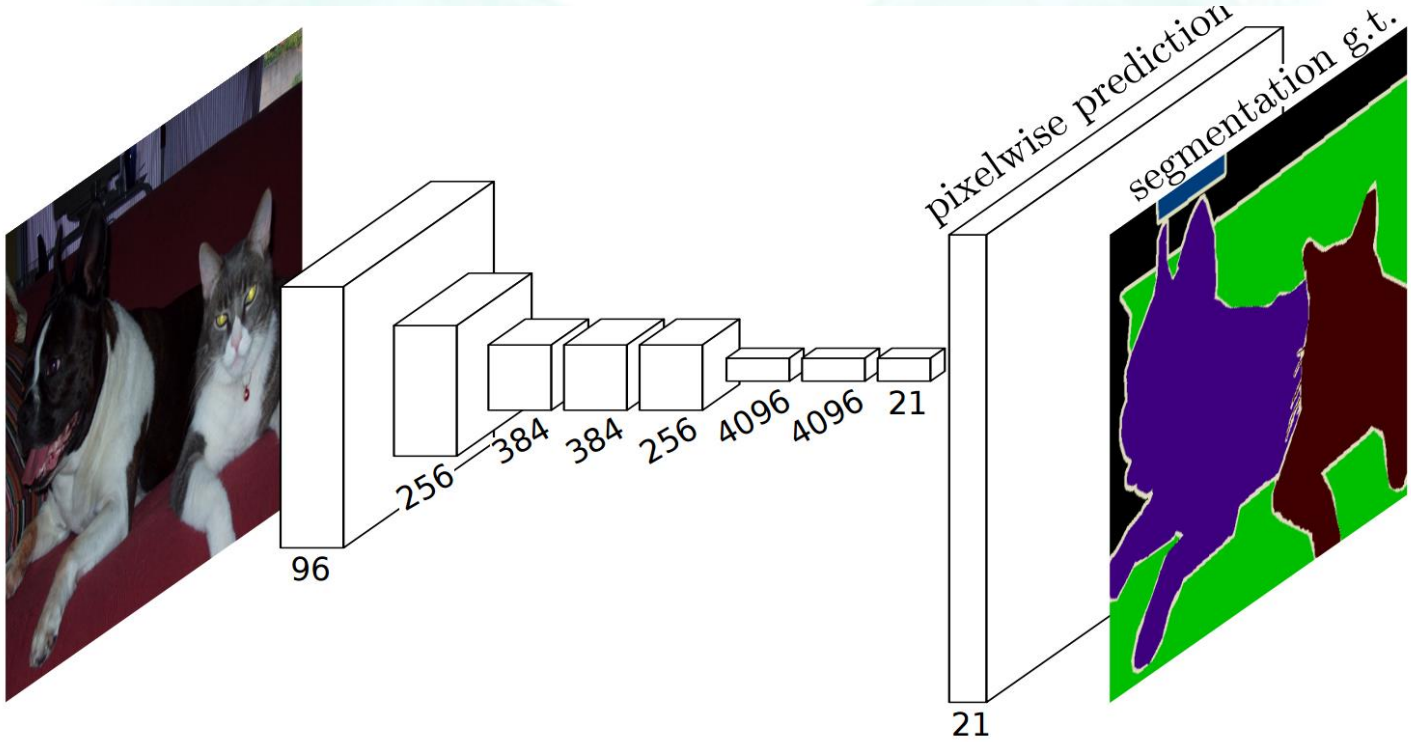
群模乱舞之图像语义分割

- 图像语义分割：将图像分割为不同种类的区域
- 目标：得到每个像素的种类



群模乱舞之图像语义分割

- 分割模型输出：大小与输入一样，层数等于区域类型的个数
- 输出包含了每个像素属于每一类别的概率
- 挑战：多层卷积后的输出大小远小于输入数据
- 思路：需要某种上采样操作将卷积输出尺寸变为与输入一样

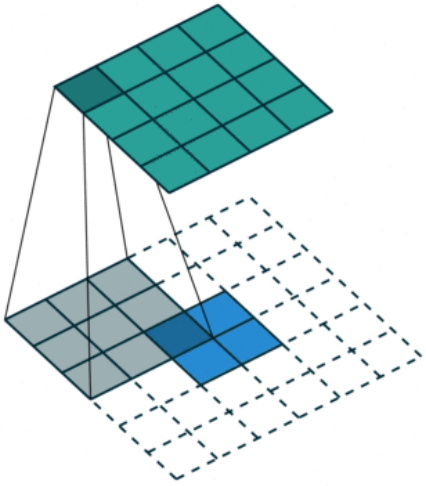


群模乱舞之图像语义分割：反卷积

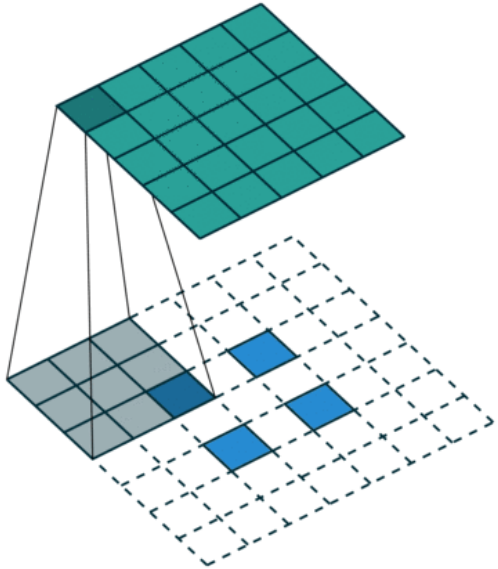
- Deconvolution（反卷积）：仍然是卷积操作
- Stride和Padding的作用与传统卷积里的作用几乎相反

输出
(绿色)

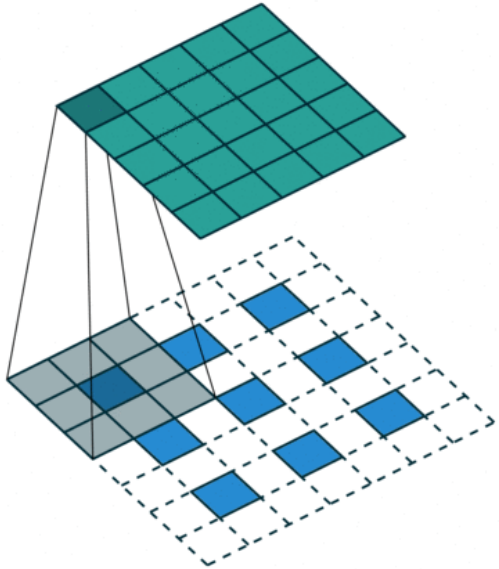
输入
(蓝色)



Stride=1
No padding



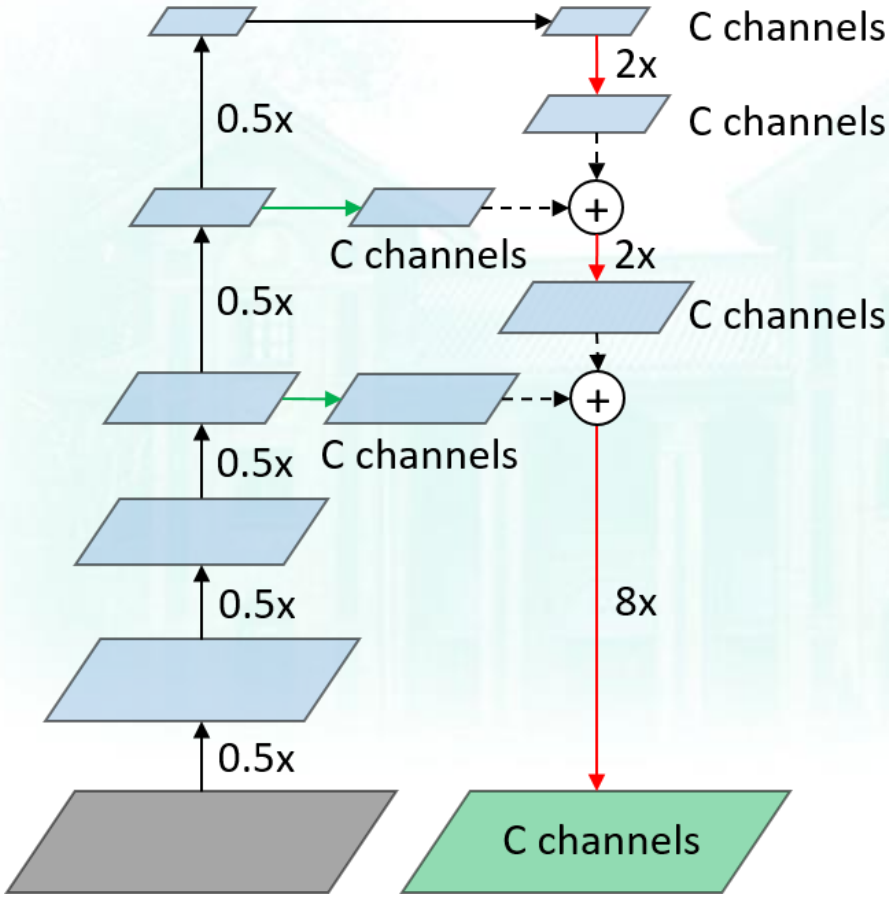
Stride=2
No padding



Stride=2
Padding=1

群模乱舞之图像语义分割：FCN

- 全卷积神经网络（Fully Convolutional Network）
- 结构：编码器（卷积层）+译码器（反卷积层）



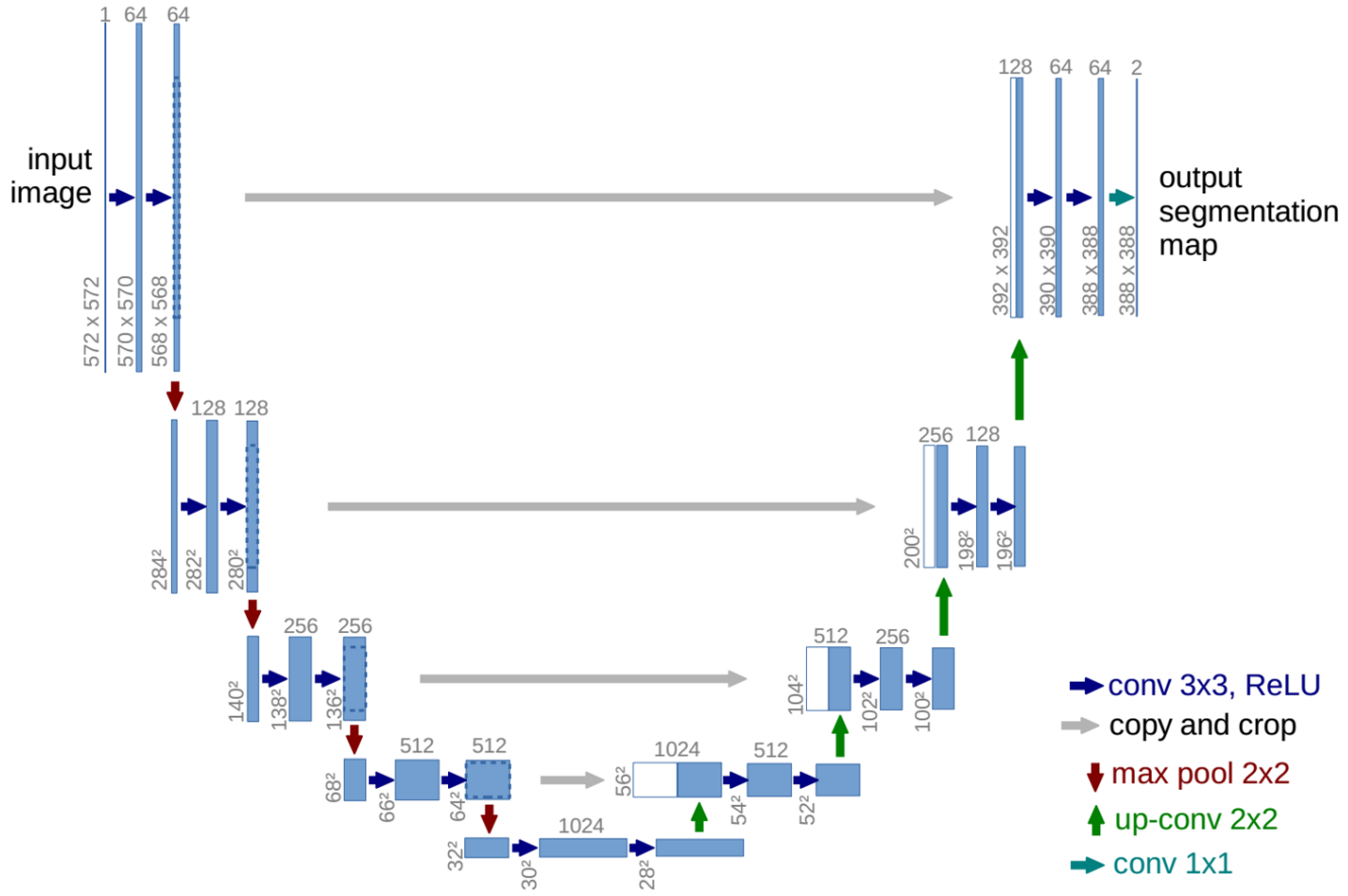
Stride=2
Kernel size 4x4

Stride=2
Kernel size 4x4

Stride=8
Kernel size 16x16

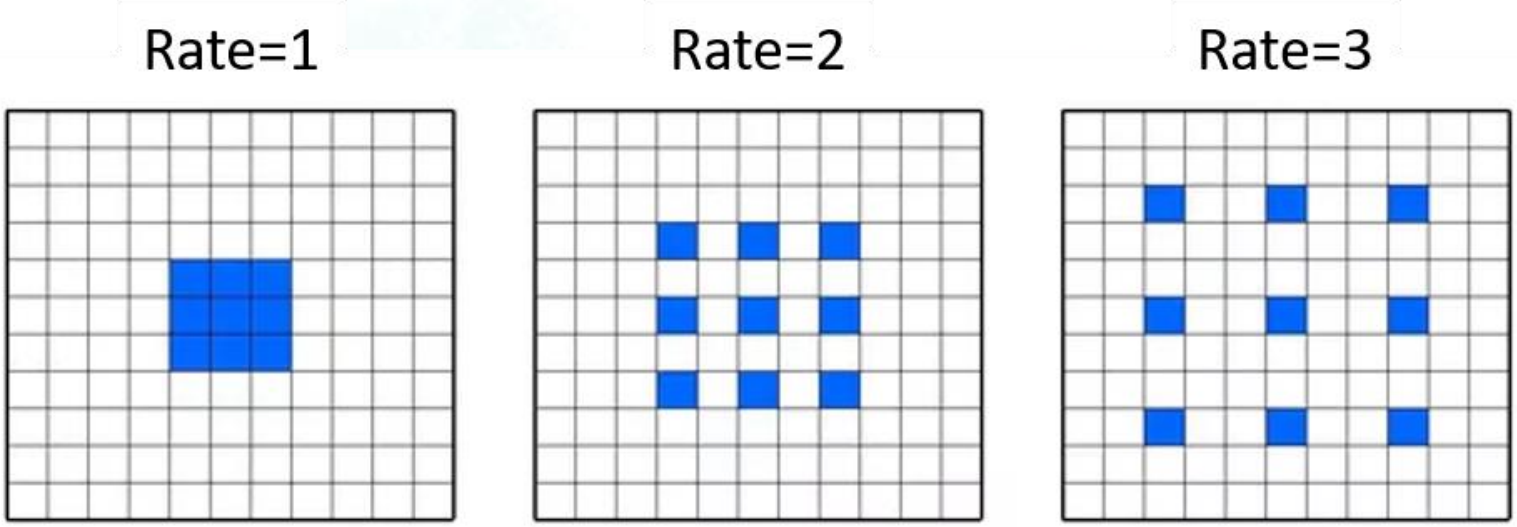
群模乱舞之分割模型：U-Net

- U-Net:FCN改进版，在上采样通道中叠加下采样通道的信息
- 上采样过程中融合了更多的图像细节信息和不同层次的特征



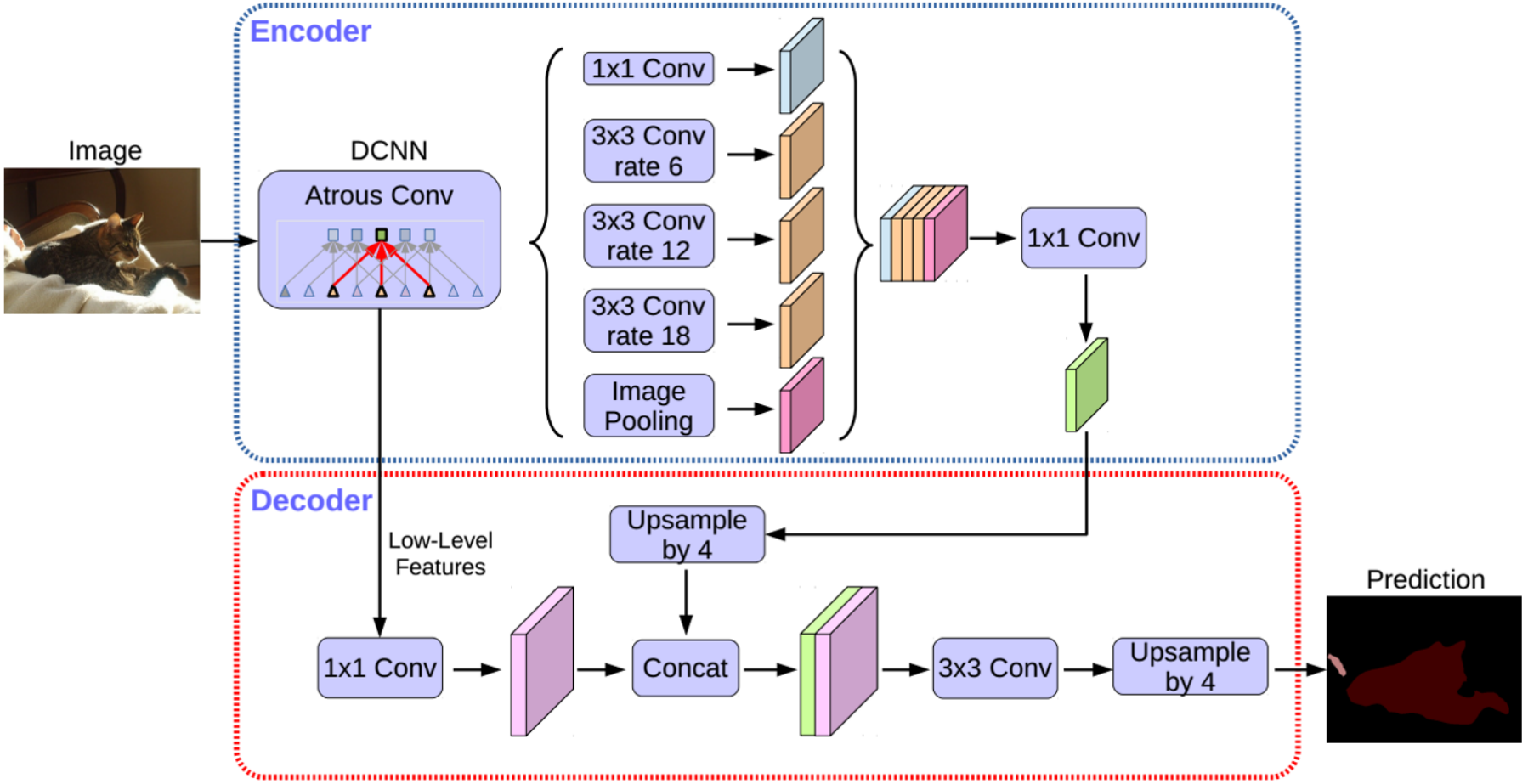
群模乱舞之图像语义分割：DeepLab

- Dilated (Atrous) convolution：卷积核参数量不变，但与之对应的特征图中像素所在区域可大于卷积核尺寸



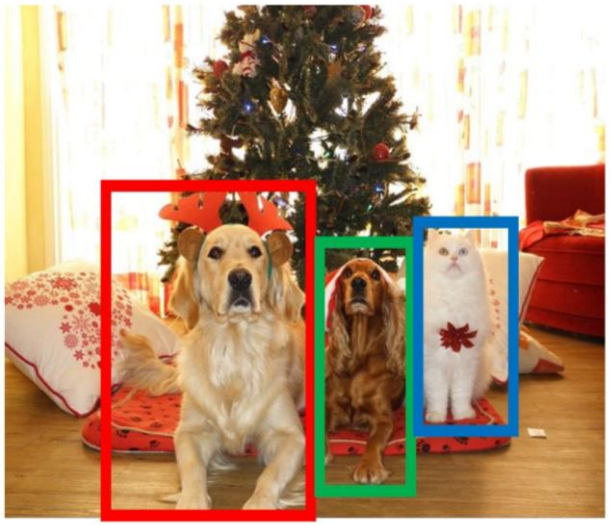
群模乱舞之图像语义分割：DeepLab V3+

- 考虑多种尺度的图象特征，并结合其它分割模型编码器优点



群模乱舞之目标检测

- 任务：从图像中检测出（未知数量）物体的位置、大小、类型



一般分两步来解决任务：

第一步：找到可能的物体区域（边框）

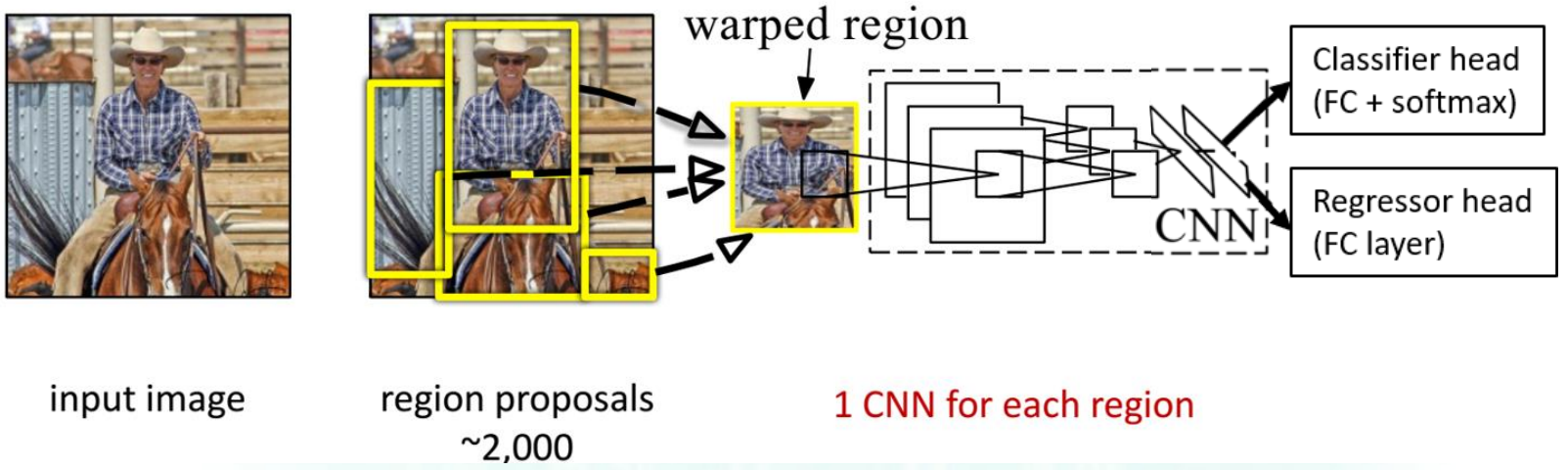
第二步：估计每个物体区域内物体类型，并自动微调边框大小与位置

分类任务

回归任务（边框坐标）

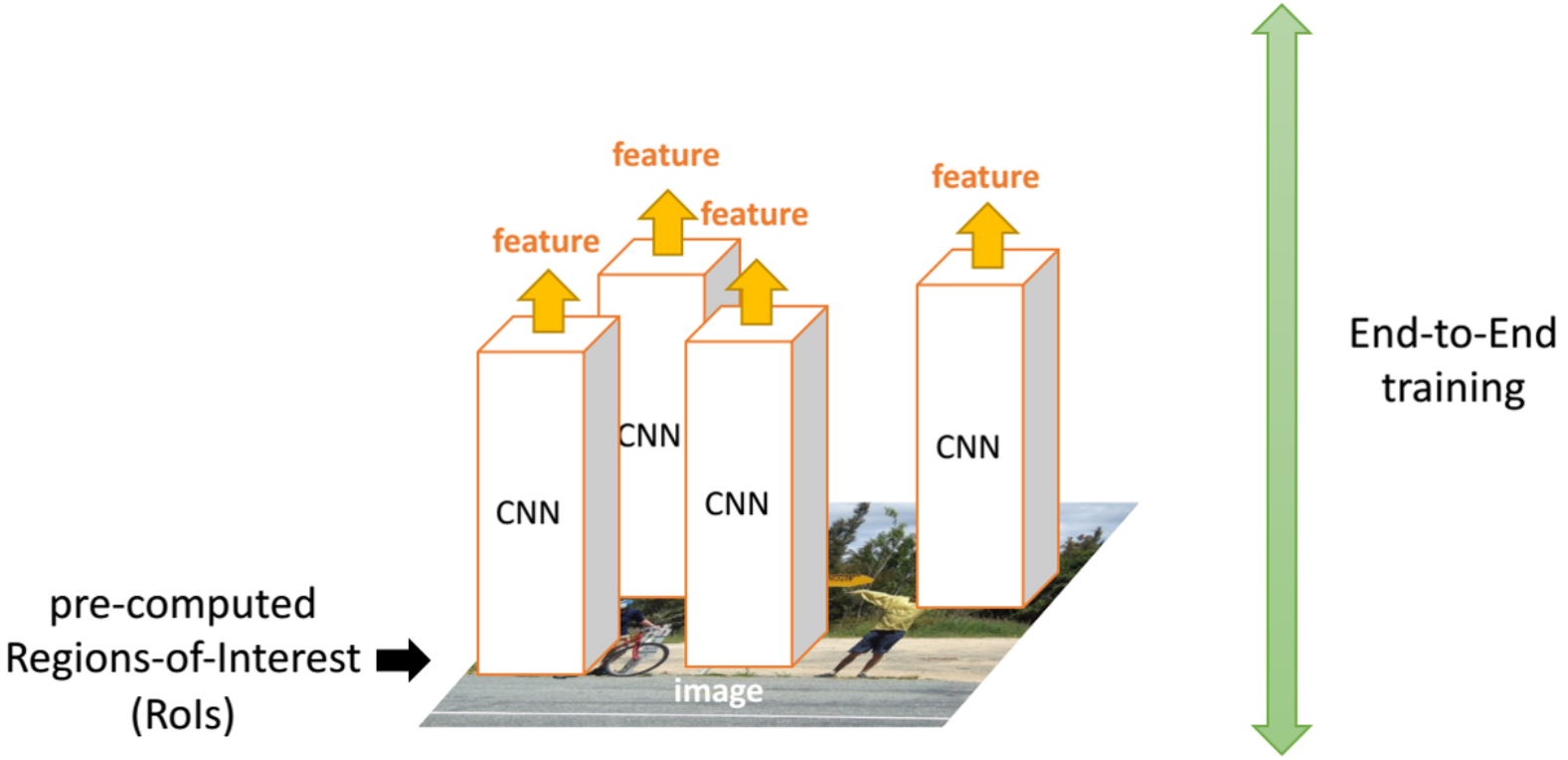
群模乱舞之目标检测：R-CNN

- Region-based CNN
  - 第一步：利用其它已有方法预先找到大量可能的物体区域
  - 第二步：基于这些区域训练共享卷积层的分类器和回归模型



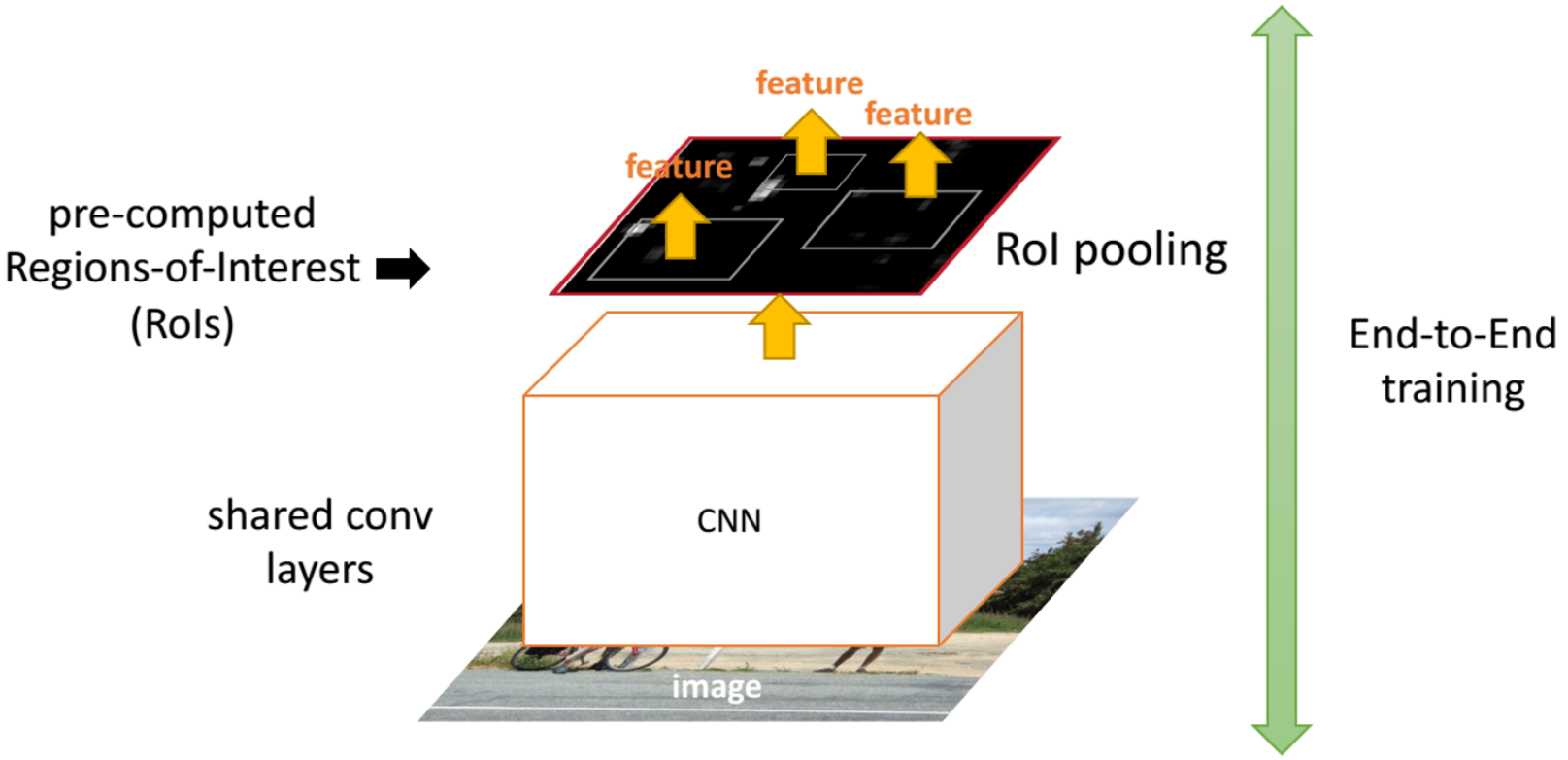
群模乱舞之目标检测：R-CNN

- 可能的物体区域从原始图像中估计得到
- 从图像中进行目标检测时，同一个CNN被执行多次！
- 因此，检测速度很慢！



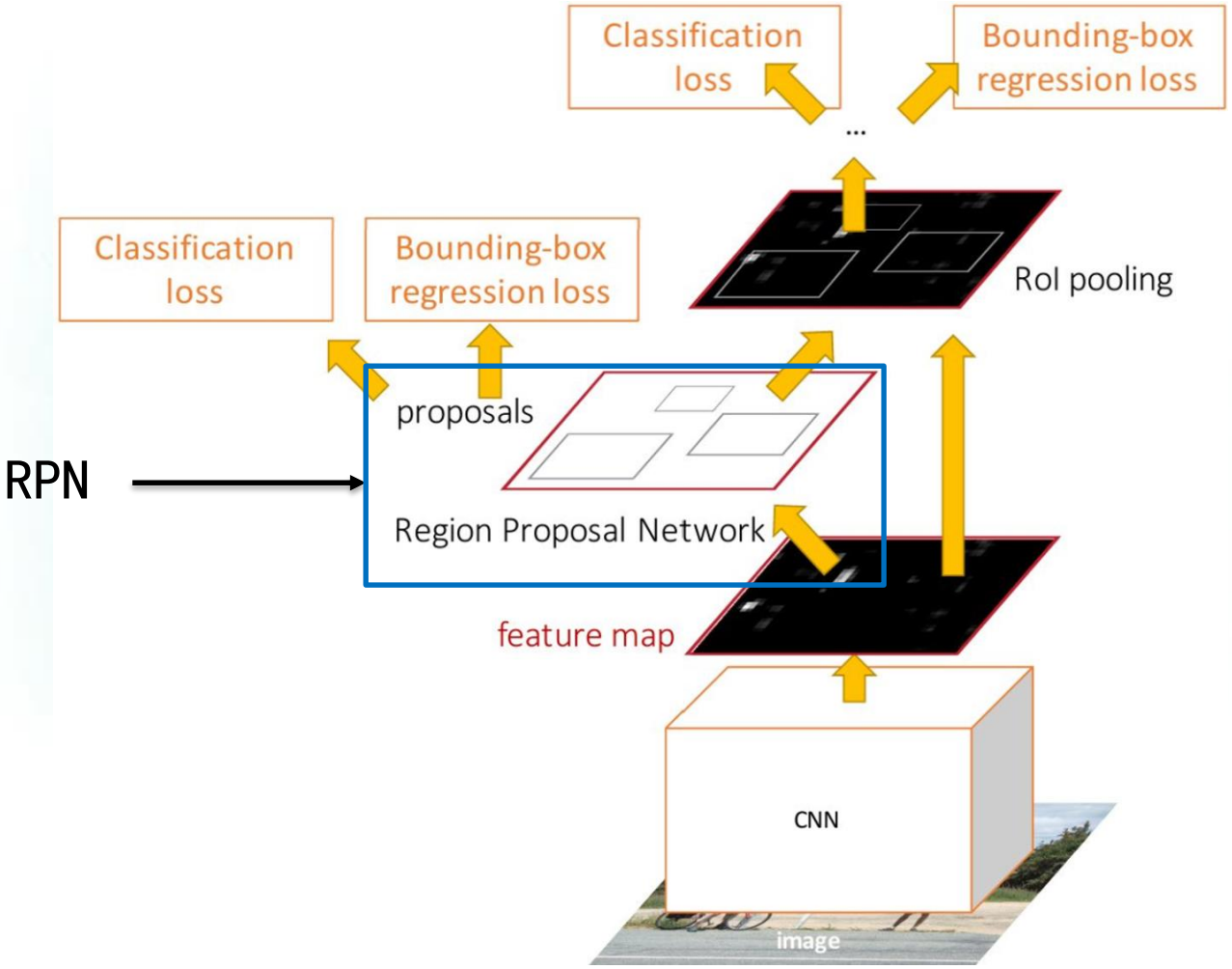
群模乱舞之目标检测：Fast R-CNN

- 加速思路：可能的物体区域从卷积层输出（特征图）得到
- 对一张图像中所有可能的物体检测，卷积操作只被执行一次！
- 大大加快了检测速度！



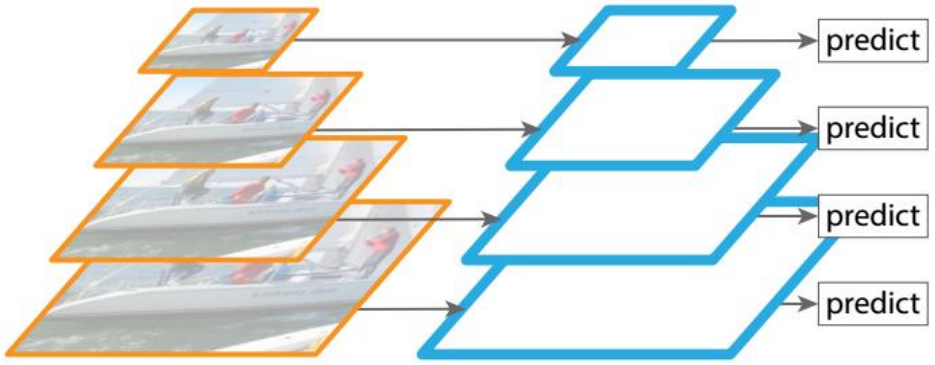
群模乱舞之目标检测：Faster R-CNN

- 进一步加速：同时训练一个子网络RPN寻找可能的物体区域

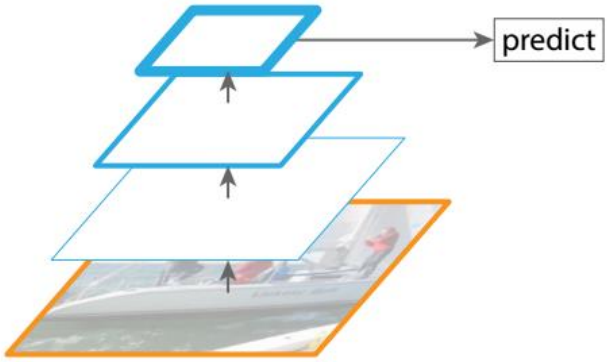


群模乱舞之目标检测：提升检测精度

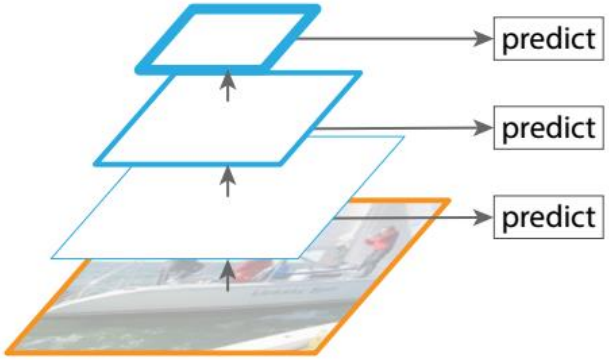
- 特征金字塔网络(FPN)在不同尺度的特征图里检测目标



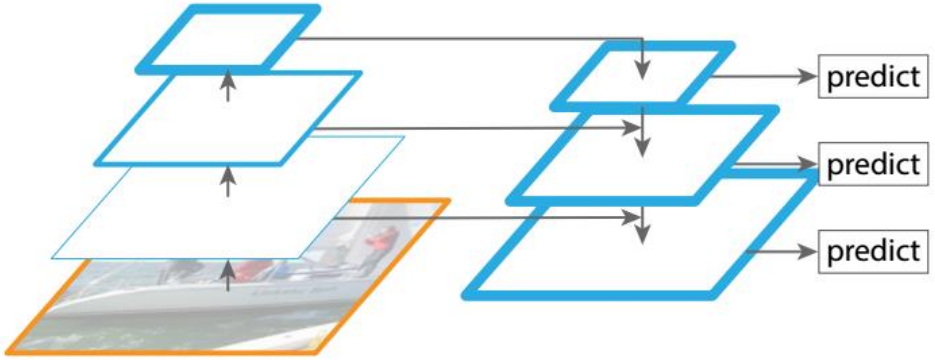
(a) Featurized image pyramid



(b) Single feature map



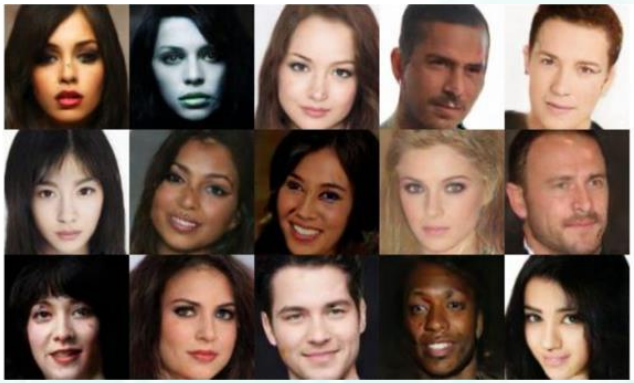
(c) Pyramidal feature hierarchy



(d) Feature Pyramid Network

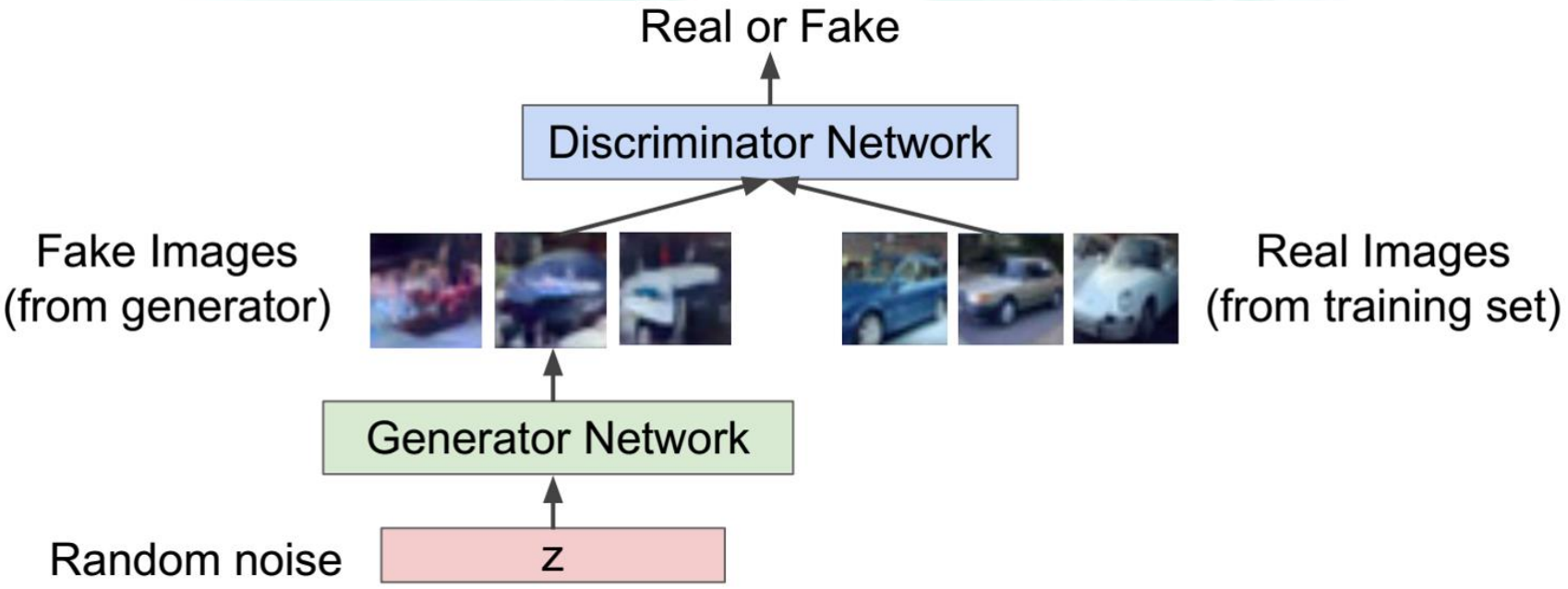
数据生成任务

- 基于已有数据，生成更多的相似数据
- 应用广泛：数据增广，高清成像，艺术设计，风格转移等



群模乱舞之GAN：初始模型

- 生成式对抗网络GAN：生成器网络G + 判别器网络D
- 核心思想：利用判别器判断生成器生成的数据是否足够的真实
  - 训练生成器，使其生成的数据尽量让判别器判断不出真假
  - 训练判别器，使其尽量能区分真实数据与生成的数据

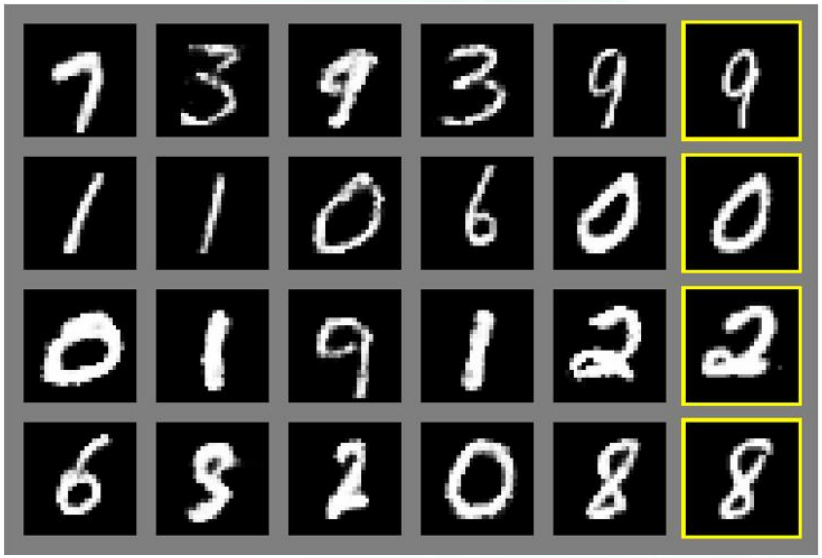


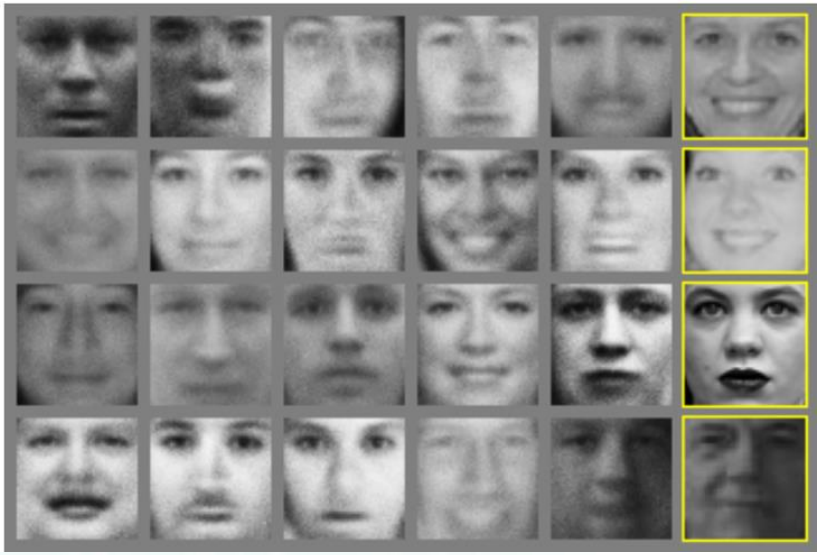
群模乱舞之GAN：初始模型

- 目标函数

真实数据 ↓ 生成的数据 ↓

$$\min_{G_\theta}\max_{D_w}\{\mathbb{E}_{x\sim P_r}[\log D_w(x)]+\mathbb{E}_{z\sim P(z)}[\log(1-D_w(G_\theta(z)))]\}$$

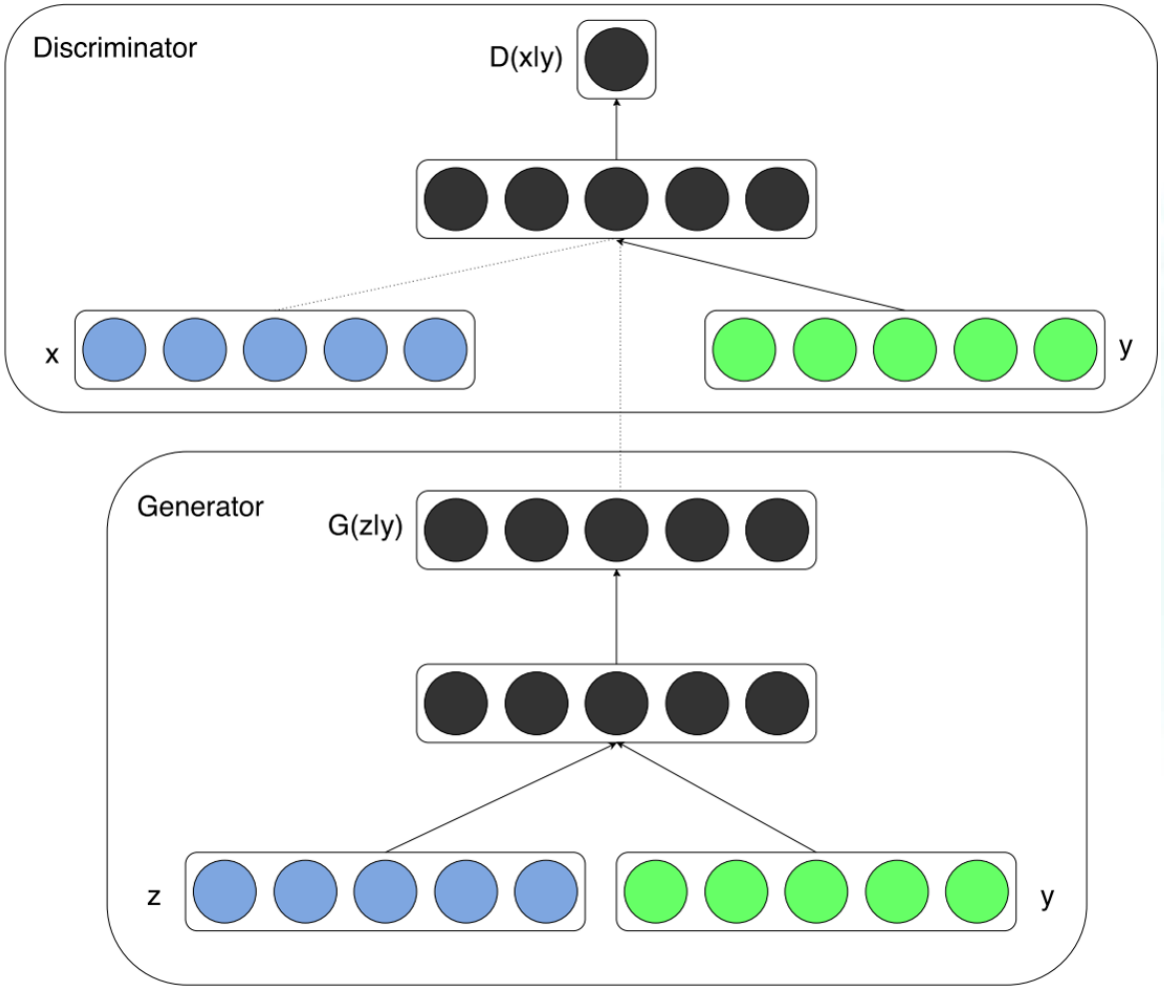




黄色框中图像为训练好的GAN生成的数据；
每行其它数据是与黄色框数据最相似的几个真实数据

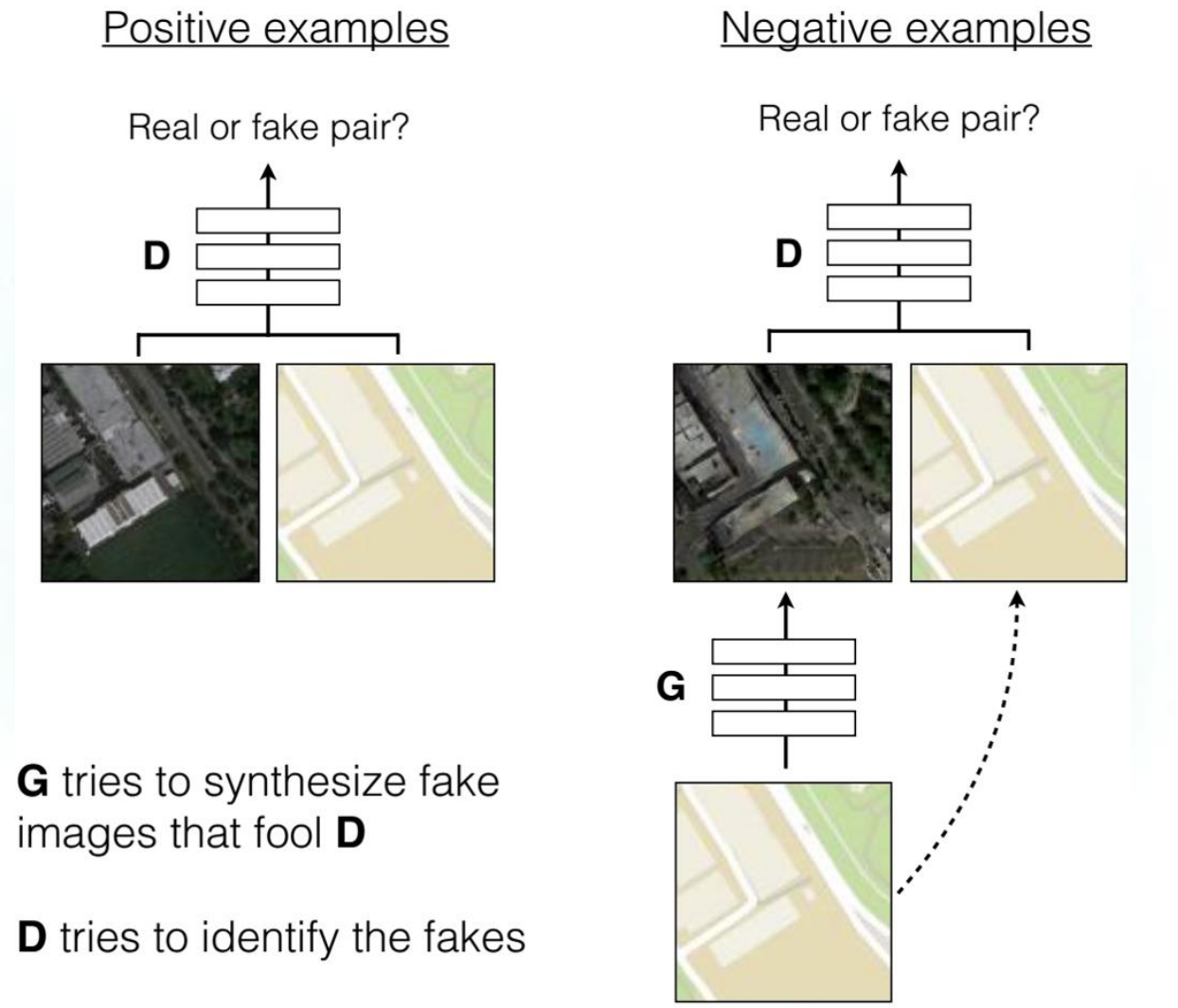
群模乱舞之GAN： Conditional GAN

- Condition ‘y’ 作为生成器和判别器输入的一部分
- ‘y’可以是向量，也可以是图像等更复杂的表示



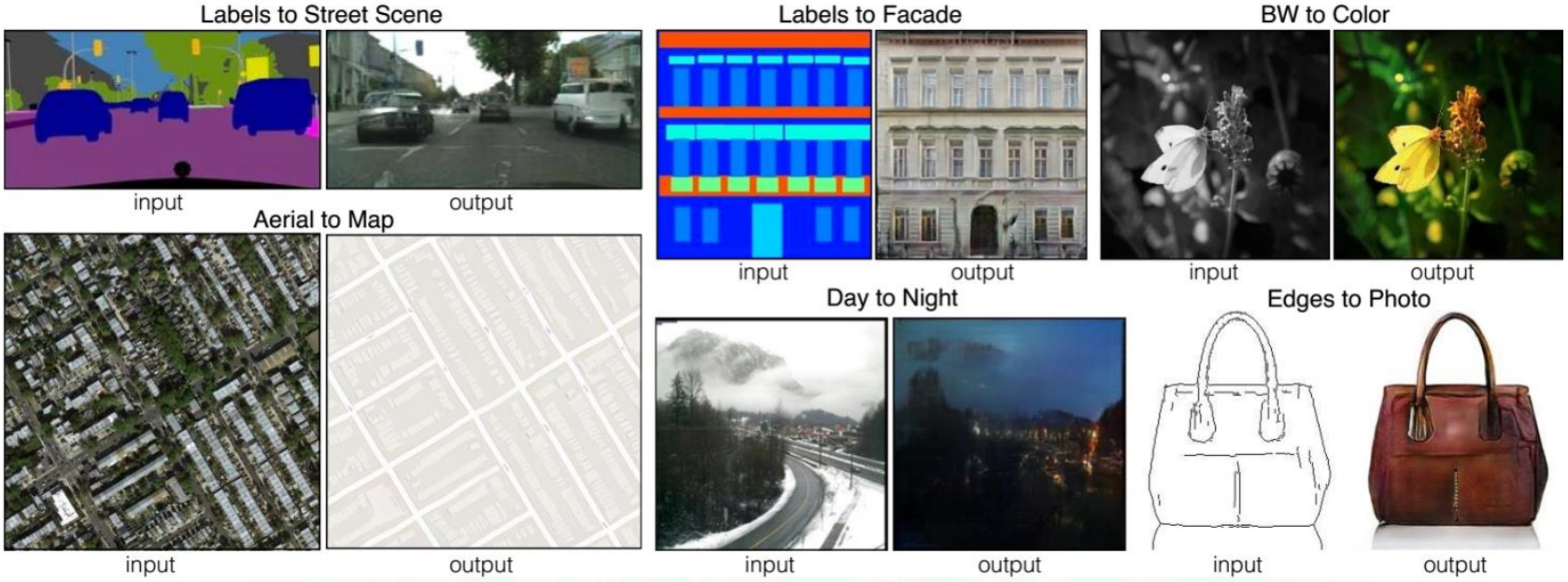
群模乱舞之GAN： Conditional GAN

- CGAN用于图像翻译



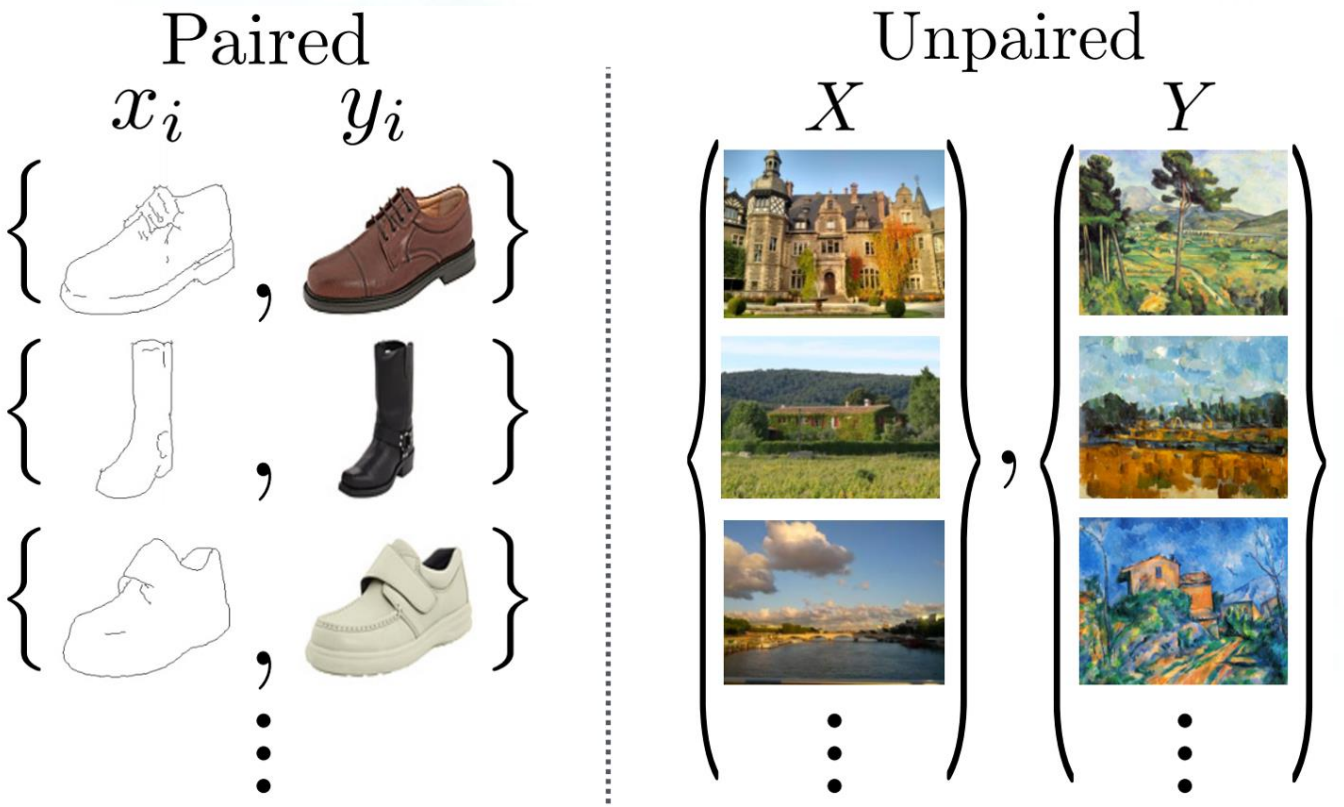
群模乱舞之GAN：Conditional GAN

- CGAN用于不同功能的图像翻译：黑白->彩色；素描->实物，等



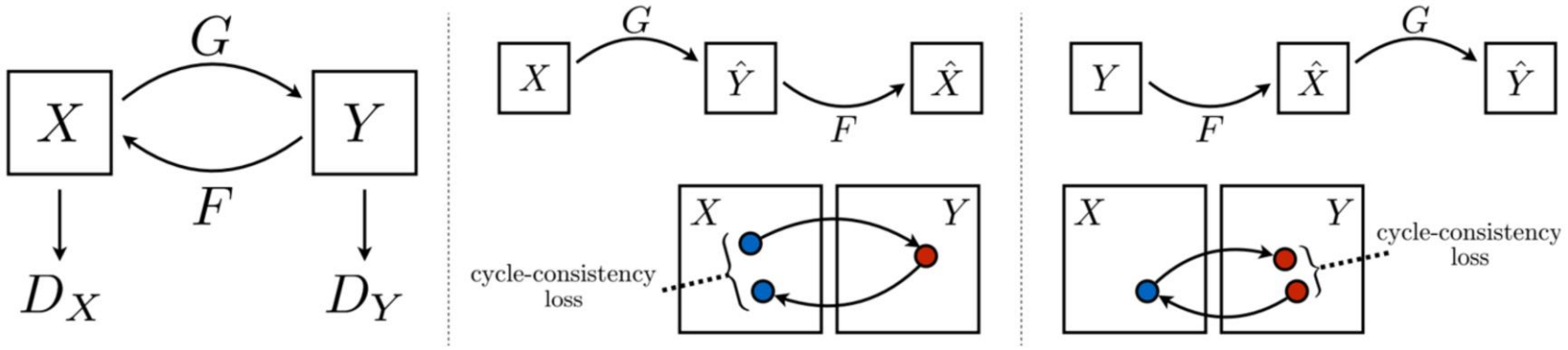
群模乱舞之GAN：CycleGAN

- 更加挑战情况下图像翻译：不存在一一对应的训练数据（右图）



群模乱舞之GAN：CycleGAN

- 思路：将图像从一个domain转换到另一个domain，然后再转换回来，得到的图像与原图像尽量一样；两个GAN模型

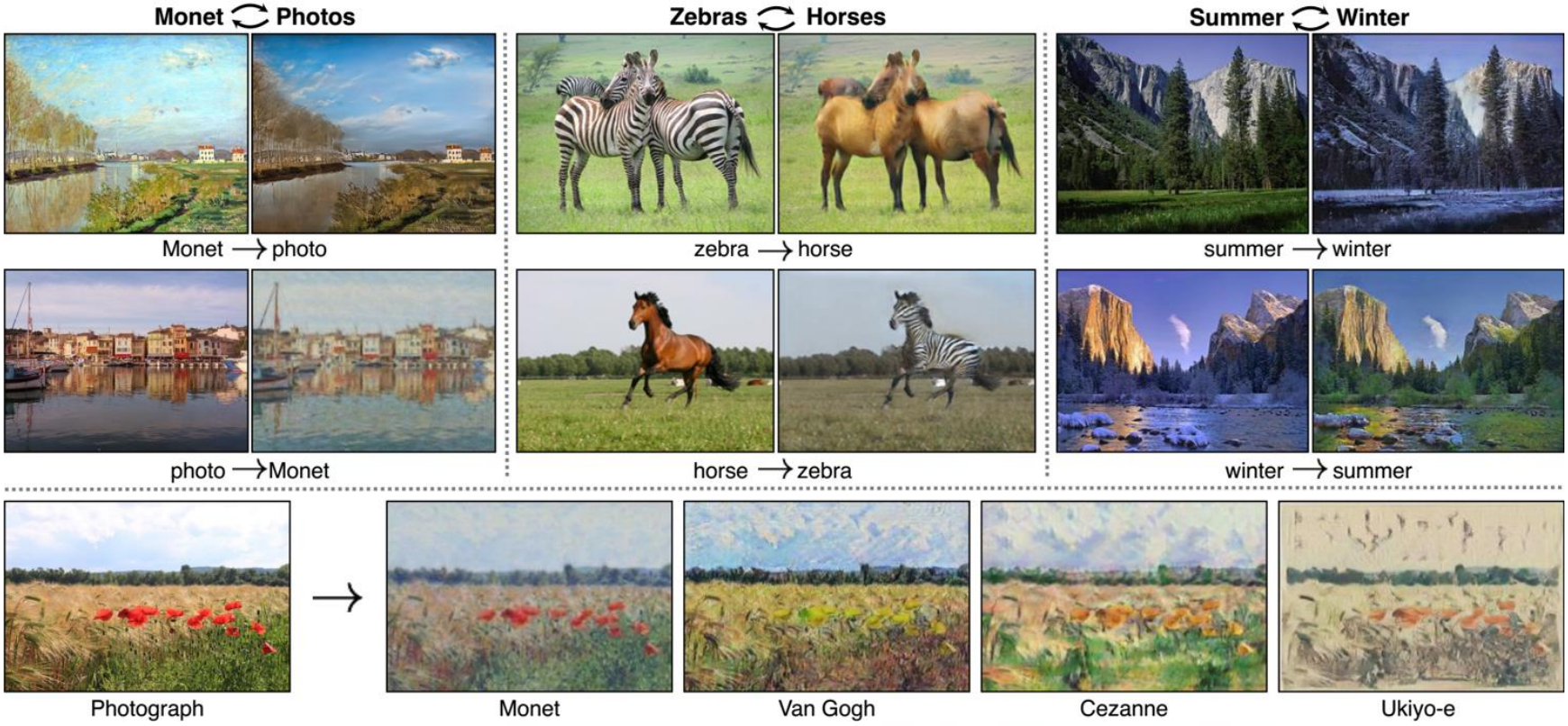


$$\mathcal{L}_{\text{cyc}}(G,F) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y\sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1]$$

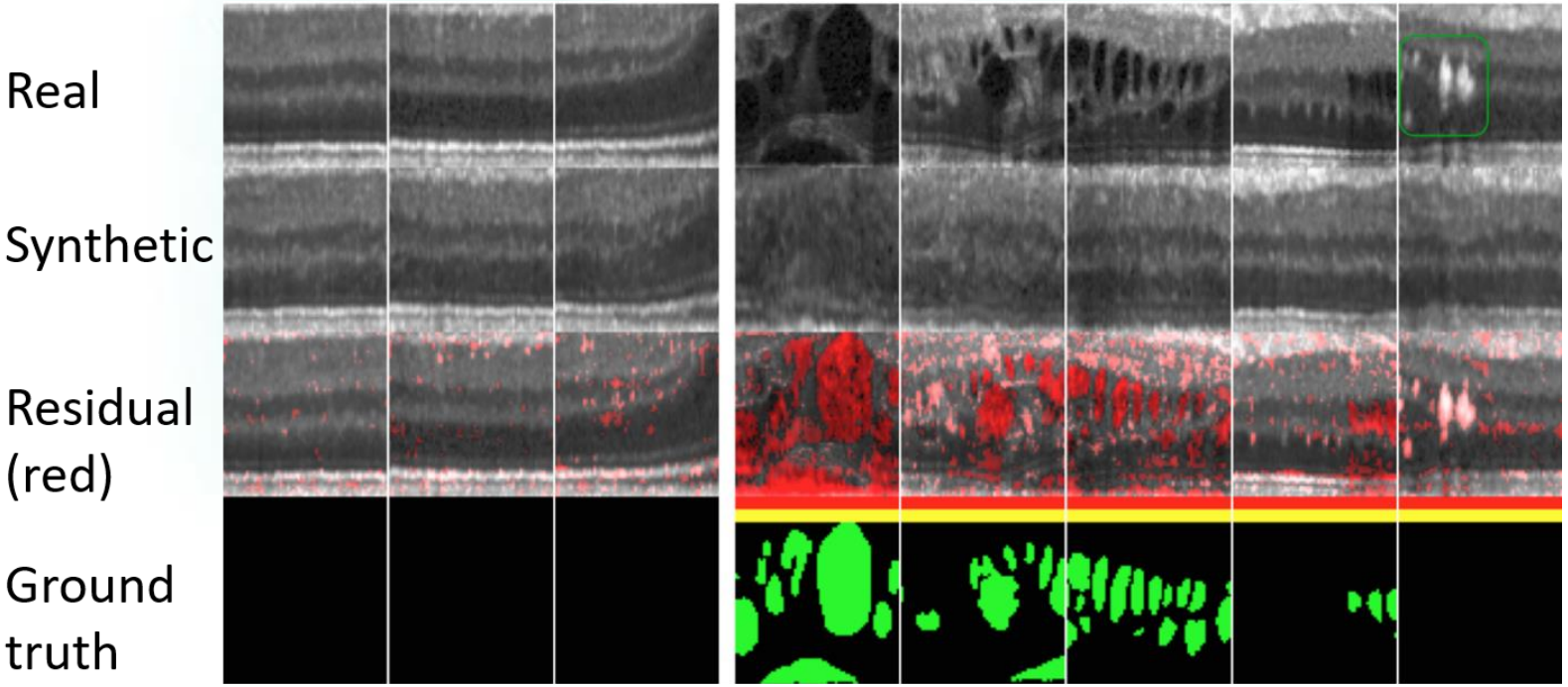
$$\mathcal{L}(G,F,D_X,D_Y) = \mathcal{L}_{\text{GAN}}(G,D_Y,X,Y) + \mathcal{L}_{\text{GAN}}(F,D_X,Y,X) + \lambda\mathcal{L}_{\text{cyc}}(G,F),$$

群模乱舞之GAN：CycleGAN

- CycleGAN用于图像风格转移



群模乱舞之GAN：更多应用

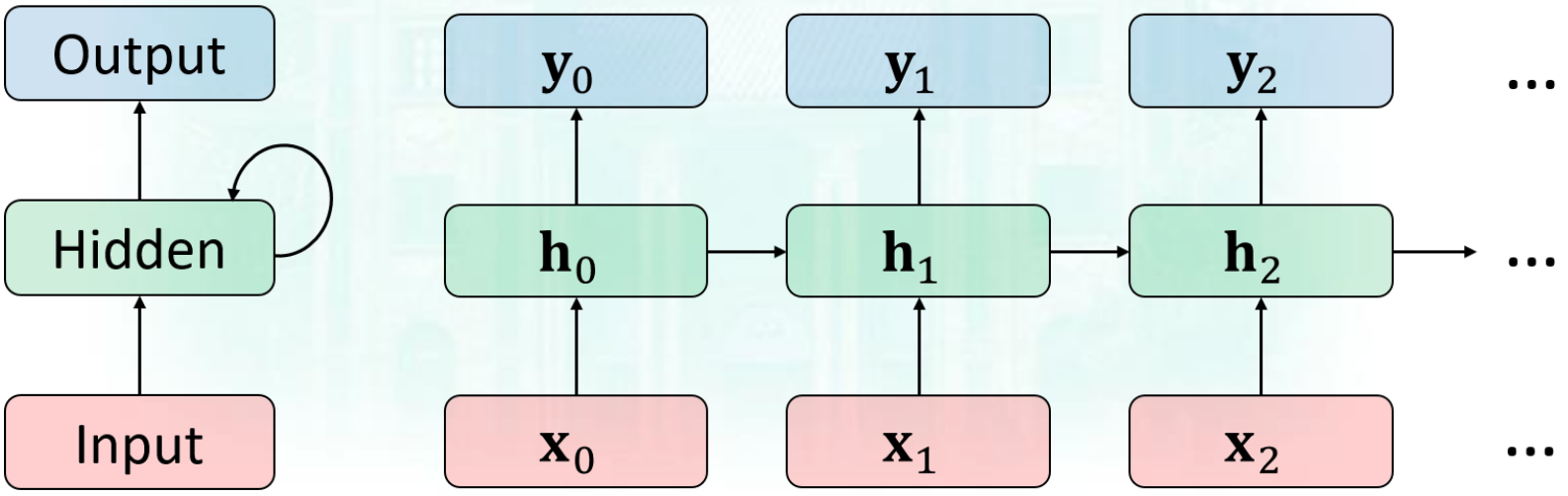


群模乱舞：自然语言处理（NLP）任务

- 文献分类
- 观点分析
- 机器翻译
- 阅读理解
- 聊天机器人
- 个人助手
- 文章总结
- . . .

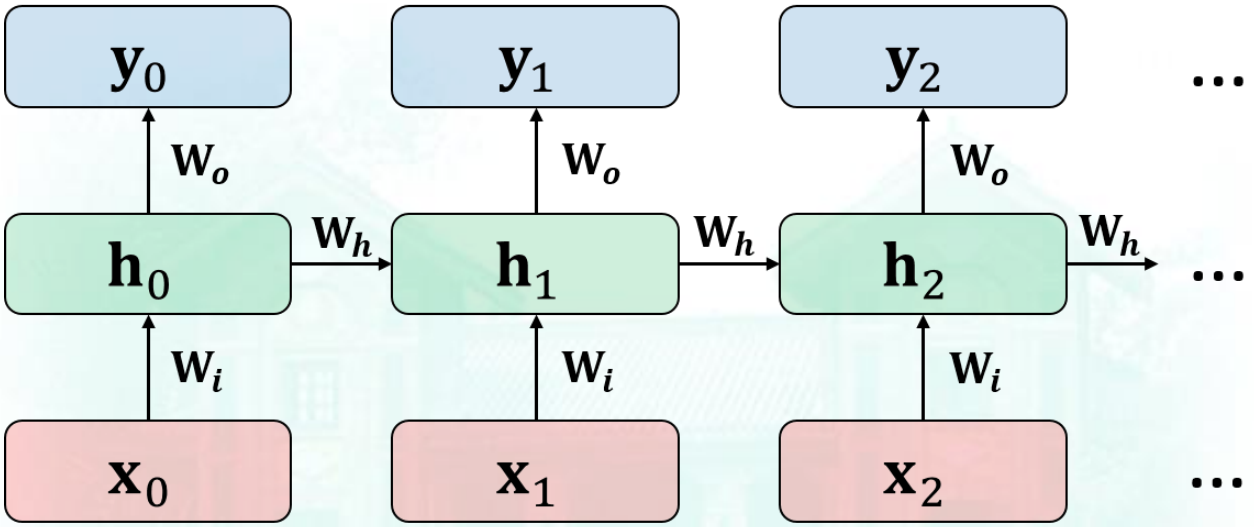
群模乱舞之序列模型：原始RNN

- 循环神经网络(Recurrent Neural Network)：每个时刻隐含层的输出作为下一个时刻隐含层的部分输入
- 在时间维度展开可对序列数据进行分析
- RNN是个深度神经网络么？



群模乱舞之序列模型：原始RNN

- 模型参数在不同时刻被共用！

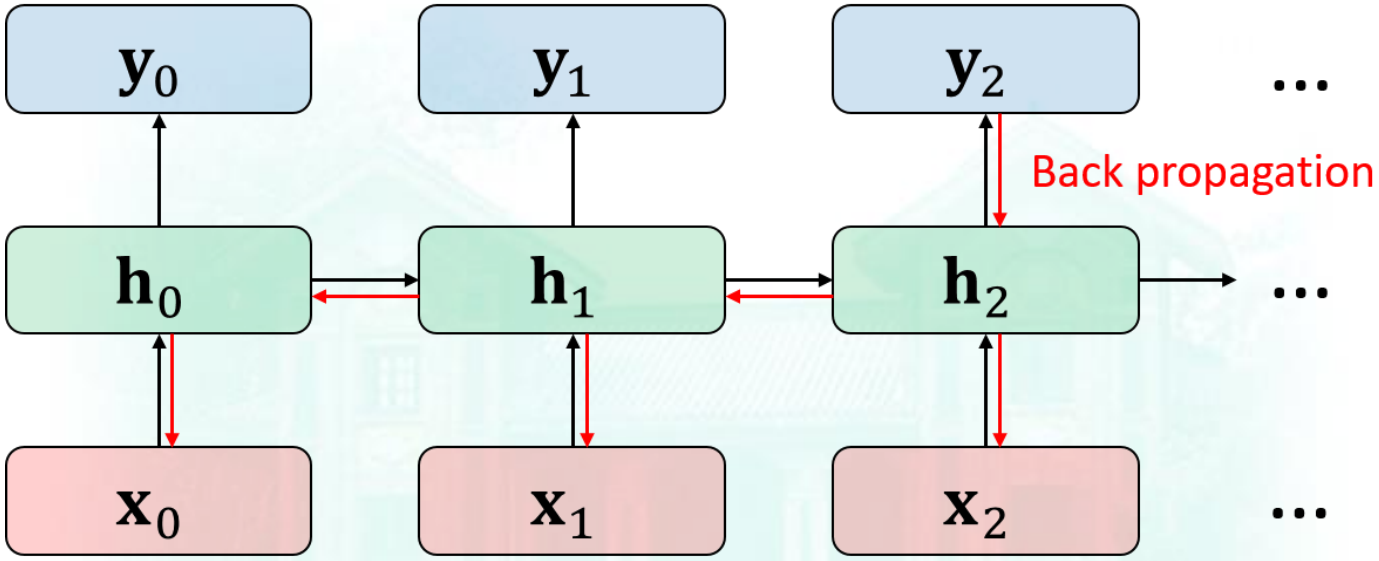


$$\mathbf{h}_t = g(\mathbf{W}_h\mathbf{h}_{t-1} + \mathbf{W}_i\mathbf{x}_t + \mathbf{b}_h)$$
$$\mathbf{y}_t = \sigma(\mathbf{W}_o\mathbf{h}_t + +\mathbf{b}_o)$$

g：激活函数，σ：softmax函数

群模乱舞之序列模型：原始RNN

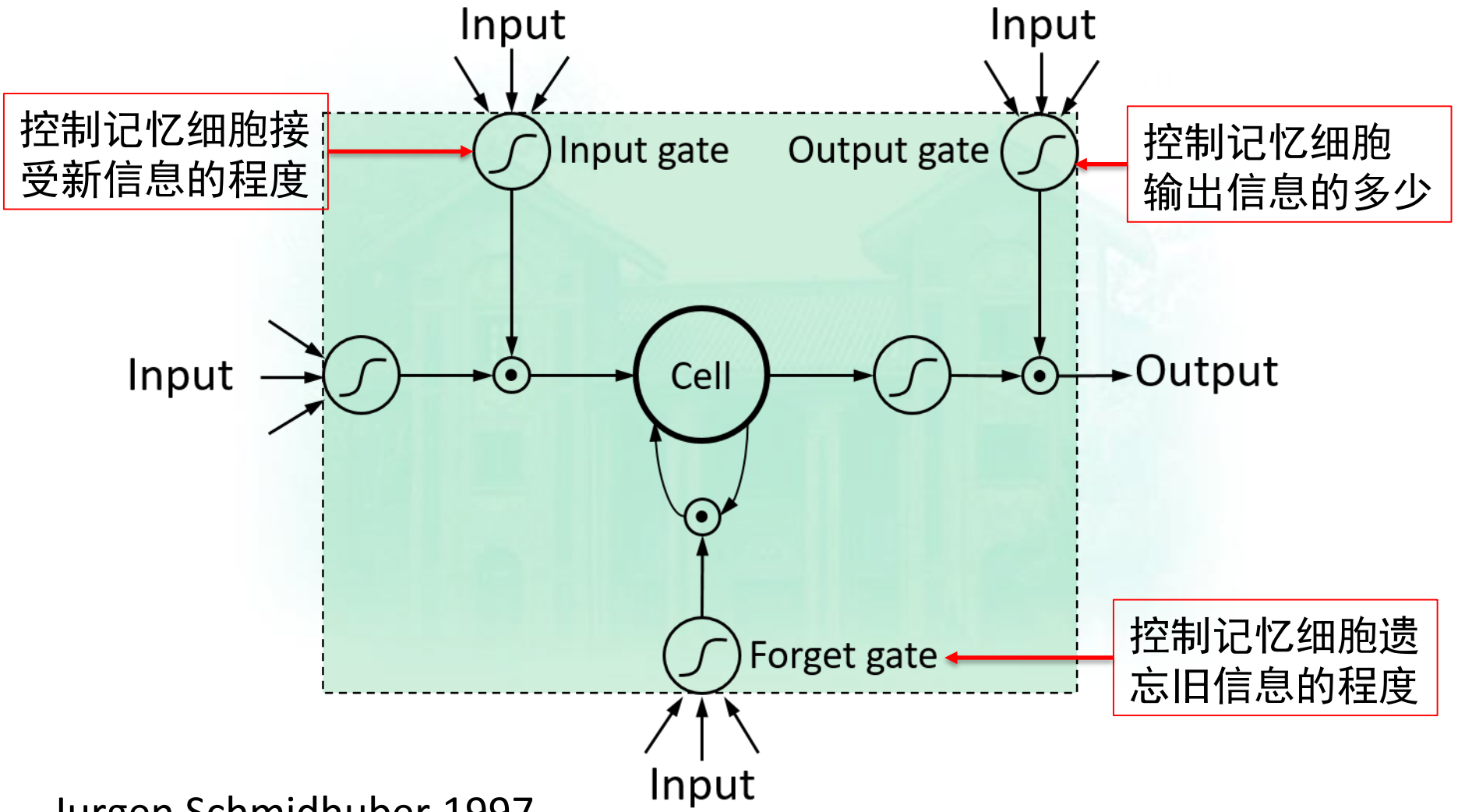
- 训练RNN：也是使用梯度下降法（反向传播算法）



- 梯度消失导致RNN不能有效学习序列里间隔较远信息间的关系

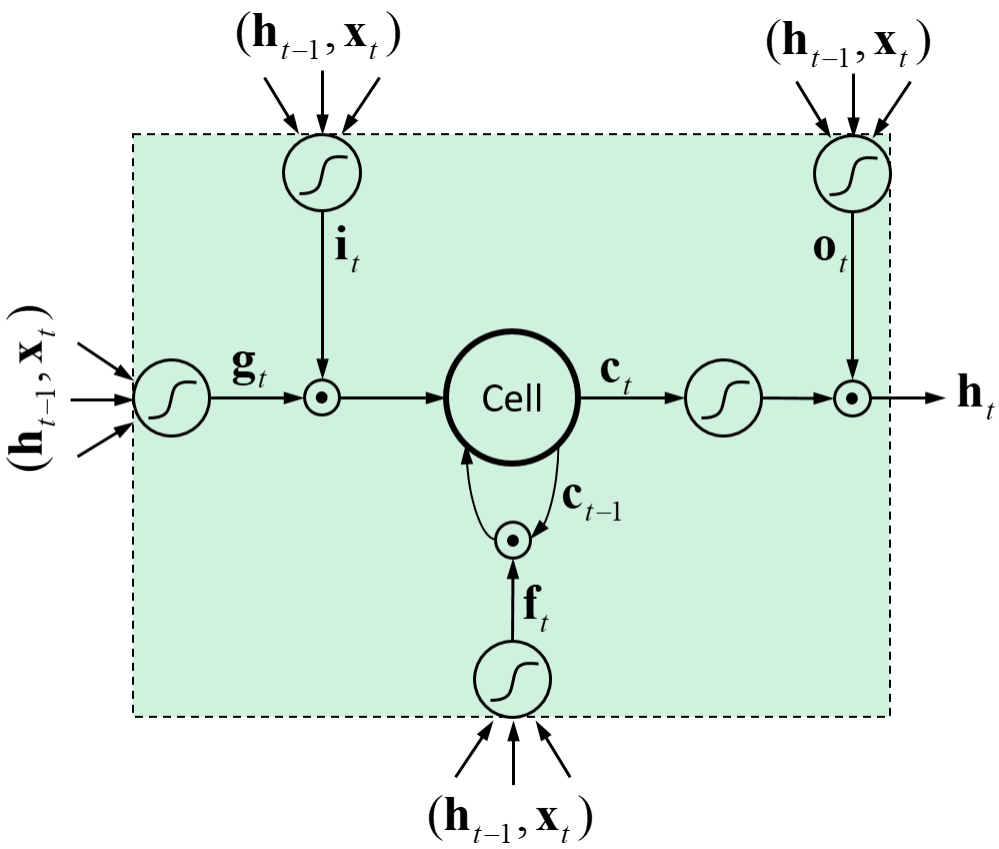
群模乱舞之序列模型：LSTM

- Long short-term memory (LSTM)



Jurgen Schmidhuber,1997

群模乱舞之序列模型：LSTM



三个控制门

$$\mathbf{i}_t = \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{b}_i)$$
$$\mathbf{f}_t = \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{b}_f)$$
$$\mathbf{o}_t = \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{b}_o)$$
$$\mathbf{g}_t = tanh(\mathbf{W}_g\mathbf{x}_t + \mathbf{U}_g\mathbf{h}_{t-1} + \mathbf{b}_g)$$
$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t$$
$$\mathbf{h}_t = \mathbf{o}_t \odot tanh(\mathbf{c}_t)$$

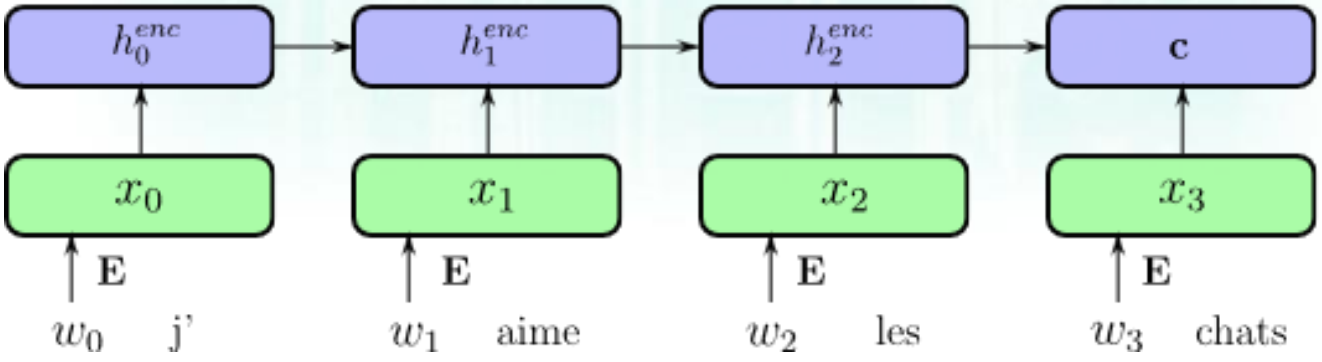
输出信息

记忆细胞信息

新信息

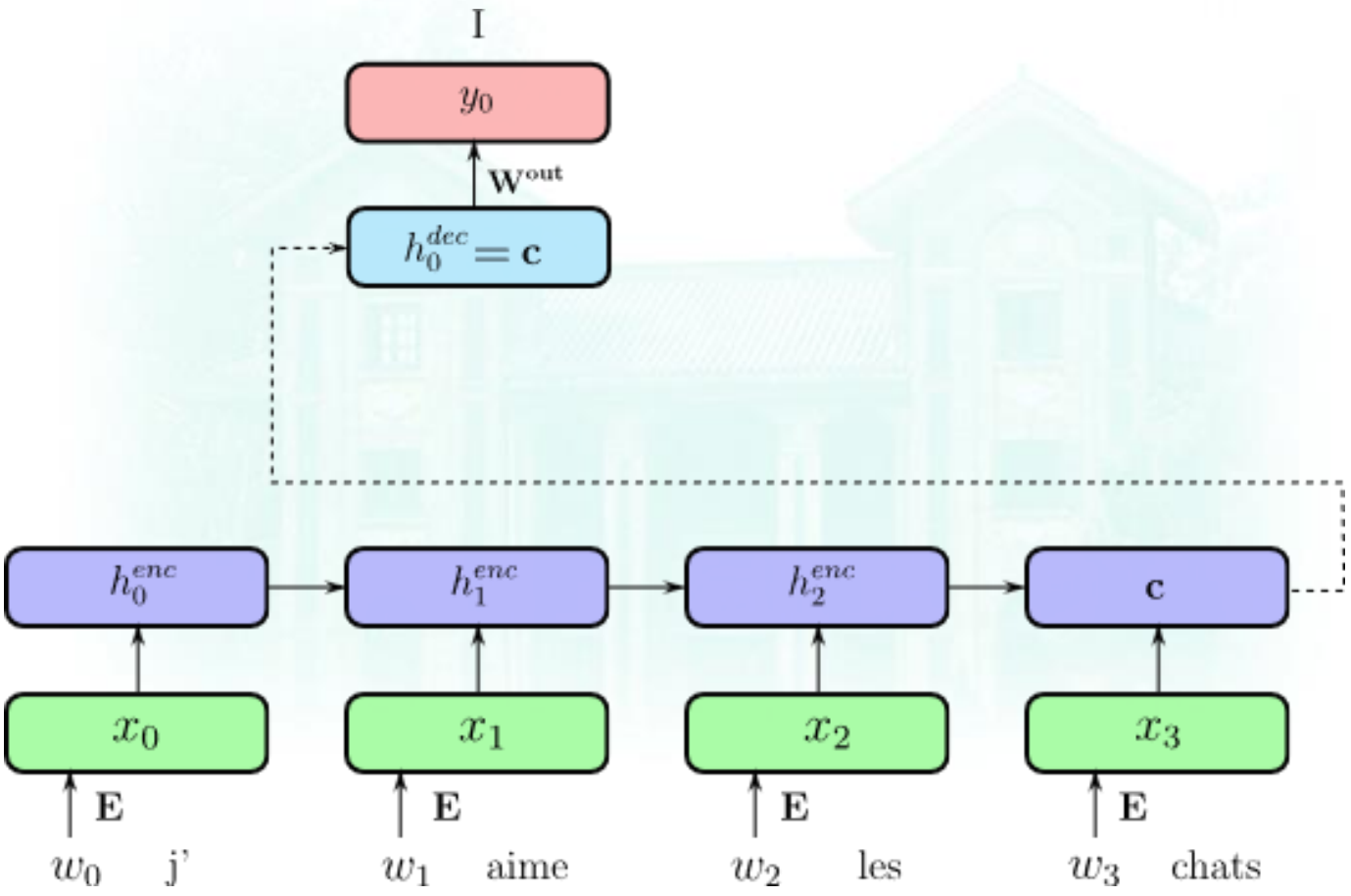
群模乱舞之序列模型：机器翻译

- 编码器-译码器模型：编码器（RNN/LSTM）对源域的句子进行编码...



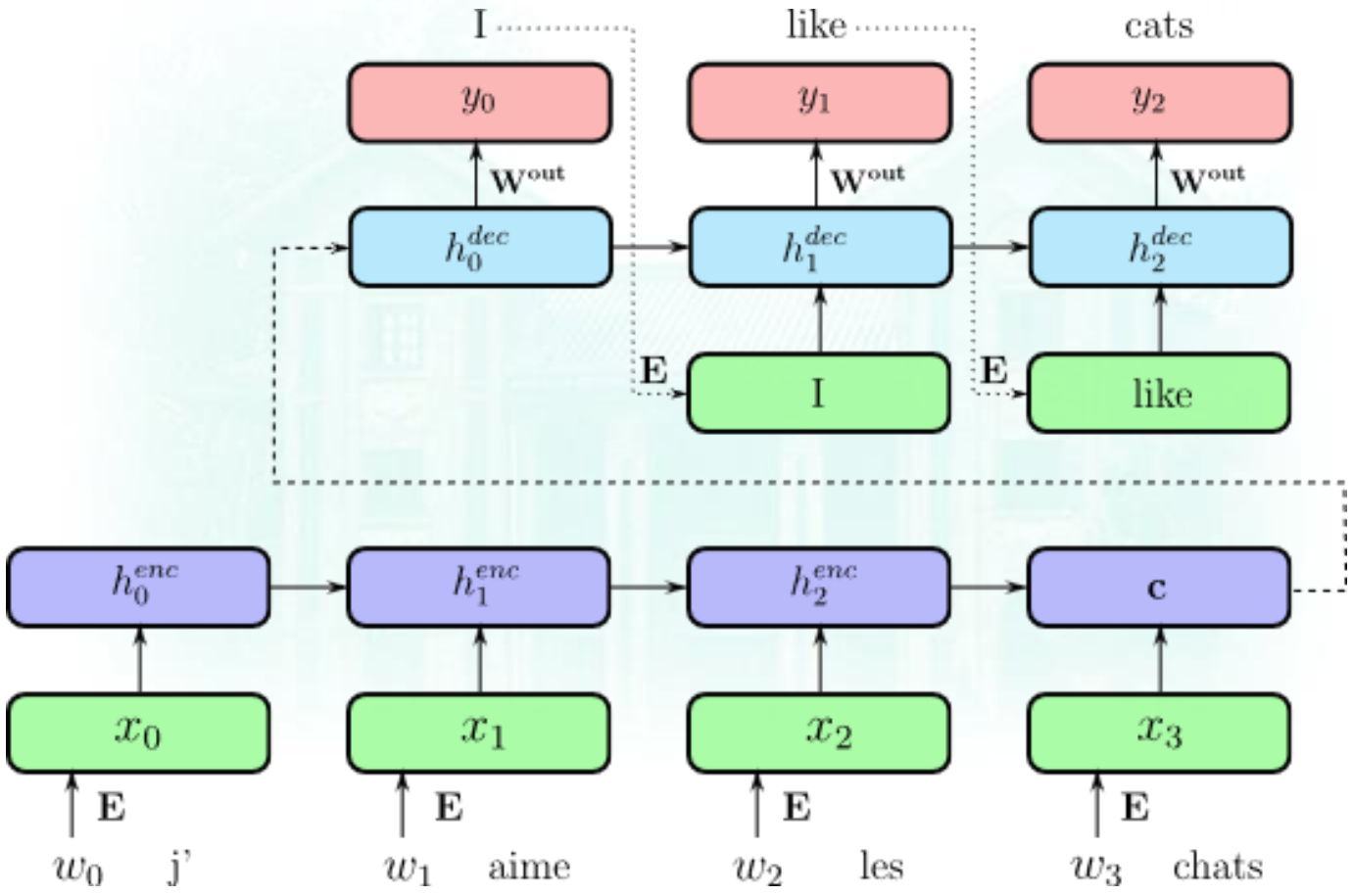
群模乱舞之序列模型：机器翻译

- 编码器-译码器模型：...译码器(另一个RNN/LSTM)基于编码器最后时刻信息预测目标域中第一个单词



群模乱舞之序列模型：机器翻译

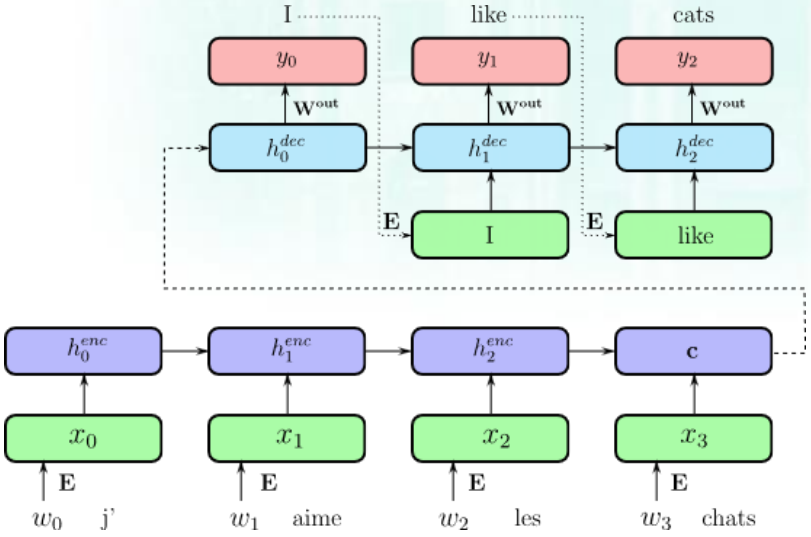
- 编码器-译码器模型：...译码器基于上一时刻译码器输出及隐含层信息预测目标域中下一个单词



群模乱舞之序列模型：机器翻译

- 编码器–译码器模型训练
  - 训练集：｛（源域句子，目标域对应句子）｝
  - 损失函数：Cross-entropy测量模型预测的每一个单词与正确单词的相似性

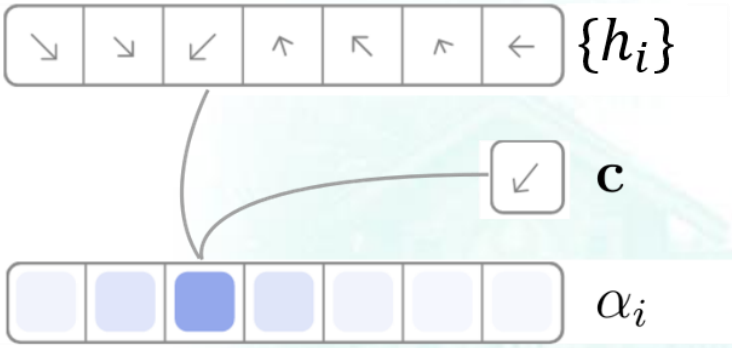
- 编码器–译码器模型预测
  - 译码器上一时刻多个预测分别作为当前时刻（部分）输入
  - 从中选择当前时刻译码器多个预测，分别用于下一个时刻的输入



缺点：译码器只用到编码器最后时刻的输出信息！

群模乱舞之序列模型：注意力机制

- h_i表示源域中第i个单词相关的信息
- $\mathbf{c}$ 表示译码器中当前时刻隐含层的信息



$$e_i = f(h_i, \mathbf{c})$$

$$\alpha_i = \frac{\exp(e_i)}{\sum_k \exp(e_k)}$$

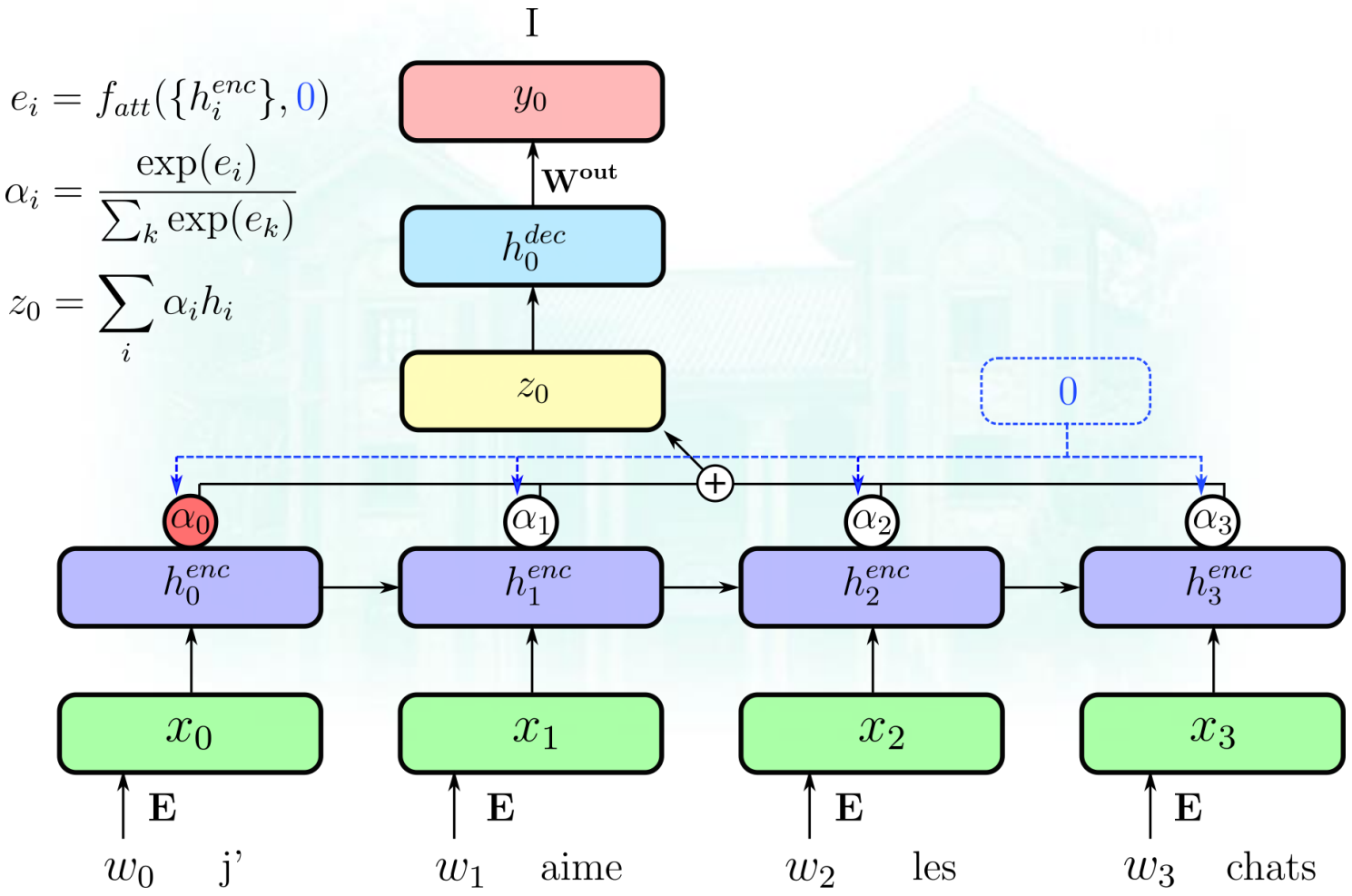
整个过程（函数）连续可导，可自然潜入模型训练中

基于注意力大小，提取源域中的信息，帮助预测下一个单词。

预测目标域中下一个单词时，源域中每个单词的相关性，即模型对每个单词的注意力大小。

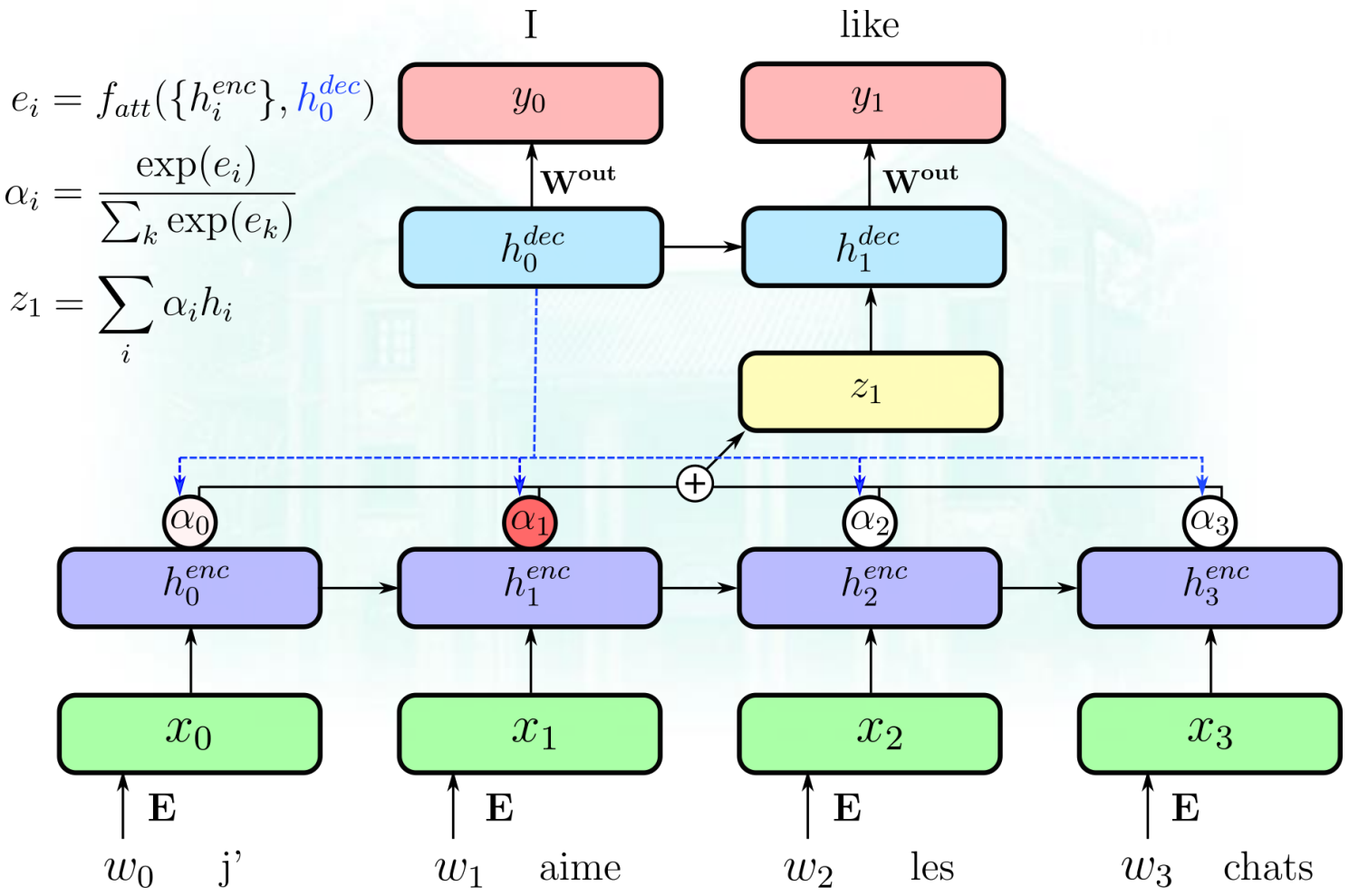
群模乱舞之序列模型：注意力机制

- 预测第一个单词：译码器中初始信息与编码器中每一时刻的编码信息用于估计注意力，注意力调节后的编码信息输入译码器



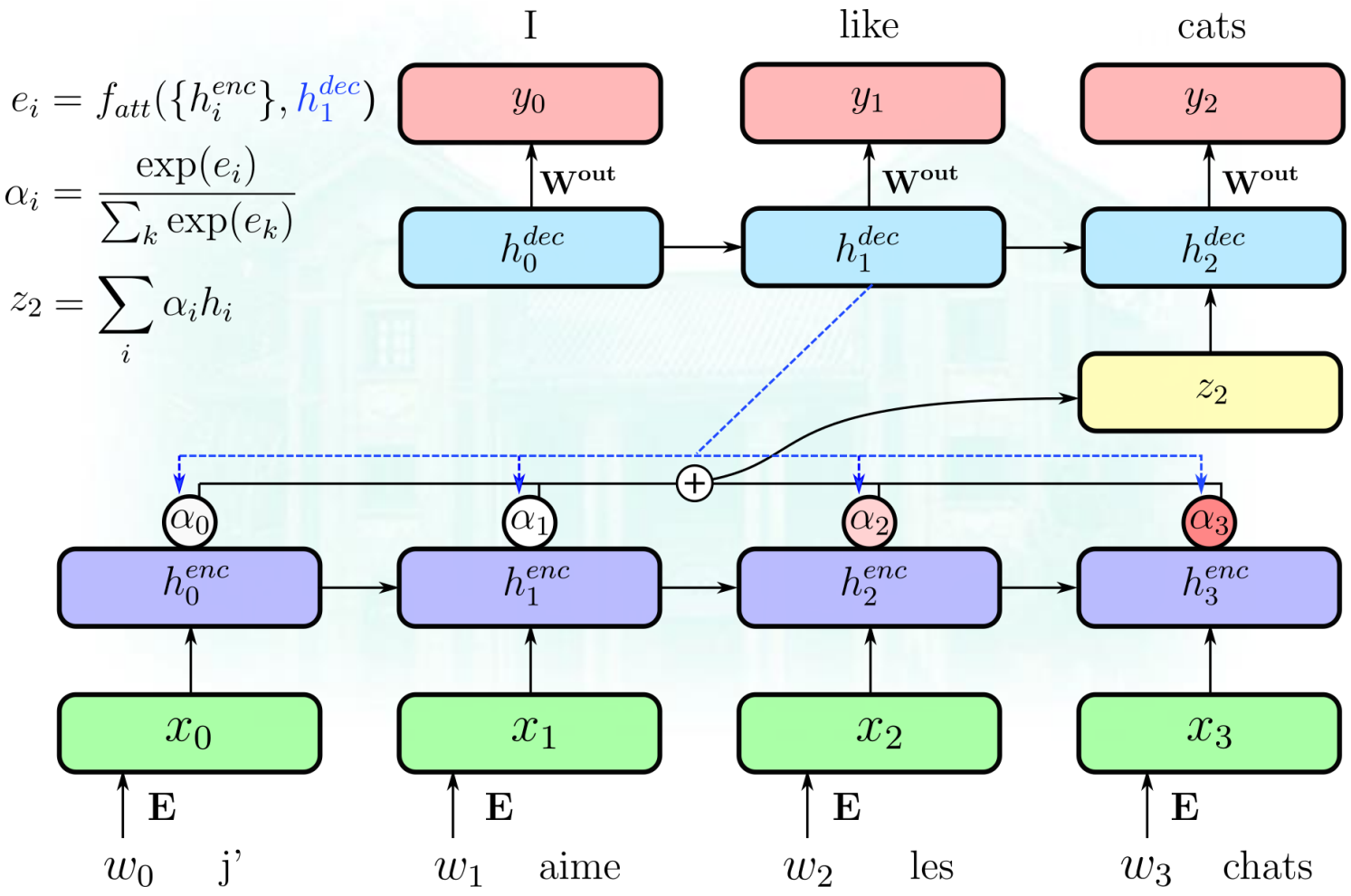
群模乱舞之序列模型：注意力机制

- 预测第二个单词：译码器上一时刻的信息与编码器中每一时刻编码信息来估计注意力，注意力调节后的编码信息输入译码器



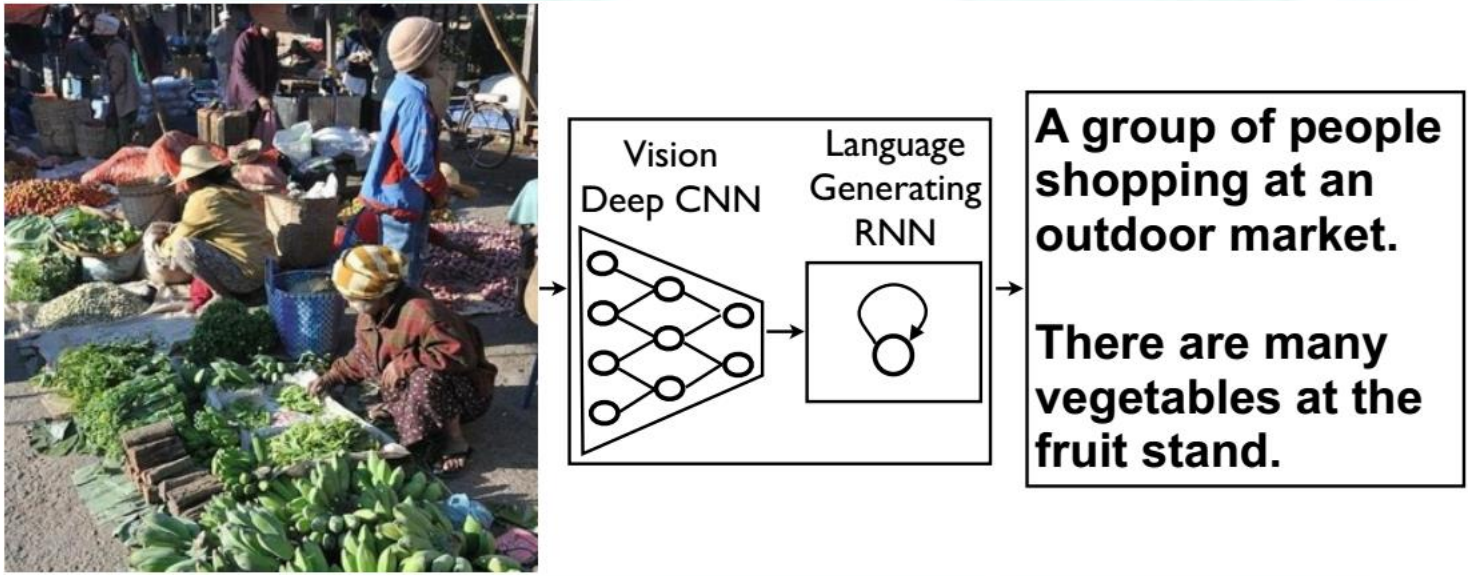
群模乱舞之序列模型：注意力机制

- 预测第三个单词：类似



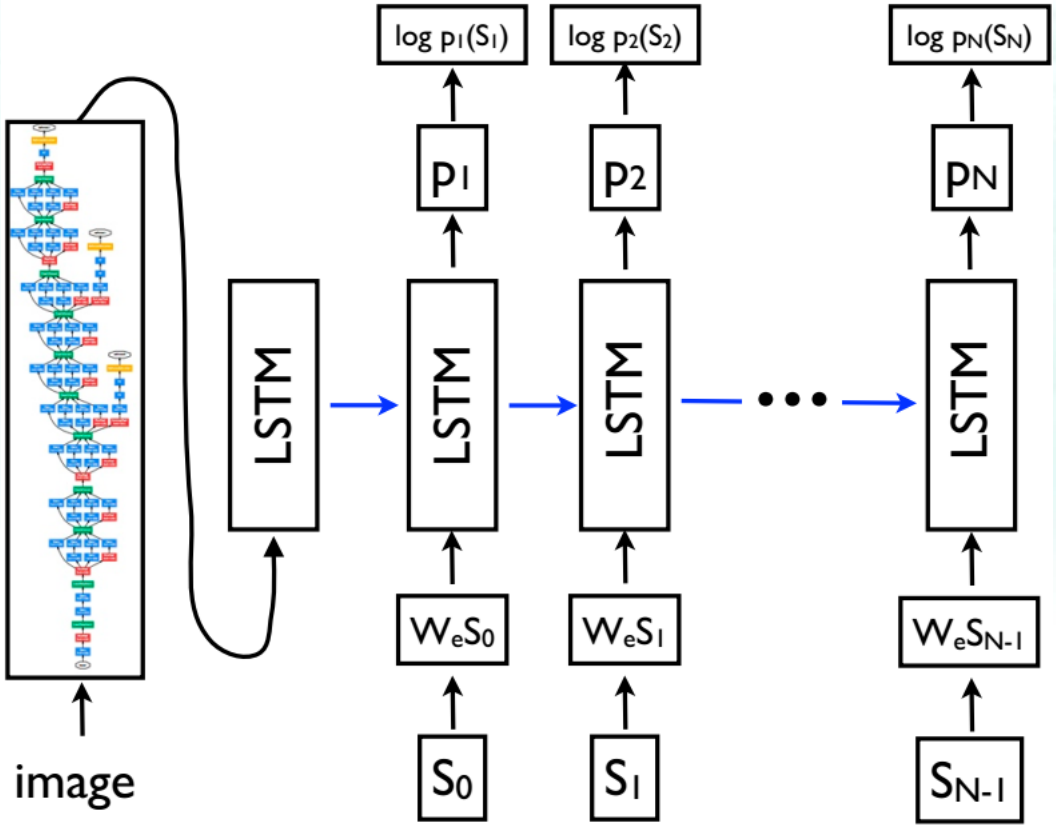
群模乱舞之序列模型：看图说话

- Image captioning：给一幅图像，用一句话描述图像内容
- 可看作一个特殊的翻译问题：由图像翻译为句子
- 所以，解决思路与机器翻译几乎一模一样！



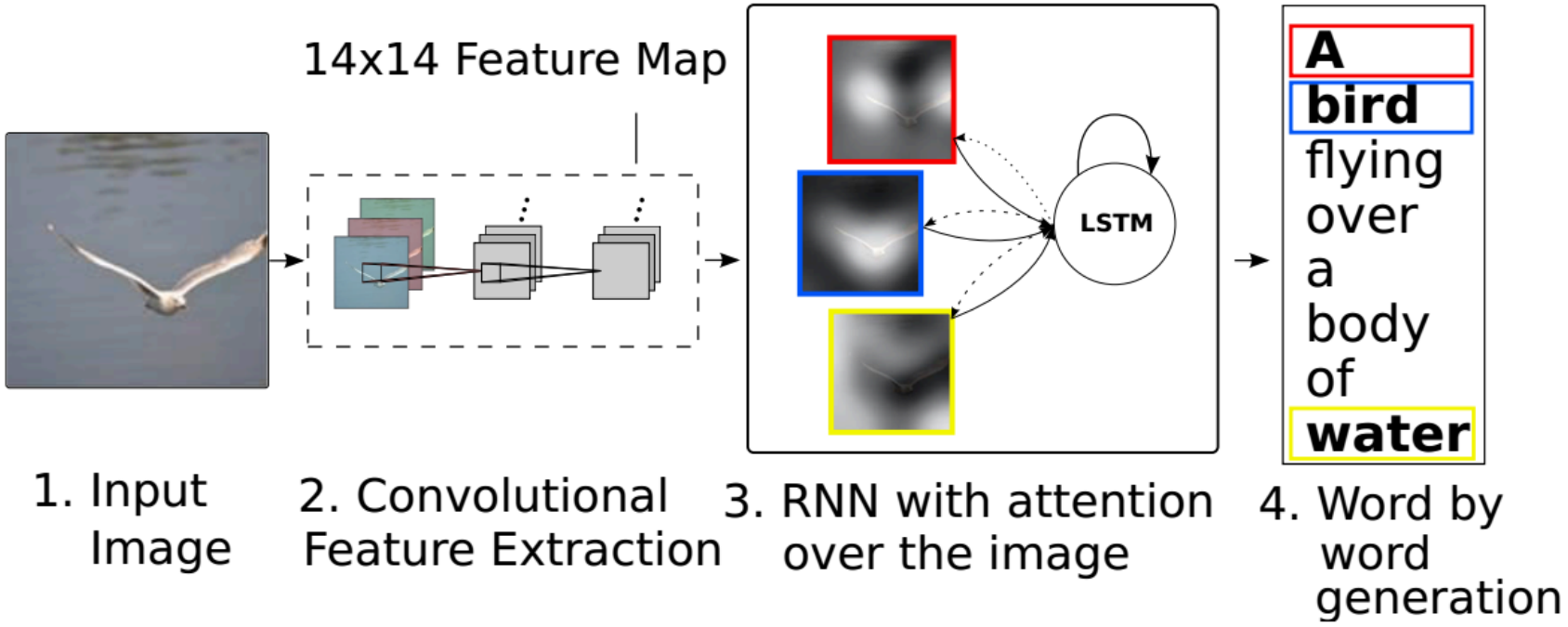
群模乱舞之序列模型：看图说话

- 编码器-译码器模型：
  - 编码器：CNN特征提取器替换掉机器翻译中的RNN编码器
  - 译码器：与机器翻译中的RNN一样
  - 编码器固定，只训练译码器



群模乱舞之序列模型：看图说话

- 含有注意力机制的看图说话
- 机器翻译模型中注意力机制的简单应用
  - 最后特征图大小为14×14，所以原域中有196个“words”



寻万能之源：为什么超越其它方法？

- 特定任务下的学习：自动学到任务相关的特征
- 深度模型 -> 多层次特征，更复杂非线性关系

为什么最近几年深度学习模型才表现更好？

- 大数据
- 硬件运算速度，多核并行
- 模型结构创新：ResNet,GAN,Memory N,GCN,...
- 模块/操作创新：BN,ReLU,dropout,skip, deconv, attention,...
- 算法创新：初始化，优化，损失函数，强化学习，...
- 研发平台：TensorFLow, PyTorch, ...
- 应用驱动：无人驾驶，安防，个人助手，智慧城市与医疗

深度学习可以解决所有任务了？

既然DL这么厉害，那我们还研究什么？

只是比传统方法相对好点而已！并不完美！

深度学习并不Perfect

需要大数据量 —数据少时怎么办?→ 小样本学习

数据标注费时费力 —无标注数据有用么?→ 无/自监督学习

训练测试计算量大 —手机等终端怎么用?→ 模型轻量化

测试场景单一 —场景变化时怎么办?→ 模型鲁棒性

预测过程不清楚 —要求预测证据怎么办?→ 模型可解释性

人工设计模型结构 —能否减少人工因素?→ 自动机器学习

持续学习能力差 —能否使用中提升能力?→ 终身学习

……