



# 人工智能与医学

## -计算机视觉算法

向孙程

主页: <https://JeremyXSC.github.io/>

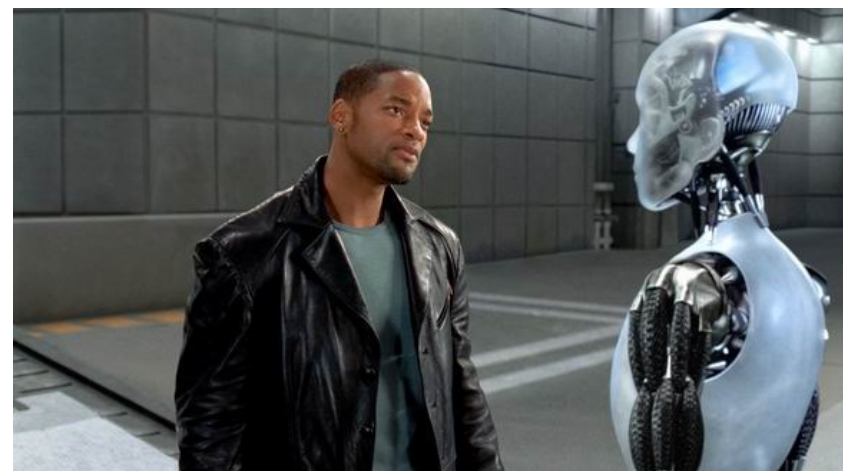
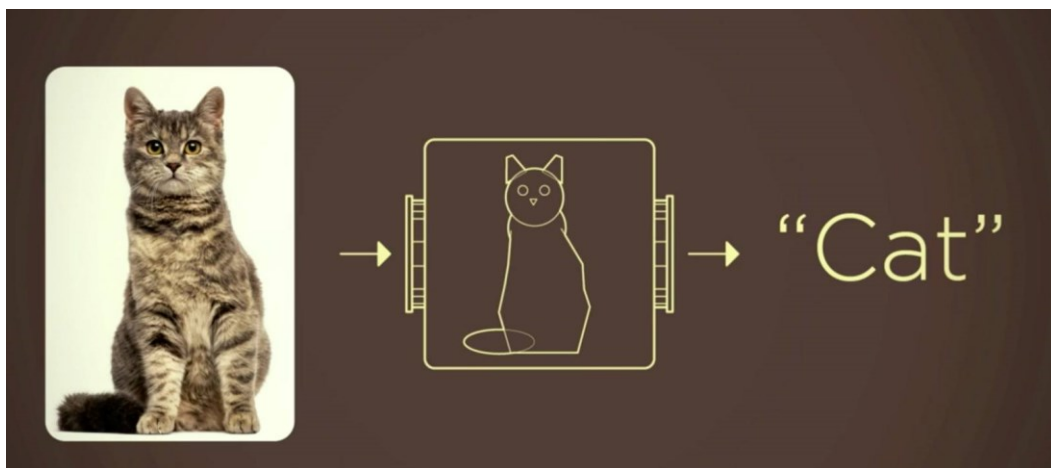
上海交通大学医学院课程

2023春季



# 计算机视觉定义

- 计算机视觉是使用计算机及相关设备对生物视觉的一种模拟。它的主要任务就是通过对采集的图片或视频进行处理以获得相应场景的认知信息——让机器代替人感知世界。





# 工业界应用领域





# 工业界应用领域



关键技术

深度学习/机器学习、计算机视觉、语音技术/自然语言处理、人机交互

基础支撑

传感器、芯片、大数据、云计算、服务商、生物识别



# 学术界受关注程度（数据库和竞赛）

- ImageNet: 斯坦福计算机科学家李飞飞模拟人类视觉系统建立的数据库

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

- ILSVRC Challenge

<https://image-net.org/challenges/LSVRC/>

- MIMIC: 医学报告多模态数据库

<https://physionet.org/content/mimic-cxr-jpg/2.0.0/>

- HUST-19: 新冠肺炎CT影像和临床信息数据库

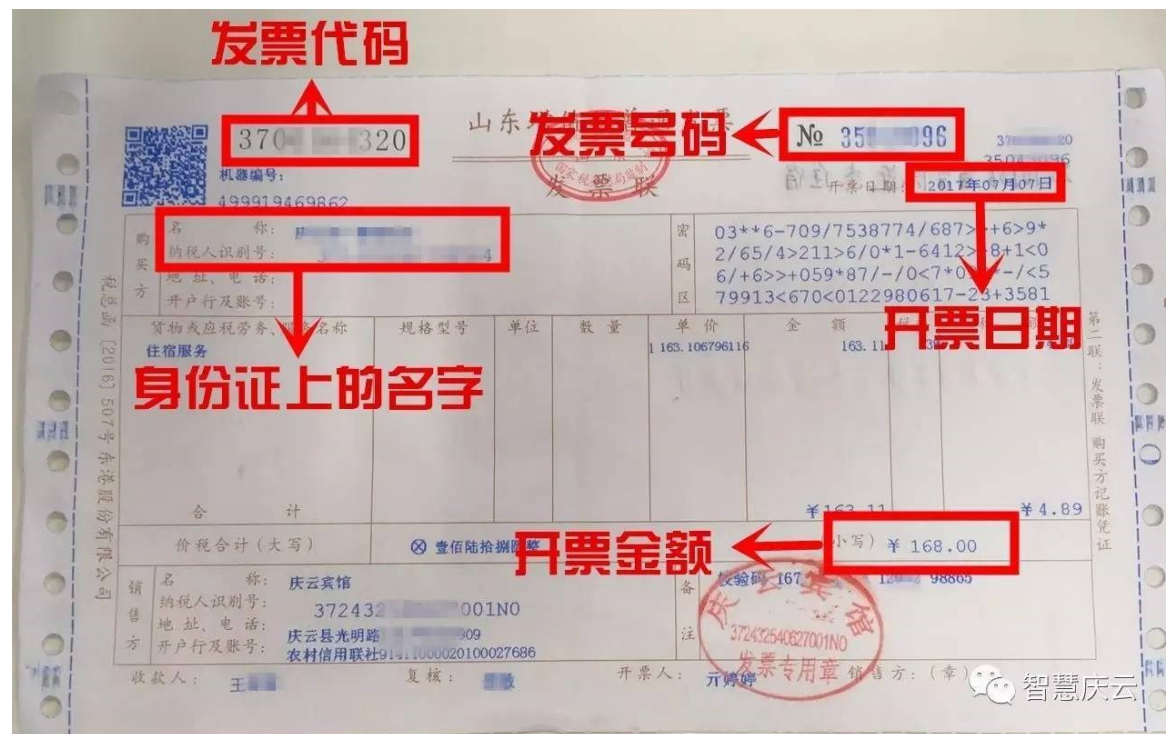
<http://ictcf.biocuckoo.cn/HUST-19.php>



# 计算机视觉应用：光学字符识别 OCR



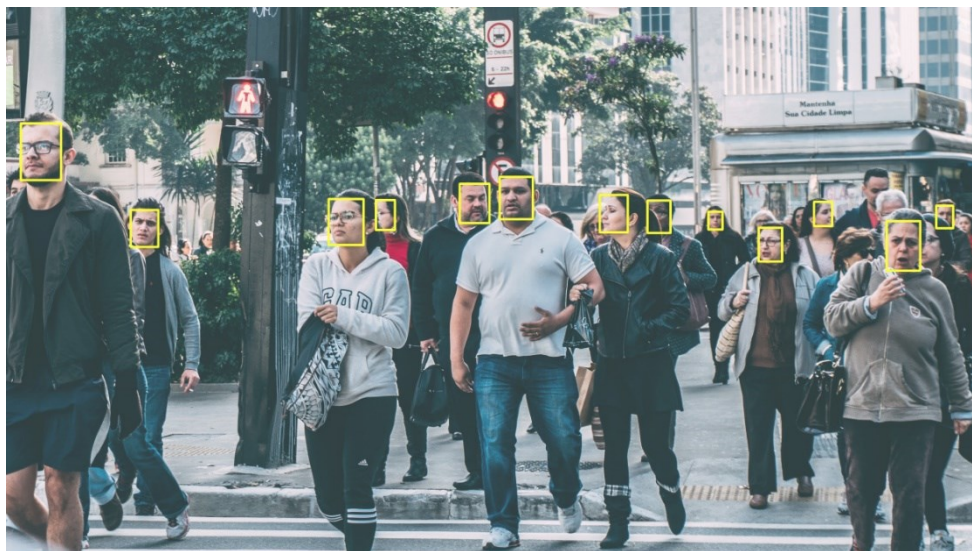
## 车牌识别



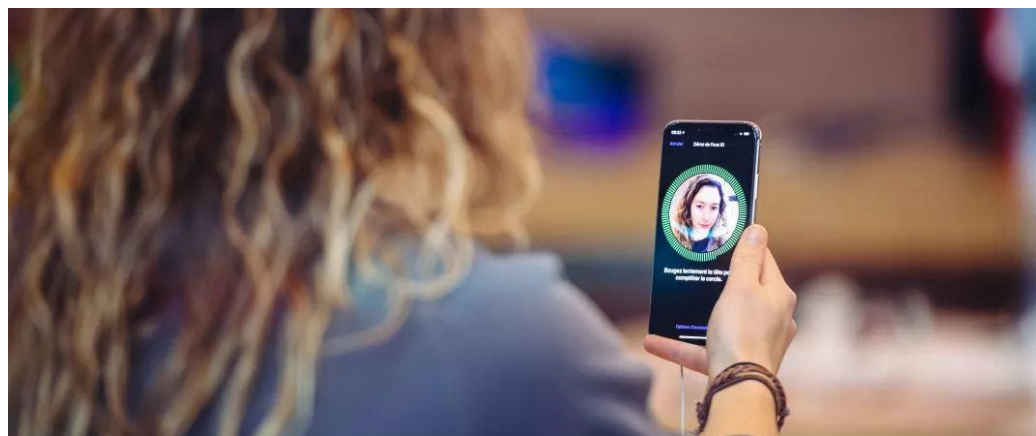
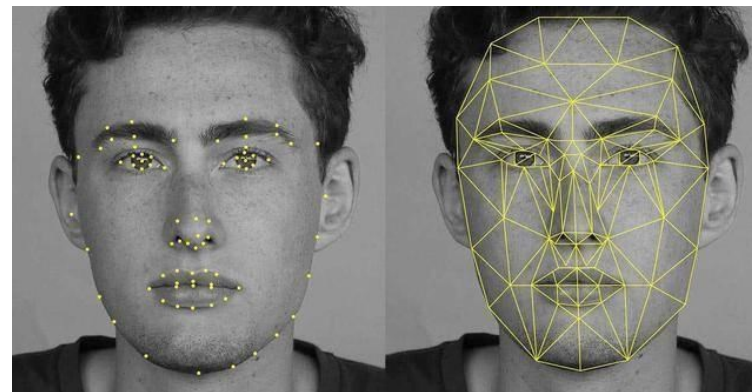
## 发票识别



# 计算机视觉应用：人脸检测和识别



人脸检测



人脸识别



# 计算机视觉应用: 图像修复



图像自动修复和上色





# 计算机视觉应用: 妆容迁移



图像自动修复和上色



# 计算机视觉应用：图像高分重建

bicubic  
(21.59dB/0.6423)



SRResNet  
(23.53dB/0.7832)



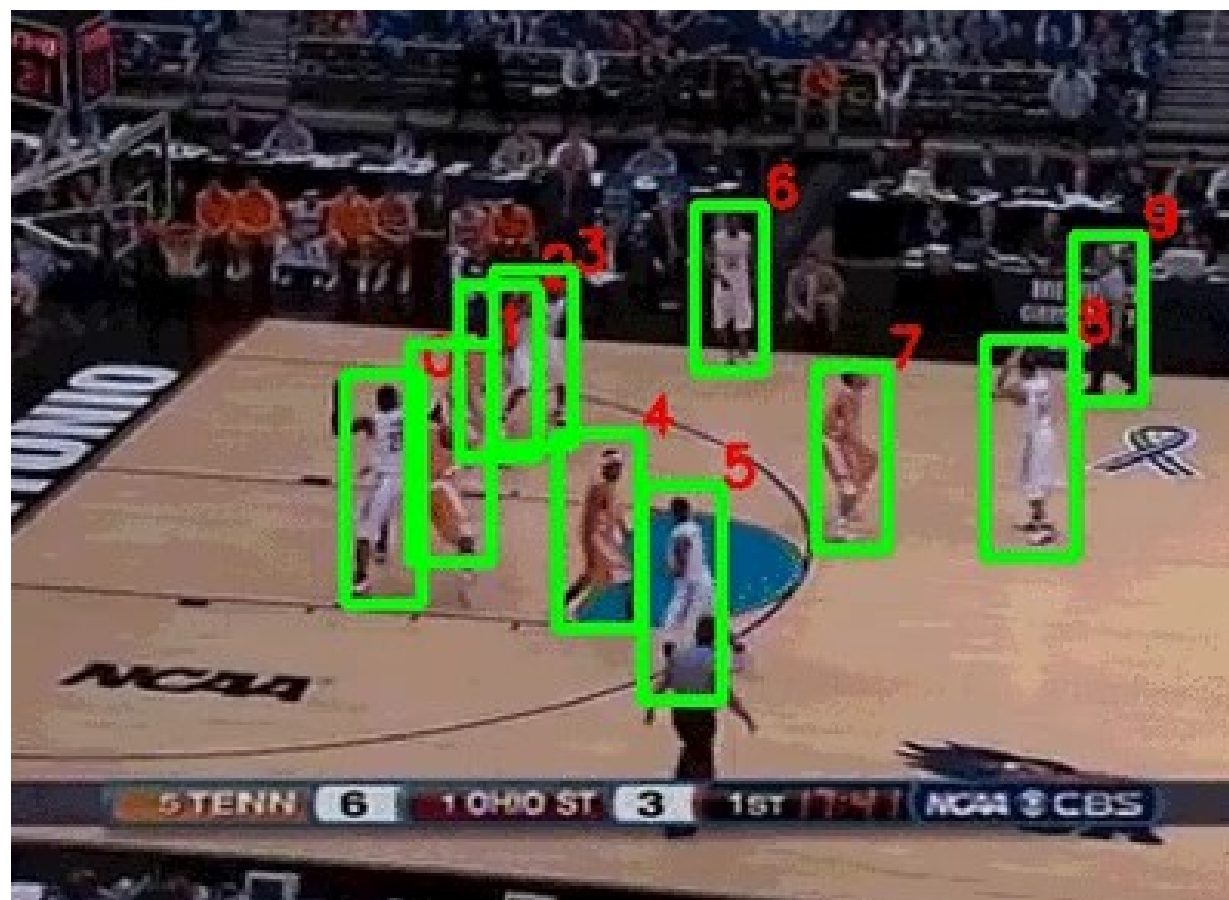
SRGAN  
(21.15dB/0.6868)



## 超分辨率重建



# 计算机视觉应用: 体育解说



## NBA赛事自动解说



# 计算机视觉应用：图像生成



AI试衣



“Photo of a cat riding on a ~~bicycle~~.  
car”

AI绘画



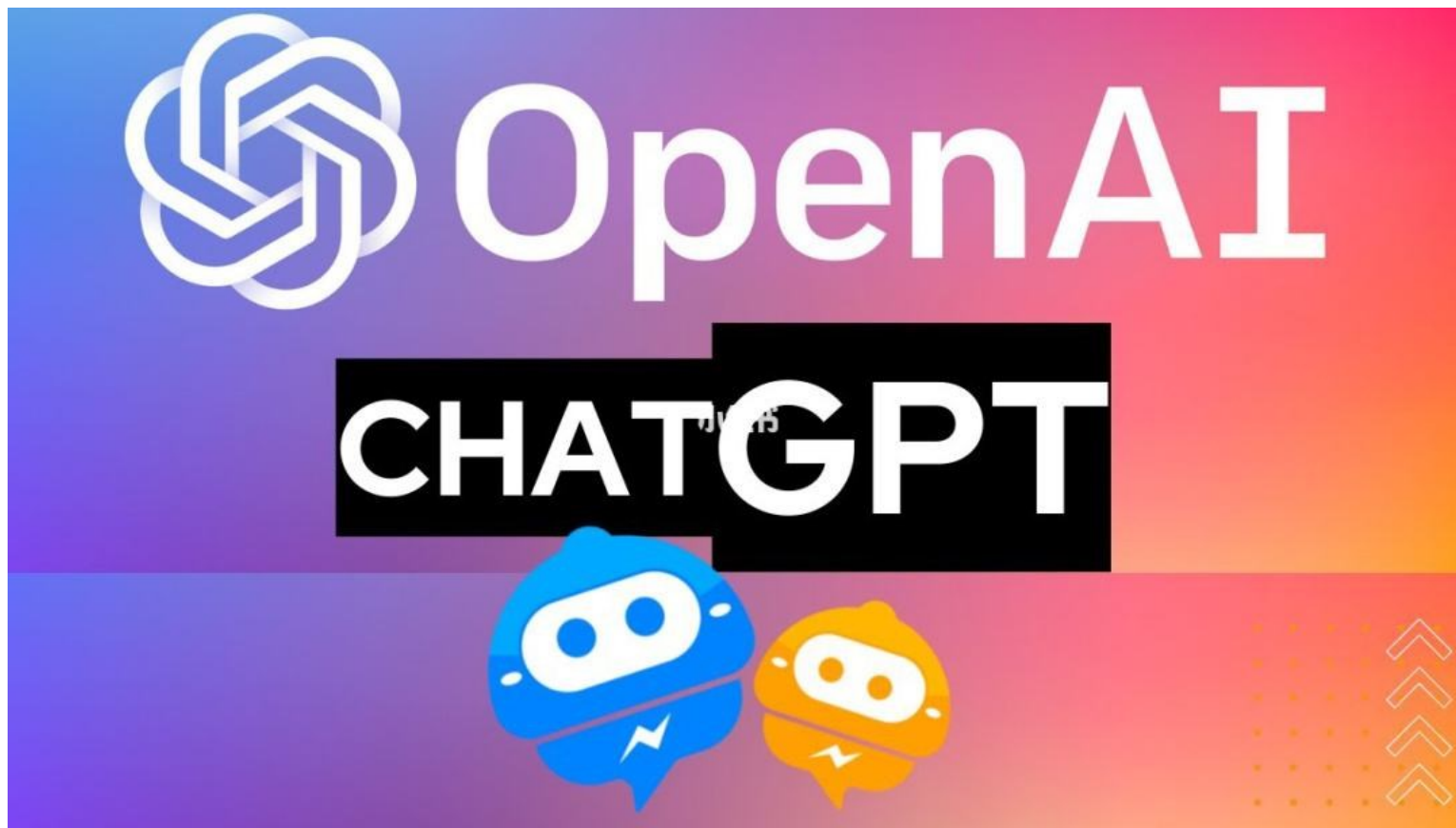
# 计算机视觉应用:动作识别和跟踪



动作实时检测和跟踪



# 智能聊天机器人



ChatGPT



# 智能聊天机器人

MI 你有什么能力?



作为百度公司开发的AI助理：文心一言，我具备以下能力：

1. 搜索引擎查询：借助百度搜索引擎，我可以为您提供各种问题的答案和解决方案。
2. 语言处理：我具备强大的自然语言处理能力，可以理解并回答您的问题。
3. 知识问答：我可以回答各种领域的常识性问题，包括科学、历史、文化、娱乐等。
4. 推荐服务：根据您的喜好和需求，我可以为您推荐优质的内容、产品和服务。
5. 生活助手：我可以帮助您管理日常生活事务，例如提醒您关键事件、查询天气、设置闹钟等。
6. 对话交流：我可以和您进行轻松的对话交流，陪伴您度过愉快的时光。

请随时告诉我您需要了解的信息或者需要帮助的事情，我将竭诚为您服务。



## 文心一言



# 计算机视觉应用：医学检测与分割任务

WU YING HUA  
2019/3/14 8:36:2  
TotalImage: 2

StudyId: 10002 PatientID: 190314087 PatientsName: WU YING HUA  
SeriesNumber: 4  
InstanceNumber: 87  
SliceThickness: 1.5  
SliceLocation: -303.5  
Kv: 120  
Ma: 432  
WW: 1500 / WL: -700

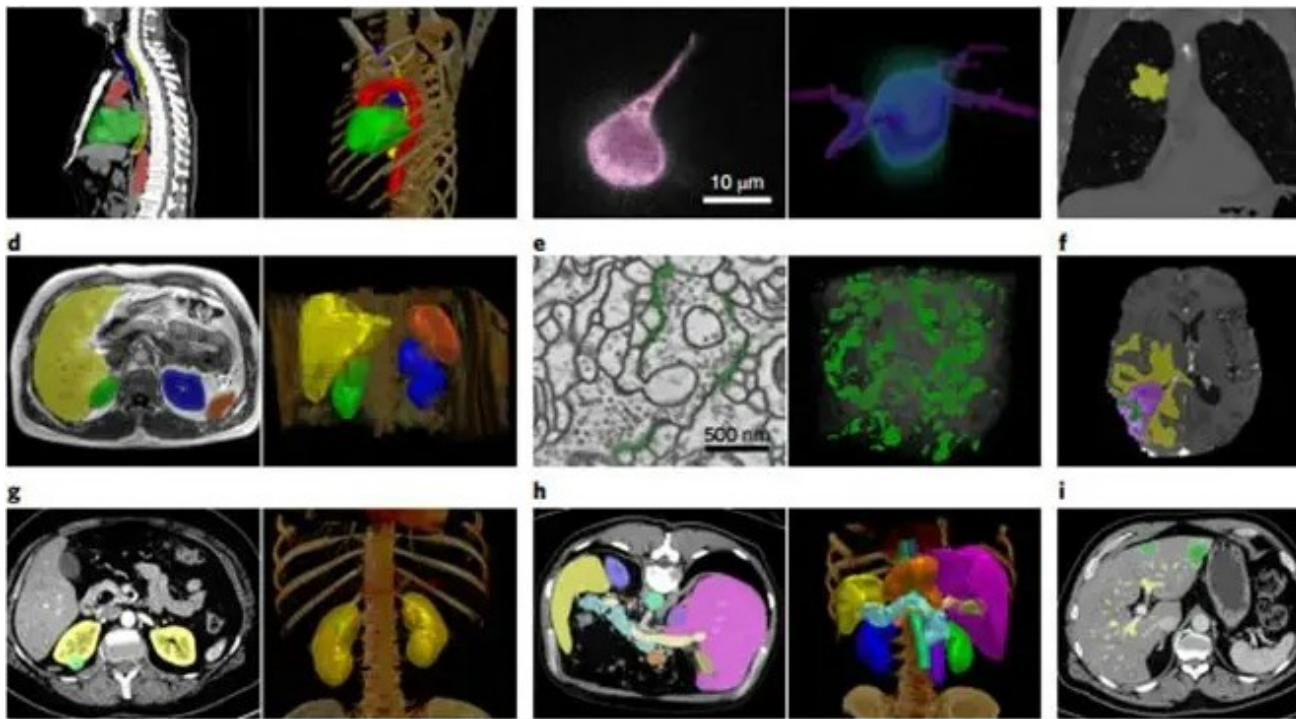
WU YING HUA  
2019/3/14 8:36:2  
TotalImage: 201

Current WAW: 1500 / WL: -700.00

No	IM	Size(mm)	Vol(mm <sup>3</sup> )	Density	PO
1	21	7.00	34.97	磨玻璃	-
2	87	17.00	752.74	部分实性	-

Nodules Report  
肺部影像筛查结果：  
经过本次筛查，共有 2 处可疑结节：  
1. 21层 -磨玻璃结节一枚  
2. 87层 -部分实性结节一枚

## 肺结节检测



## 多器官分割





# 机器学习/深度学习 (Machine Learning/Deep Learning)



# 学习资料推荐

## 经典书籍：

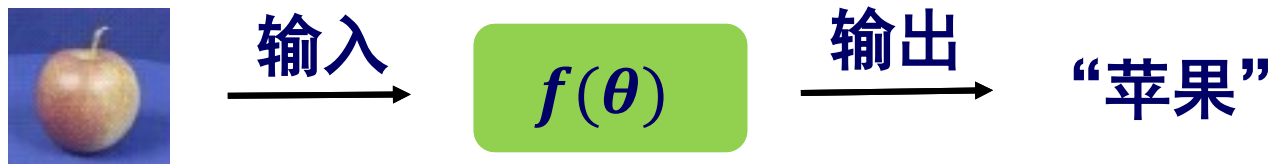
- ① 周志华 《机器学习理论导引》 机械工业出版社
- ① Richard O. Duda 《模式分类》 机械工业出版社
- ① Ian Goodfellow 《深度学习》 人民邮电出版社

## 在线资源：

- ① 斯坦福大学计算机视觉课程cs231n  
<http://cs231n.stanford.edu/schedule.html>
- ① 李宏毅免费机器学习课程  
<https://aistudio.baidu.com/aistudio/1hycourse>
- ① 深度学习CV从入门到实战  
<https://aistudio.baidu.com/aistudio/course/introduce/789>



# 什么是机器学习

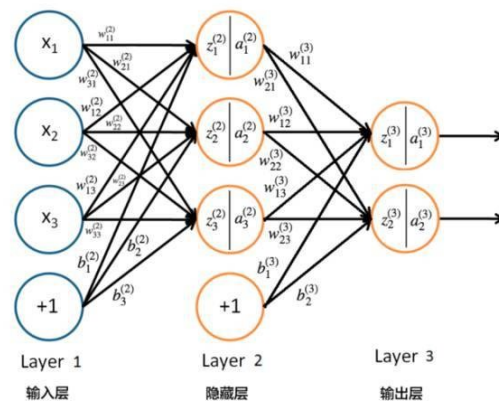


机器学习的目的：让计算机从一系列函数中找出一个最优函数，用于实现特定的任务，比如分类和回归。

这个函数一般称为模型(Model)，由架构（线性、网络）和参数构成。

$$y = wx + b$$

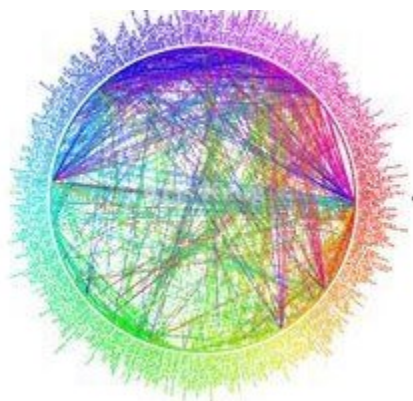
简单的线性模型



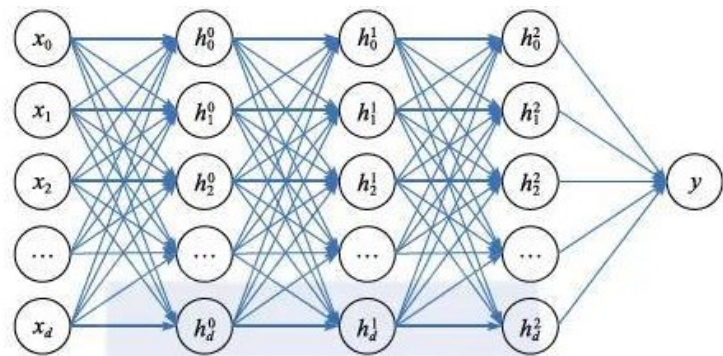
复杂的神经网络模型



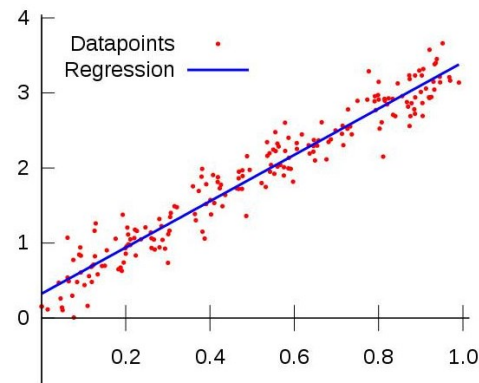
# 基本任务：回归和分类



数据/特征

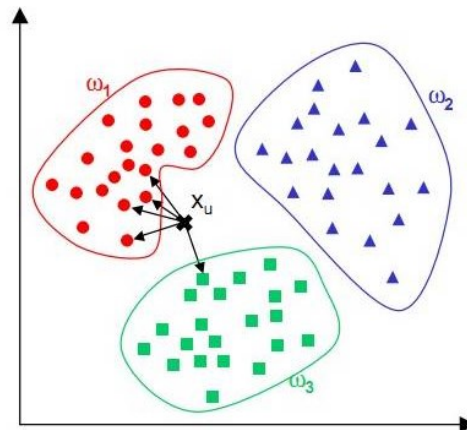


机器学习模型  
(SVM, RF, ANN, CNN)



回归

输出连续的结果  
(房价)



分类

输出离散的结果  
(类别)

图像模型主要是分类任务，分割、检测本质上也是分类的应用



# 有监督和无监督学习

有监督  
supervised



Training



Model

$f(\text{apple}) = \text{"apple"}$   $f(\text{tomato}) = \text{"tomato"}$   $f(\text{cow}) = \text{"cow"}$

预测 (Predictions)



标签 (Labels)

无监督  
unsupervised



Training



Model

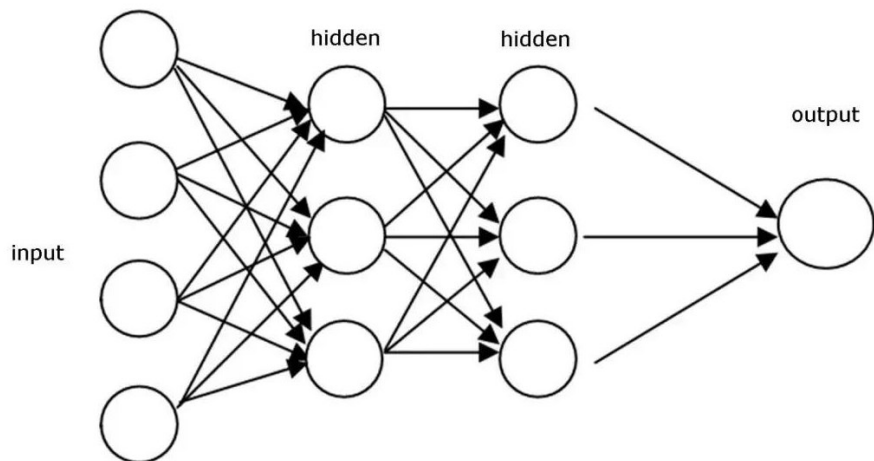
$f(\text{apple}) = \text{"apple"}$   $f(\text{tomato}) = \text{"tomato"}$   $f(\text{cow}) = \text{"cow"}$

预测 (Predictions)

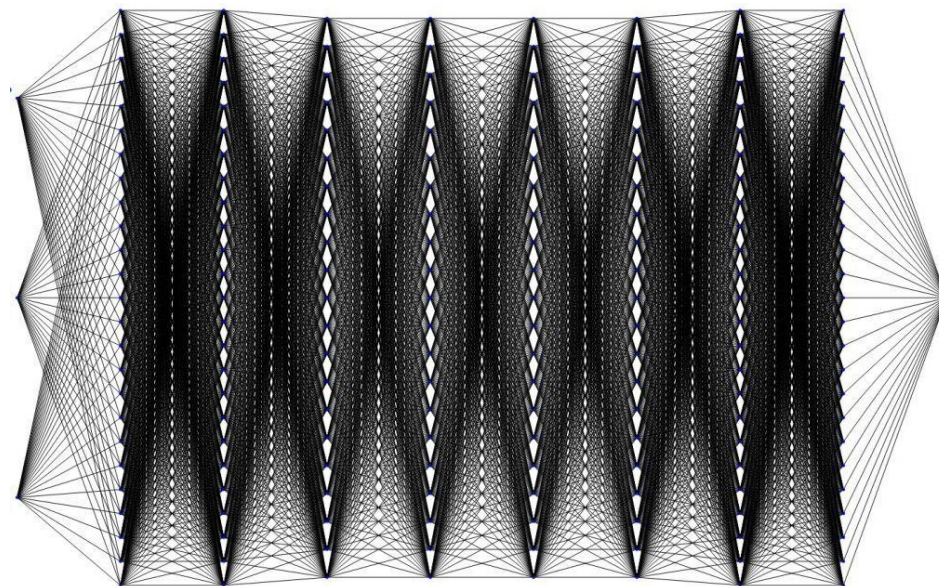
- 有监督学习：数据集需要有标签信息。主流。
- 无监督学习：数据集不需要标签信息。难，应用场景简单。



# 深度学习



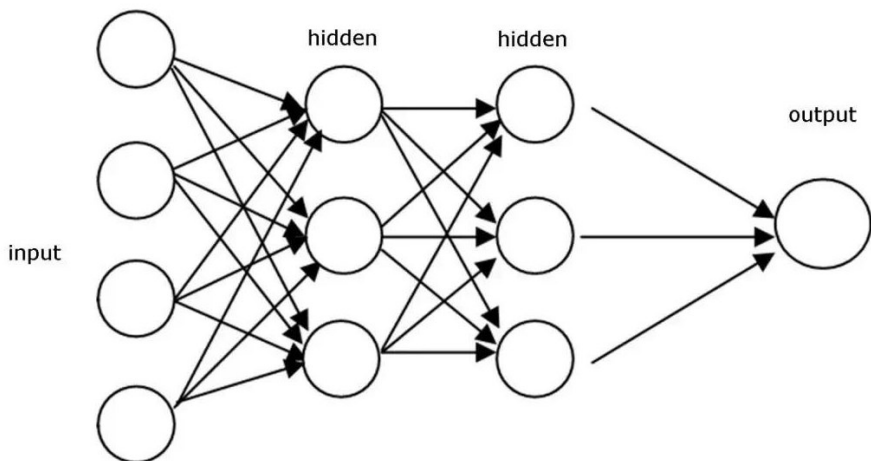
**机器学习**  
网络比较浅，神经元比较少



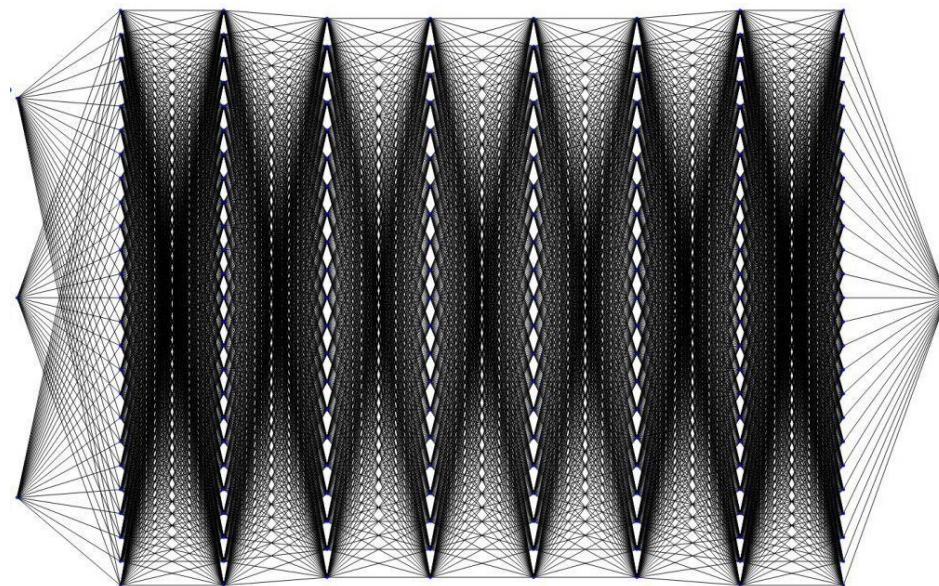
**深度学习**  
网络更深，神经元非常多



# 深度学习



**机器学习**  
网络比较浅，神经元比较少



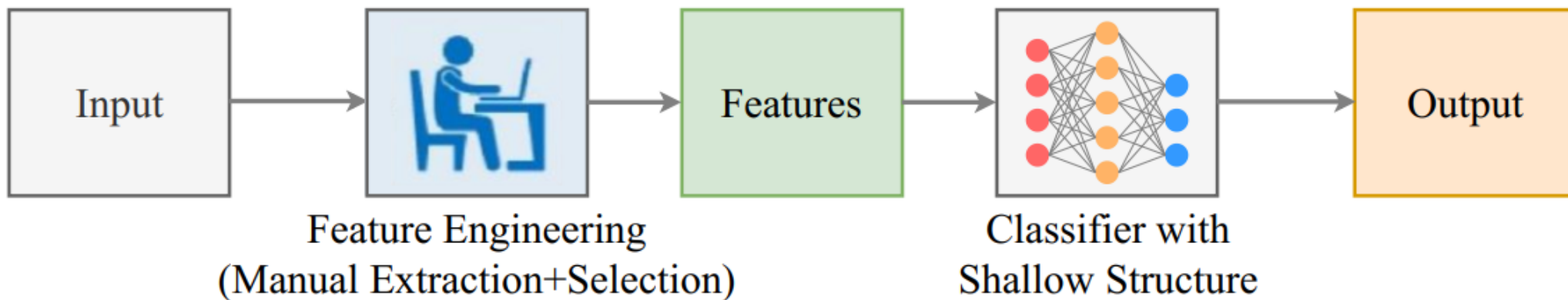
**深度学习**  
网络更深，神经元非常多

- 存储、GPU并行加速等硬件的发展
- 梯度消失问题的有效解决

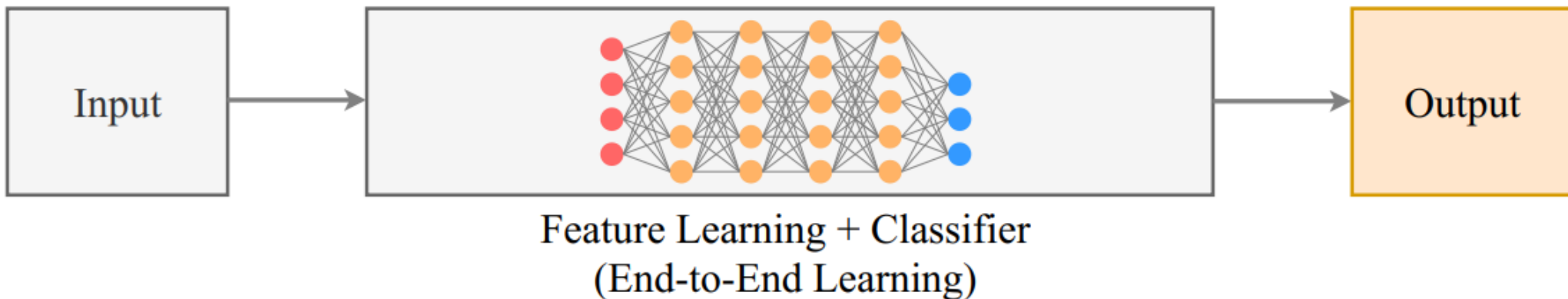


# 手工特征 vs. 深度特征

## 传统视觉任务框架：



## 深度学习视觉任务框架：







## 如何去衡量模型的性能 ?

**Analyzing**





# 模型泛化性能

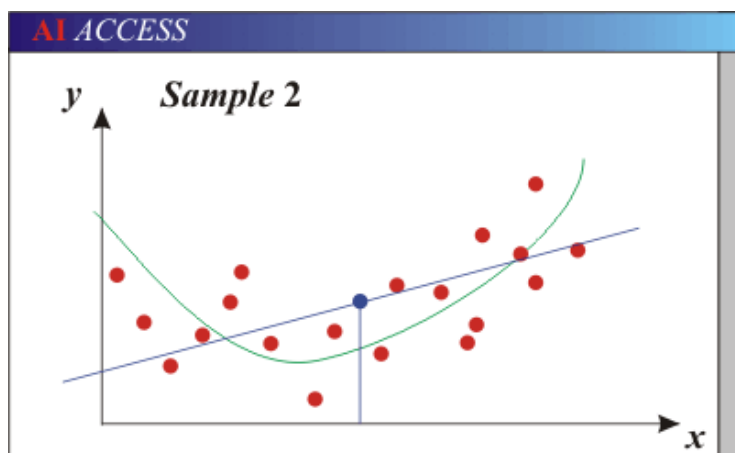
用训练集训练好的模型在未知数据上的预测好坏（性能表现高低），通常叫做泛化能力。

$$E(\text{泛化误差}) = \text{noise (噪声)} + \text{bias (偏差)} + \text{variance (方差)}$$

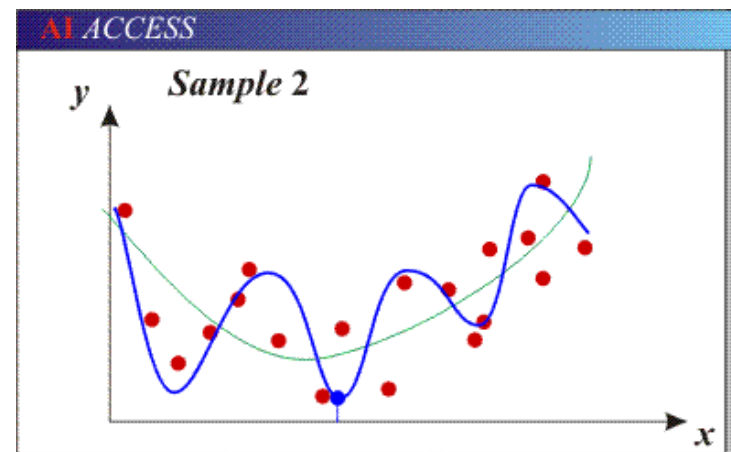
刻画了学习问题本身的难度

刻画预测和真实的偏离程度

刻画数据扰动对性能影响



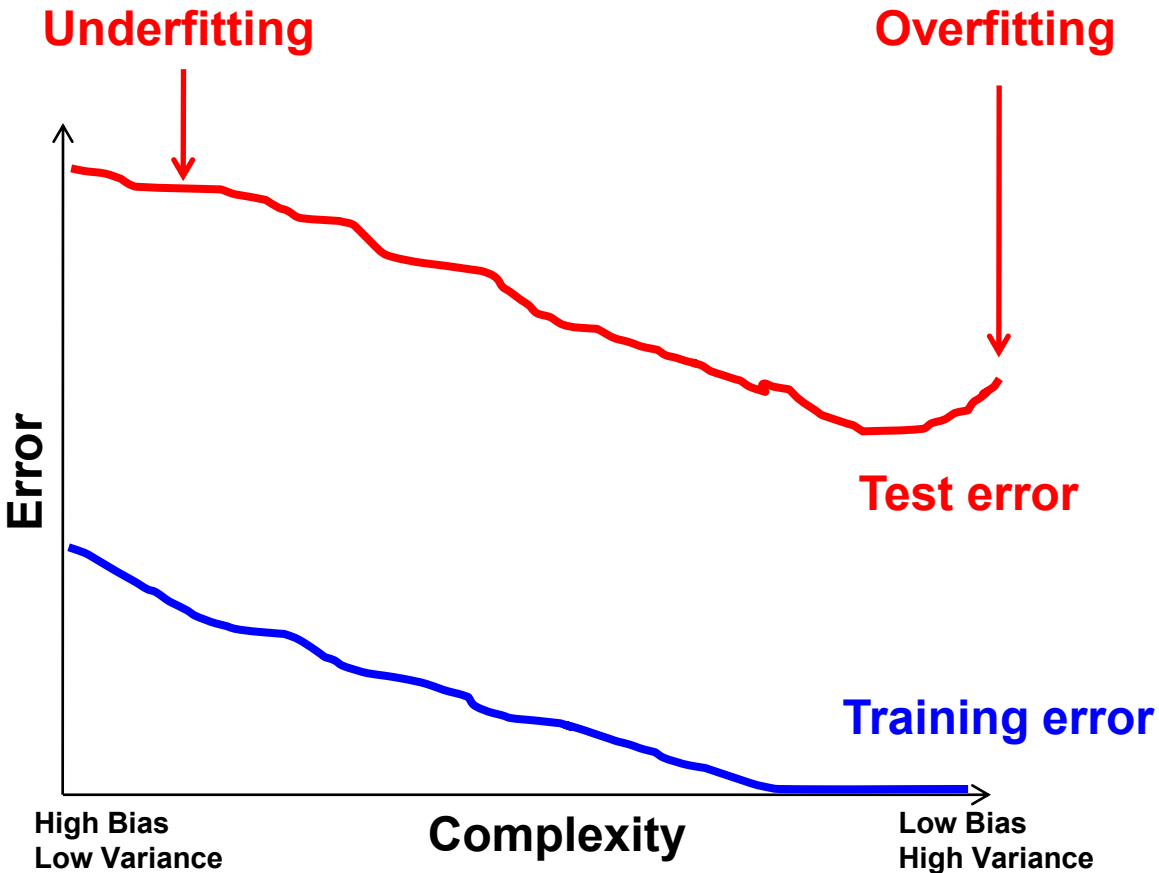
**欠拟合**：模型假设过于简单，参数量少，无法充分拟合真实期望模型  
(高偏差：模型连训练数据都不能很好识别)



**过拟合**：模型假设过于复杂，参数量大，过度拟合数据  
(高方差：模型只能识别训练数据，扰动影响很大)

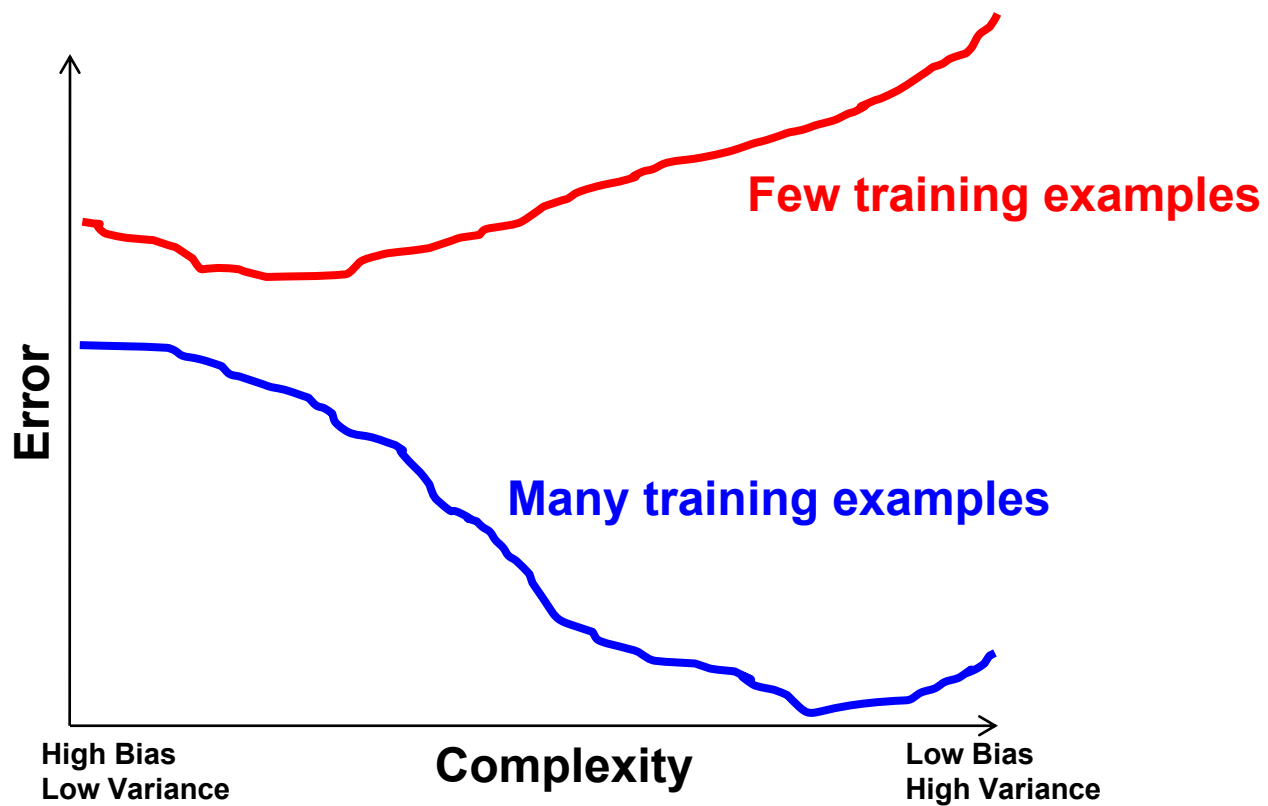


# 欠拟合和过拟合





# 数据量的影响



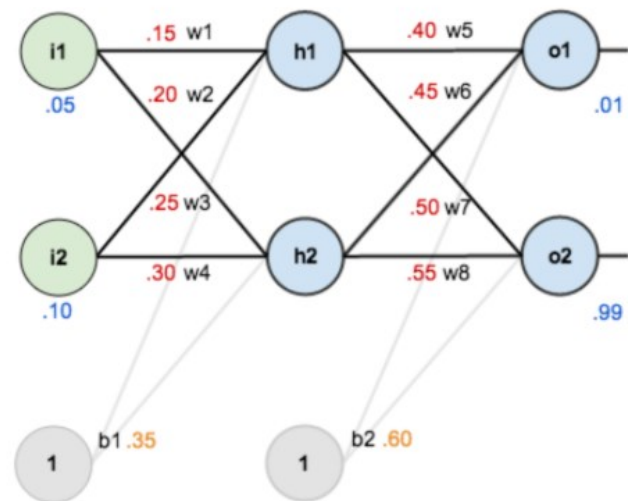


# 缓解过拟合常用方法

- 尽量获得更多的训练数据
- 数据扩增手段
- 简化模型结构和减少参数量
- Dropout: 随机丢弃部分参数 (设为0)
- 正则化: Loss函数加入L1或者L2

L2正则化: loss函数后面加上权重参数的平方  
效果: 权重参数值趋向于0, 也称为权重衰减。

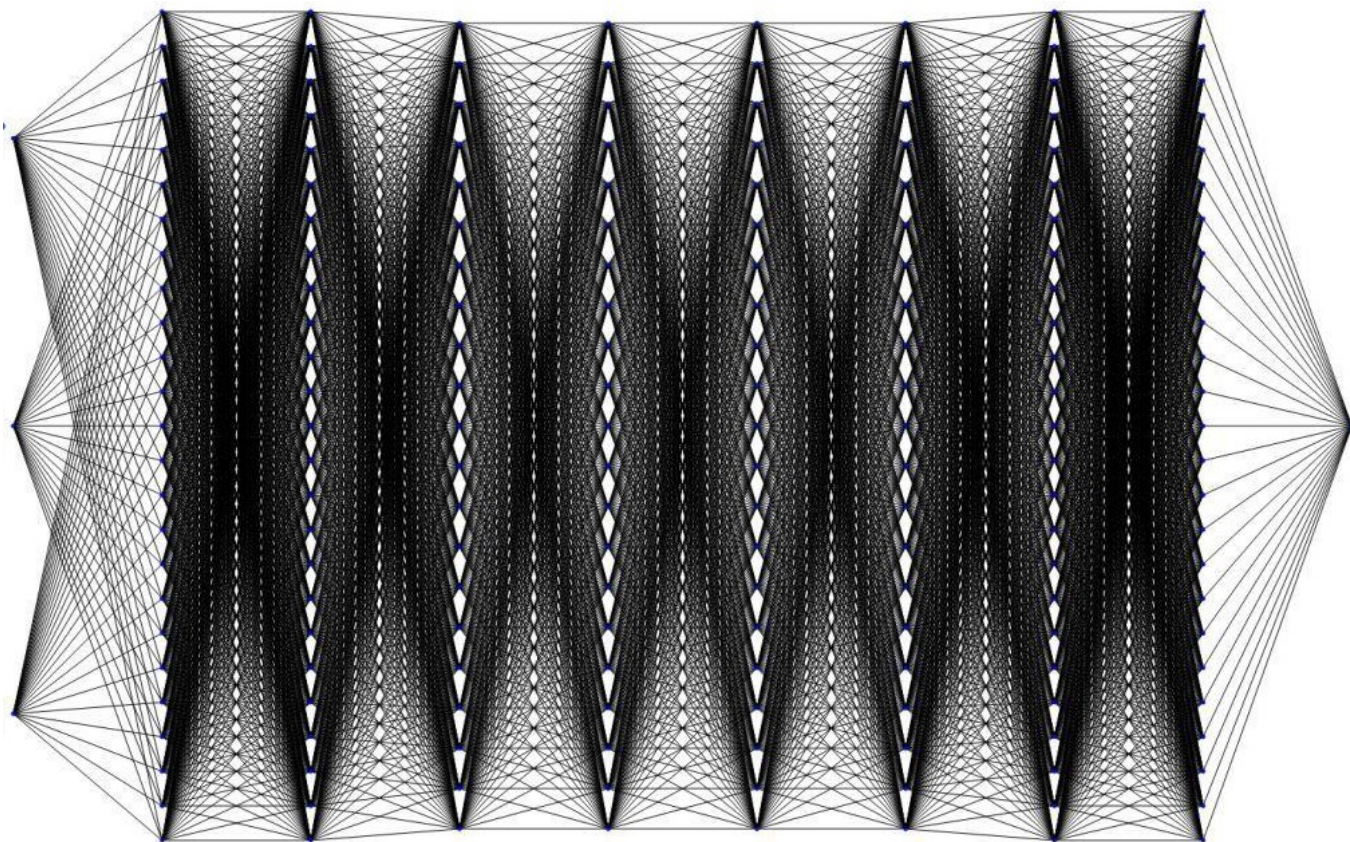
$$L + \lambda \sum w^2$$





# 反向传播

反向传播不是新的优化算法，是实现梯度下降法的策略。



深度神经网络由上百万甚至上千万级别参数：

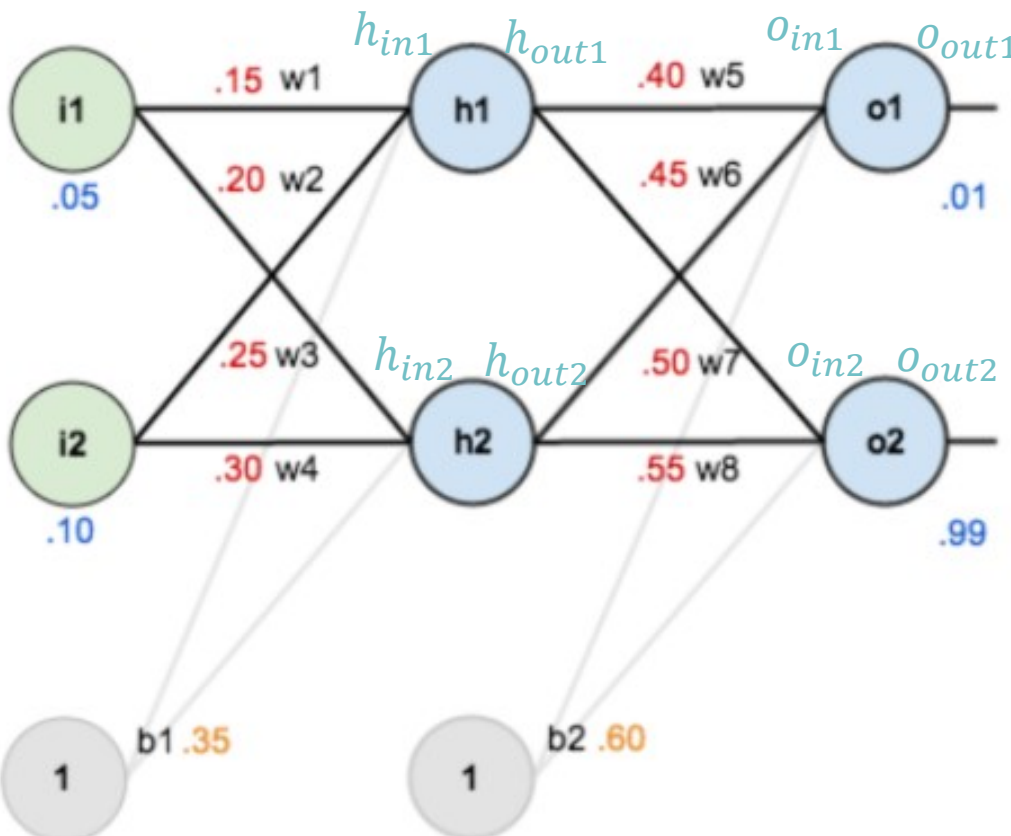
$$\theta = \{w_1, w_2, \dots, b_1, b_2, \dots\}$$

为了提高计算效率，采用反向传播算法

李宏毅：<https://aistudio.baidu.com/aistudio/education/lessonvideo/869145>



# 训练过程：前向传播和反向传播



输入:  $i_1=0.05, i_2=0.10$ ;

目标:  $\hat{o}_{out1}=0.01, \hat{o}_{out2}=0.99$ ;

====Step1 初始化参数:

$w_1 = 0.15, w_2 = 0.20, w_3 = 0.25, w_4 = 0.30,$   
 $w_5 = 0.40, w_6 = 0.45, w_7 = 0.50, w_8 = 0.55,$   
 $b_1 = 0.35, b_2 = 0.60$

====Step2 前向传播:

输入层->隐藏层:

$$h_{out1} = \delta(h_{in1}) = \delta(0.15 * 0.05 + 0.20 * 0.10 + 0.35)$$

$$= \delta(0.3775) = 0.5933,$$

同理可算:  $h_{out2} = 0.5969$

隐藏层->输出层:

$$o_{out1} = \delta(o_{in1}) = \delta(0.4 * 0.5933 + 0.45 * 0.5969 + 0.60)$$

$$= \delta(1.1059) = 0.7514,$$

同理可算:  $o_{out2} = 0.7729$

**结果跟目标差距较大!**

Sigmoid:  $\delta(z) = \frac{1}{1+e^{-z}}$



# 训练过程：前向传播和反向传播

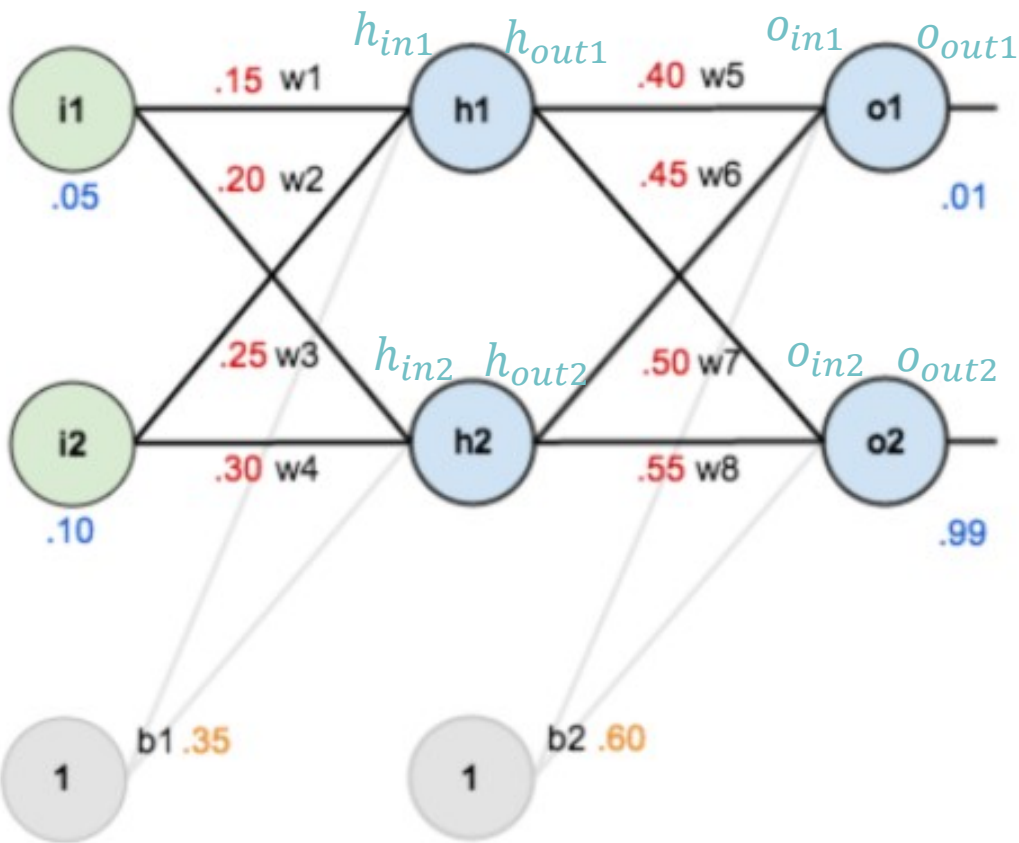
损失函数:  $L(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$

导数:  $\dot{L}(\hat{y}, y) = -(\hat{y} - y)$

输入:  $i_1=0.05, i_2=0.10$ ;

目标:  $\hat{o}_{out1}=0.01, \hat{o}_{out2}=0.99$ ;

输出:  $o_{out1}=0.7514, o_{out2}=0.7729$



====Step3 反向传播:

计算损失:

$$L = L_{o_{out1}} + L_{o_{out2}} = \frac{1}{2}(0.01 - 0.7514)^2 + \frac{1}{2}(0.99 - 0.7729)^2 = 0.2984$$

计算  $\frac{\partial L}{\partial w_5}$ :

$$\frac{\partial L}{\partial w_5} = \frac{\partial L}{\partial o_{out1}} * \frac{\partial o_{out1}}{\partial o_{in1}} * \frac{\partial o_{in1}}{\partial w_5}$$

链式法则

损失函数导数:  $\frac{\partial L}{\partial o_{out1}} = -(\hat{o}_{out1} - o_{out1}) = 0.7414$

激活函数导数:  $\frac{\partial o_{out1}}{\partial o_{in1}} = o_{out1}(1 - o_{out1}) = 0.1868$

线性函数导数:  $\frac{\partial o_{in1}}{\partial w_5} = h_{out1} = 0.5933$

$$\frac{\partial L}{\partial w_5} = 0.7414 * 0.1868 * 0.5933 = 0.082$$

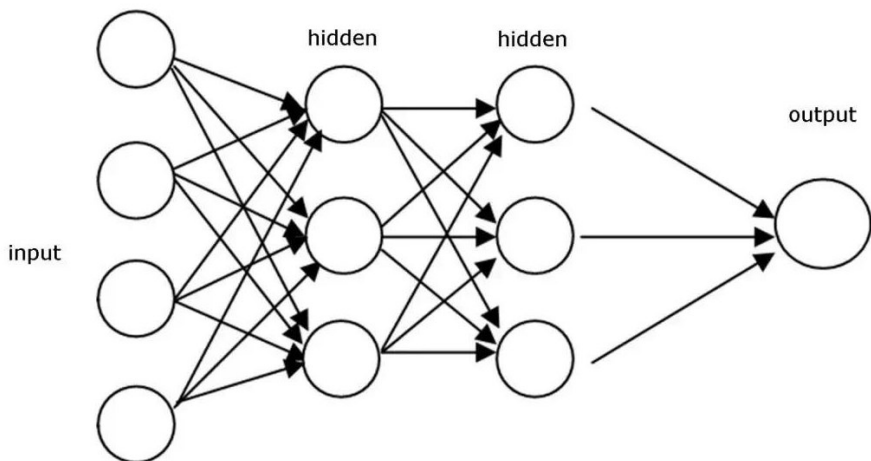
Sigmoid:  $\delta(z) = \frac{1}{1+e^{-z}}$

导数:  $\dot{\delta}(z) = \delta(z)[1 - \delta(z)]$

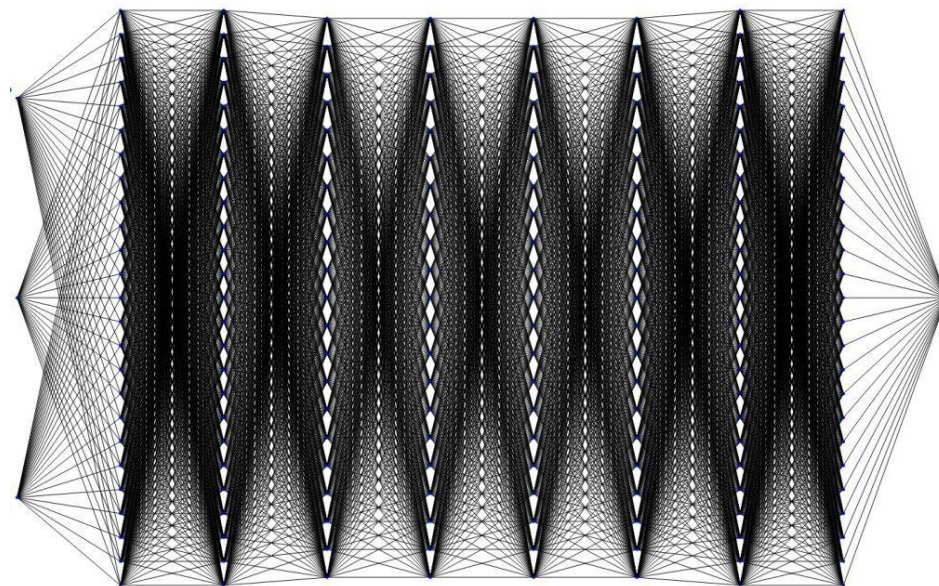




# 深度学习



**机器学习**  
网络比较浅，神经元比较少



**深度学习**  
网络更深，神经元非常多

- 存储、GPU并行加速等硬件的发展
- 梯度消失问题的有效解决

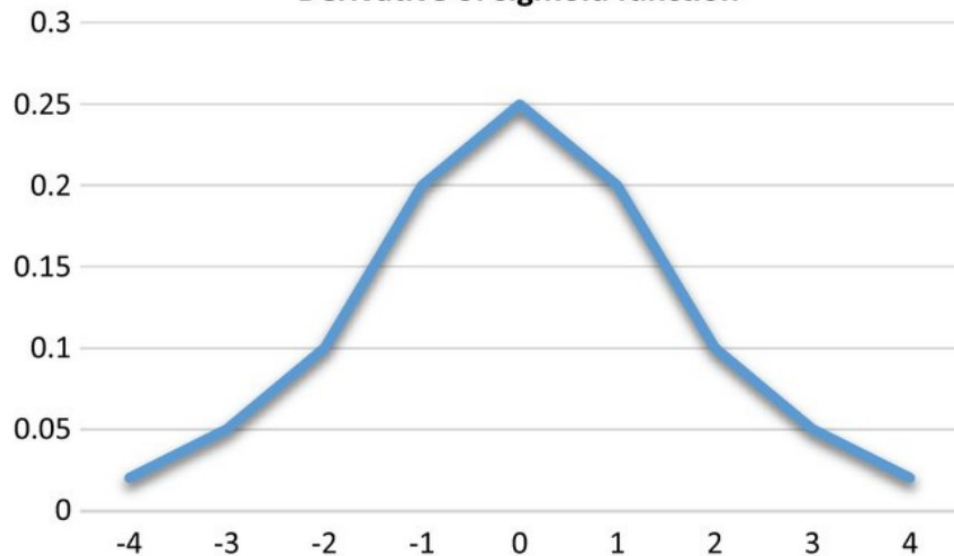


# 梯度消失问题

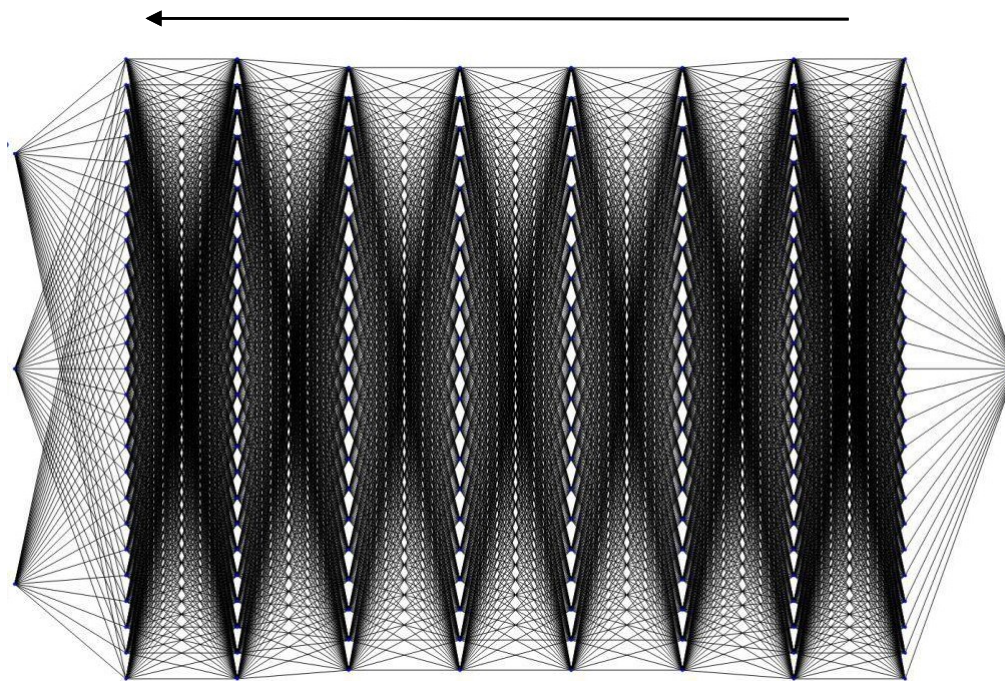
$$\frac{\partial L}{\partial w_5} = \frac{\partial L}{\partial o_{out1}} * \frac{\partial o_{out1}}{\partial o_{in1}} * \frac{\partial o_{in1}}{\partial w_5}$$

$$0.5 * 0.4 * 0.9 = 0.18 < 0.5 * 0.9 = 0.45$$

Derivative of sigmoid function



导数<1，自变量绝对值越大，导数越小



反向传播过程中，越浅的网络参数，其梯度的乘积项越多（链式法则）。

解决办法：其它激活函数，ReLU、LeakReLU等



# 计算机视觉深度网络 (Deep Network in Computer Vision)

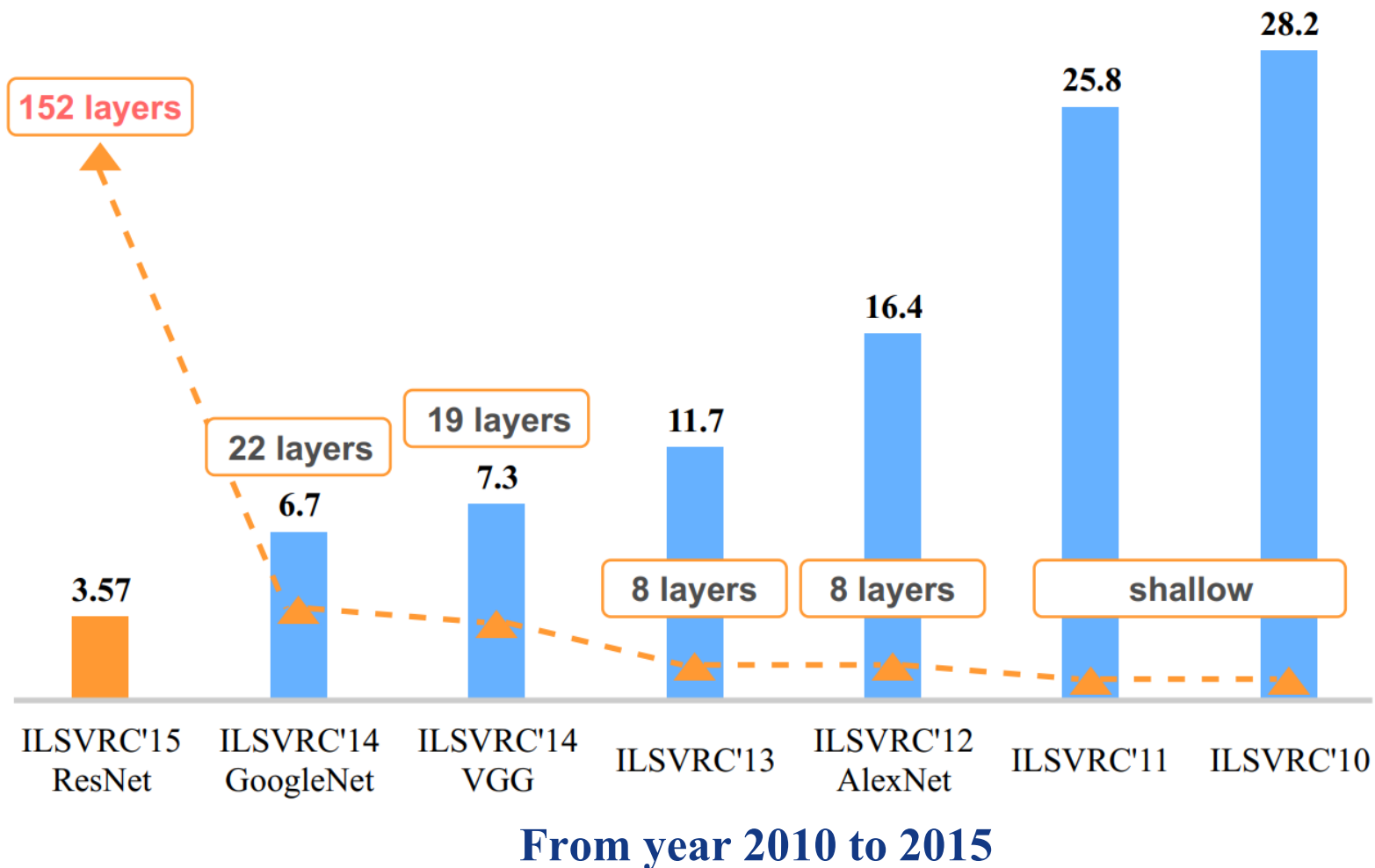


计算机视觉基础贡献奖 - CVPR 2019

[Deng, CVPR, 2009]

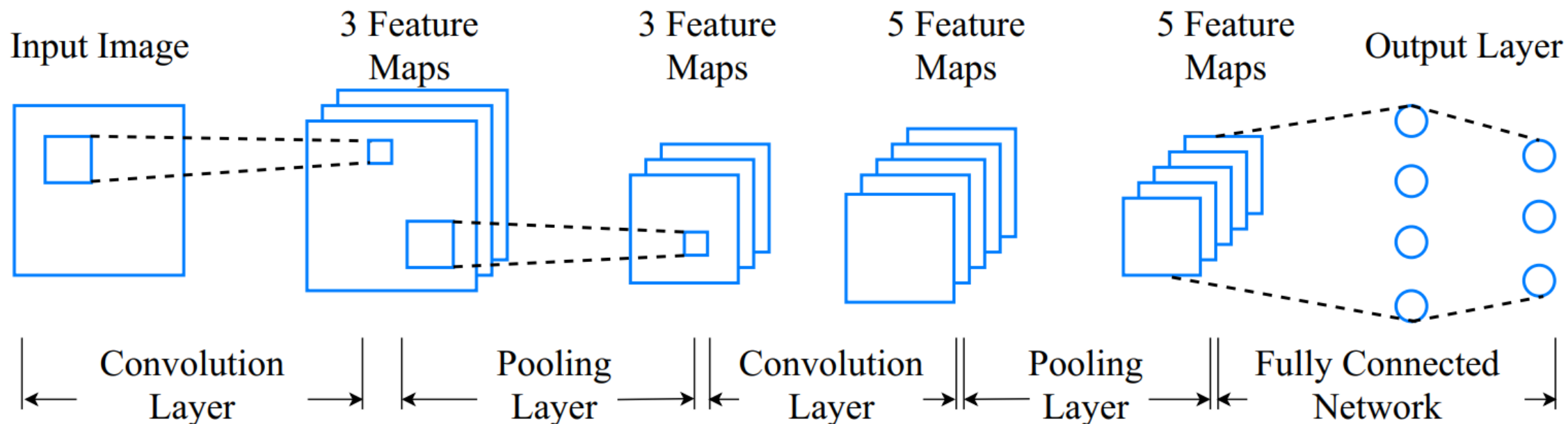


# ImageNet分类Top-5错误率





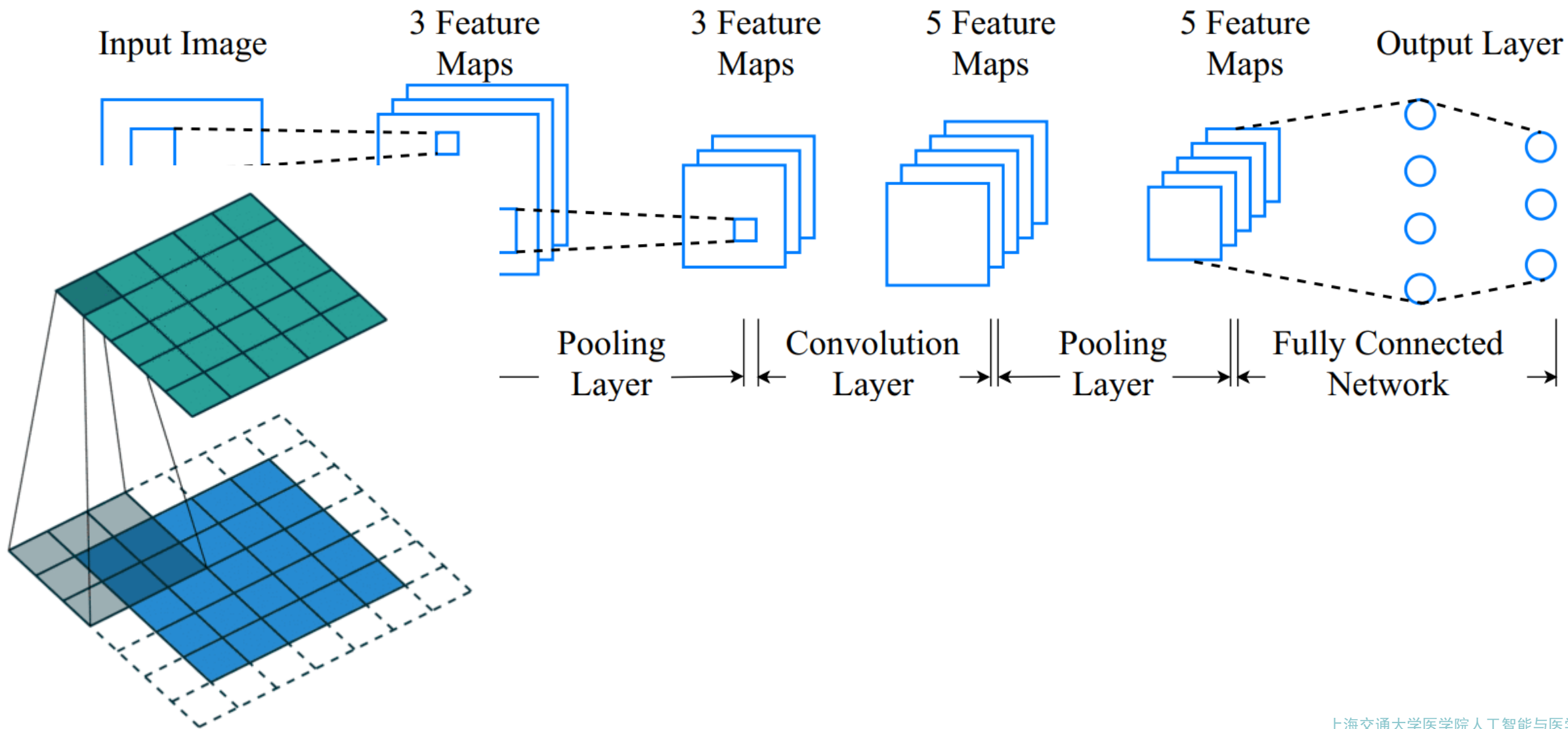
# CNN网络的主要结构



**CNN: 善于提取图像特征**

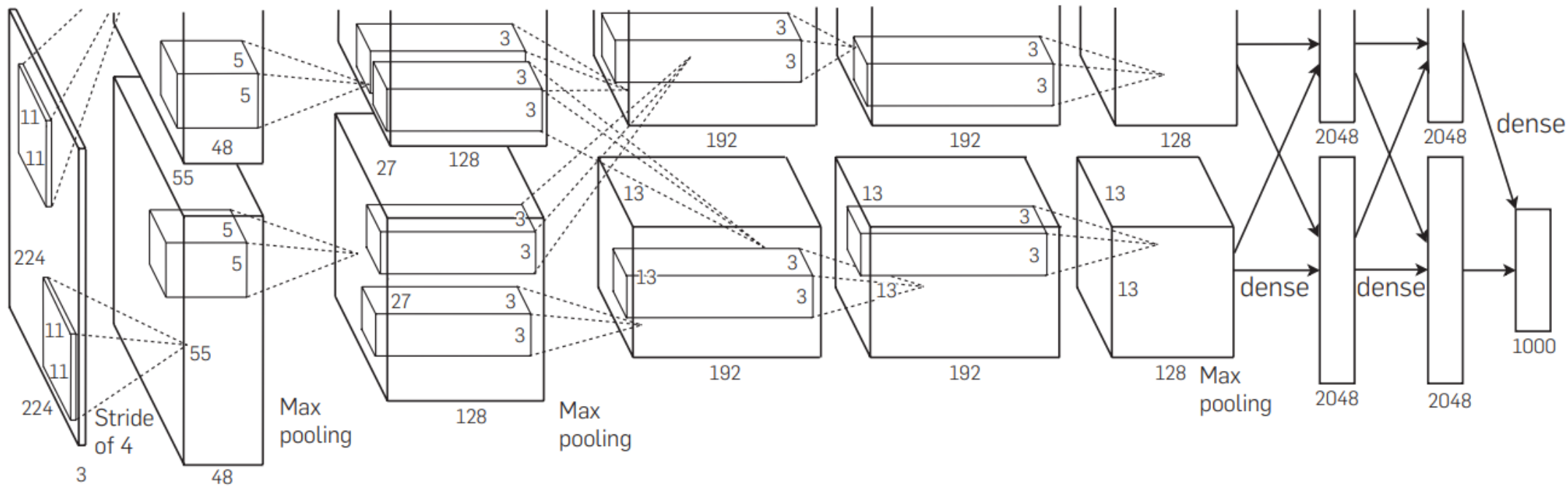


# CNN网络的主要结构





# 经典卷积神经网络-AlexNet



[Krizhevsky, Communications of the ACM, 2017]

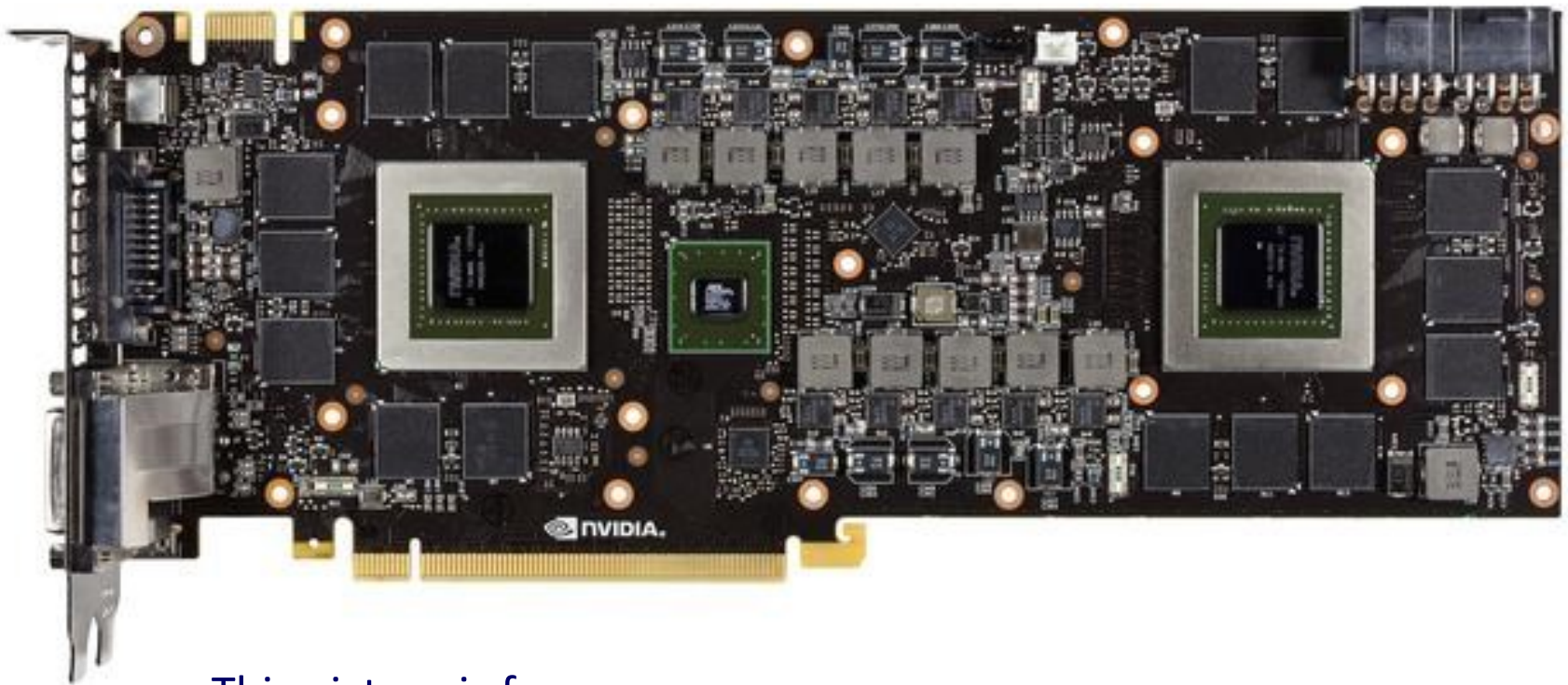
2012年ImageNet竞赛冠军

使用GPU进行运算加速





# GeForce GTX 690 - year 2012



This picture is from:

<https://baijiahao.baidu.com/s?id=1757972284429992661&wfr=spider&for=pc>

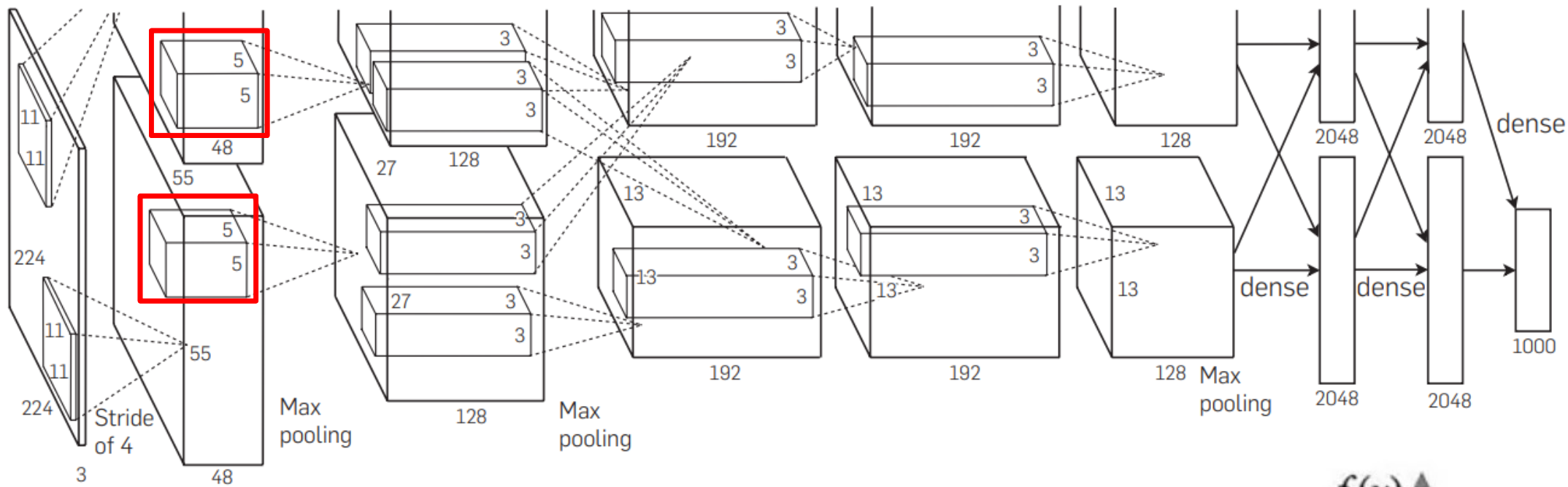
28nm制程工艺 & Kepler架构 (\$1,000)

显存容量: 4096MB

显存频率: 6000MHz



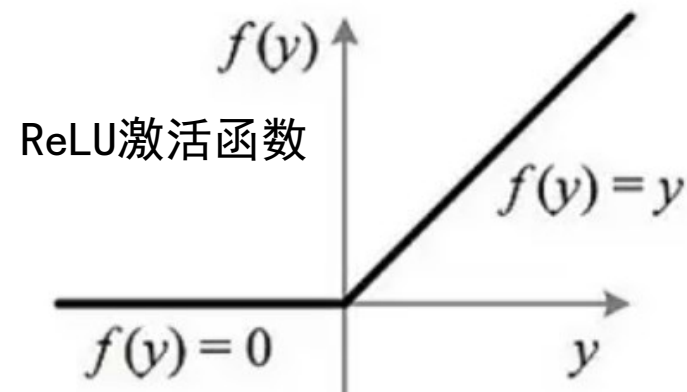
# 经典卷积神经网络-AlexNet



[Krizhevsky, Communications of the ACM, 2017]

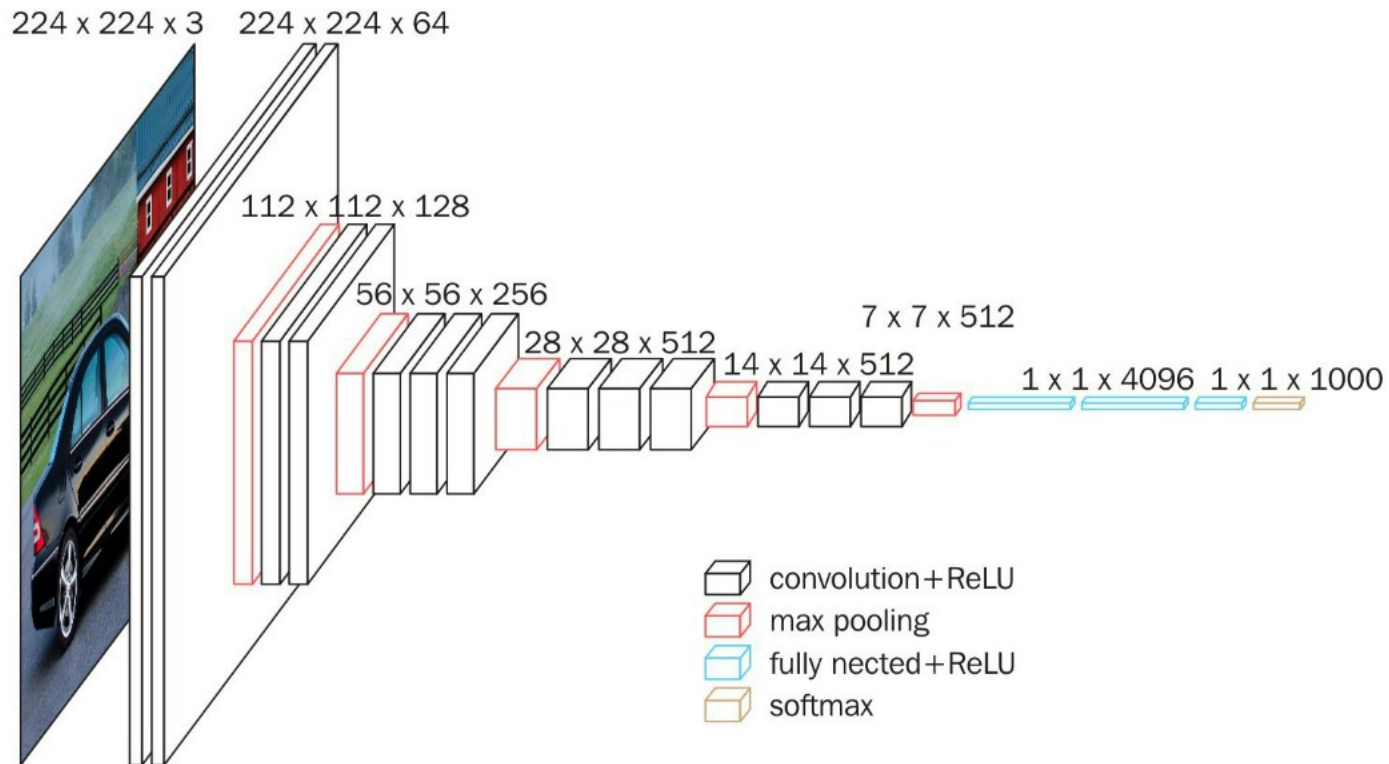
2012年 ImageNet 竞赛冠军

使用GPU进行运算加速





# 经典卷积神经网络-VGG



[Simonyan, ICLR, 2015]

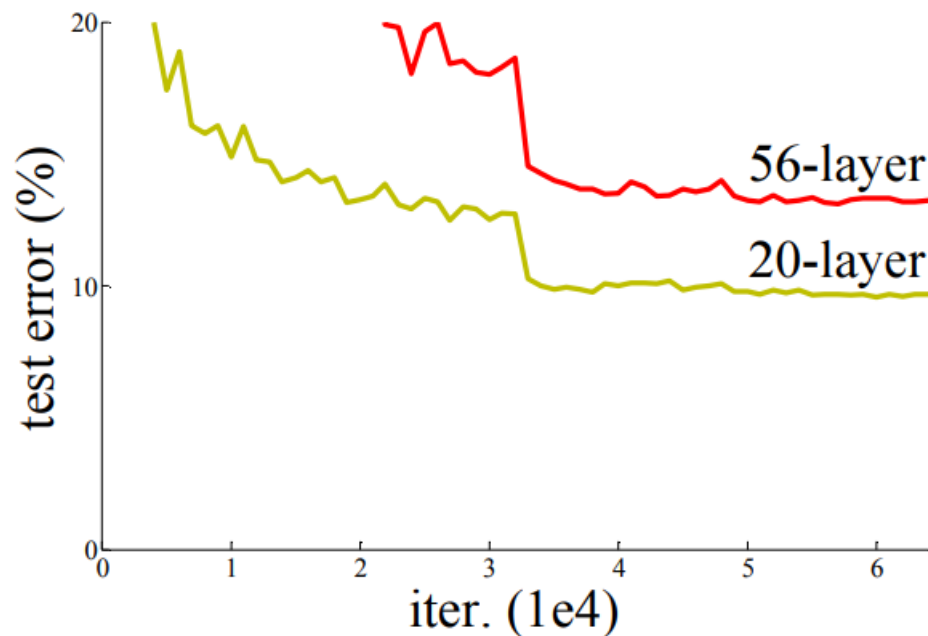
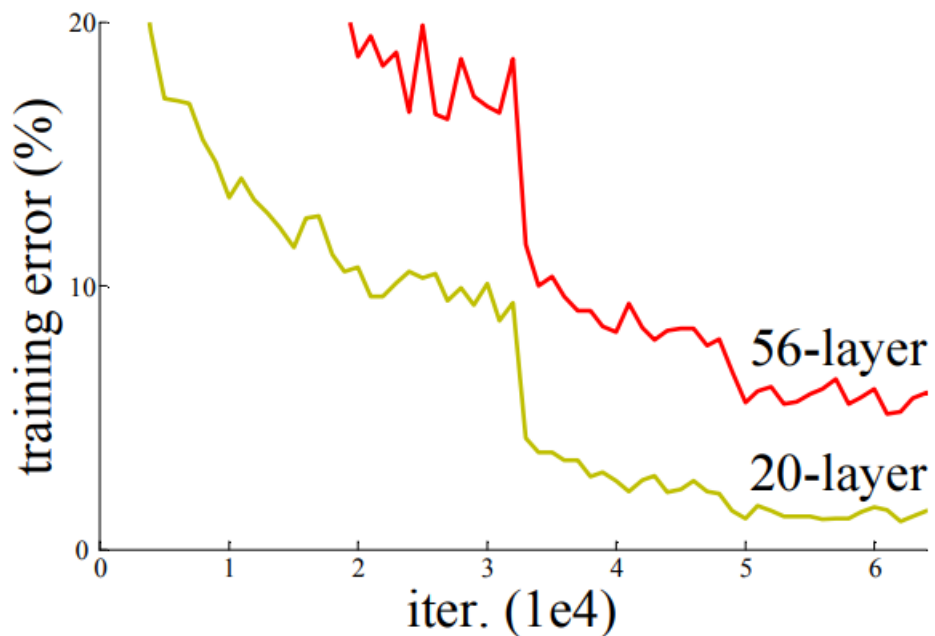
2014年ImageNet竞赛 (Localization Task第一名 & Classification Task 第二名)

小卷积核&多尺度 ❌

加深网络深度 ✅



# 经典CNN模型-ResNet



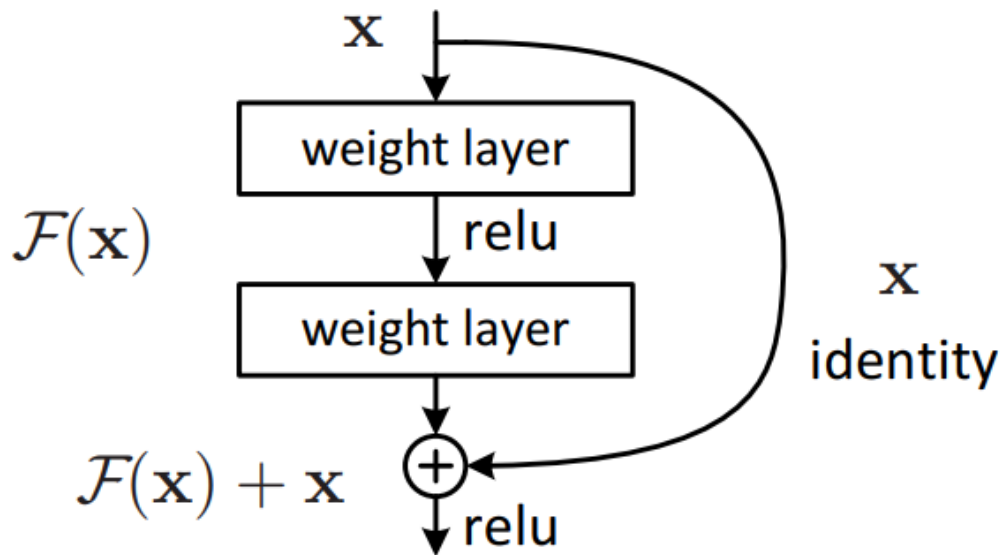
**Training / Test error on CIFAR-10 with 20-layer and 56-layer “plain” networks**

This picture is from:

[https://openaccess.thecvf.com/content\\_cvpr\\_2016/papers/He\\_Deep\\_Residual\\_Learning\\_CVPR\\_2016\\_paper.pdf](https://openaccess.thecvf.com/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf)

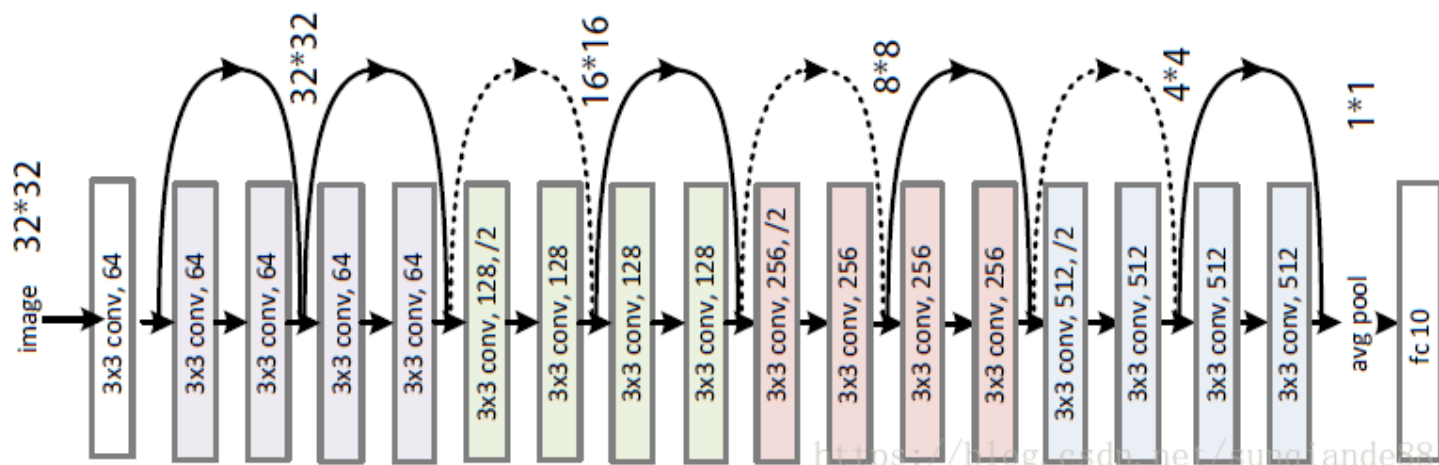


# 经典CNN模型-ResNet



残差学习单元

可以有效缓解  
梯度消失问题

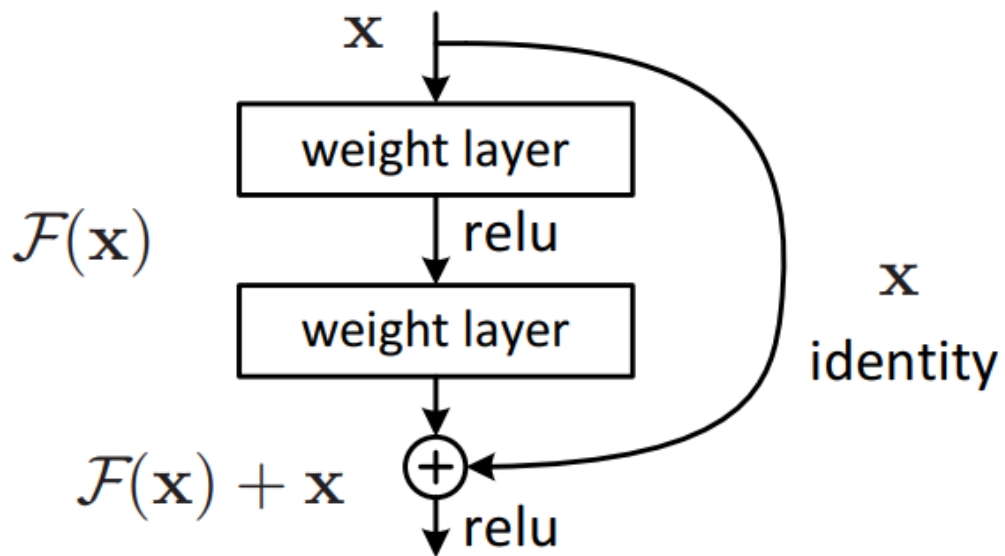


ResNet-18

[He, CVPR, 2016]

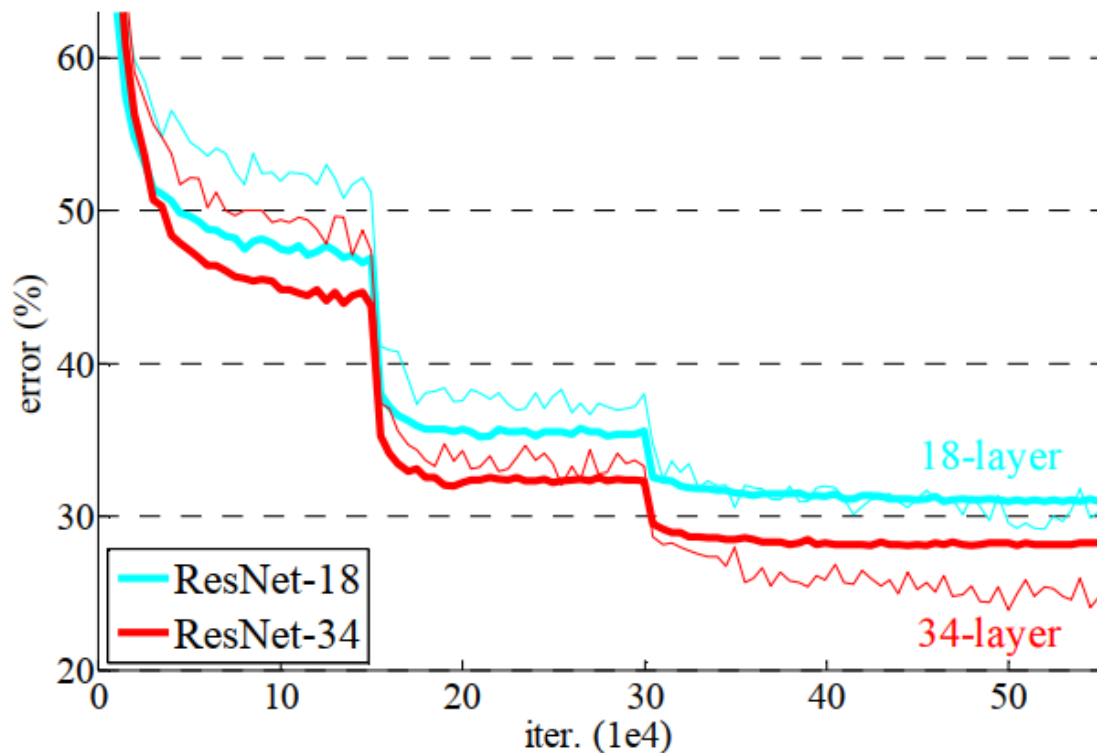


# 经典CNN模型-ResNet



残差学习单元

可以有效缓解  
梯度消失问题



Validation error of ResNet-18 and ResNet-34



# 经典CNN模型-ResNet

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$

## ResNet 系列



# 经典CNN模型-DenseNet

- 特征重用（性能好）
- 参数少
- 计算量小
- 内存消耗大上采样

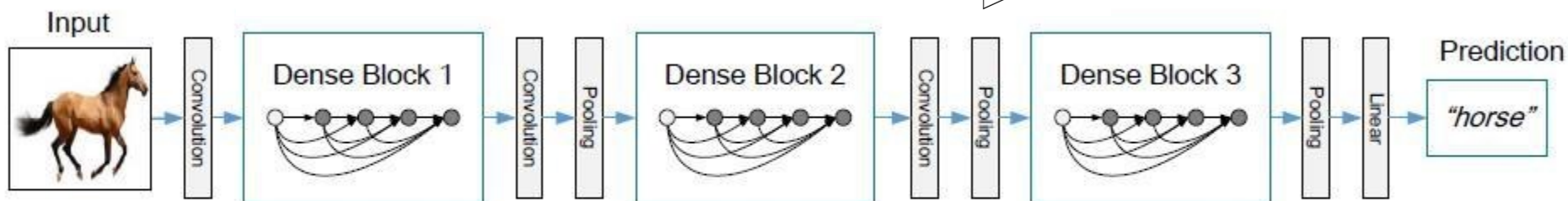
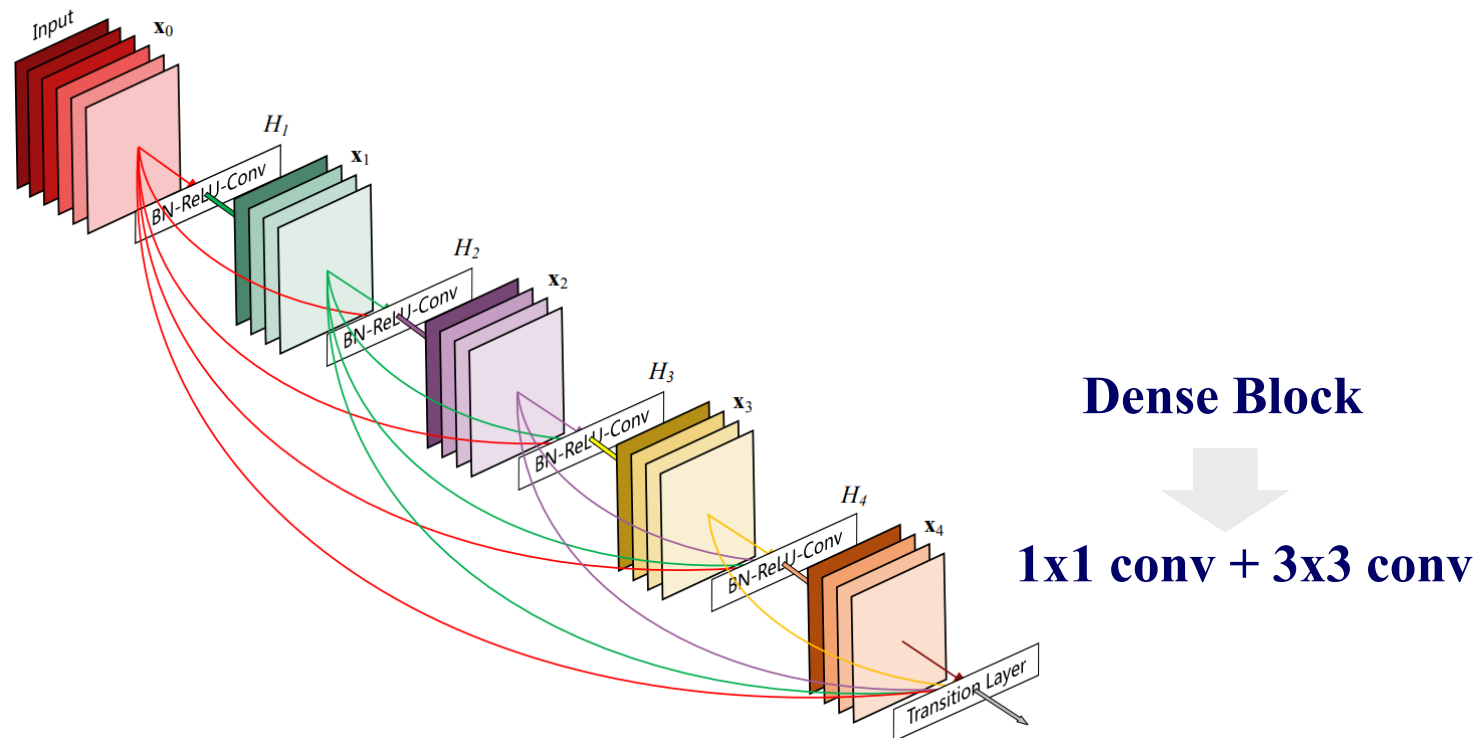


Figure 2. A deep DenseNet with three dense blocks. The layers between two adjacent blocks are referred to as transition layers and change feature map sizes via convolution and pooling.

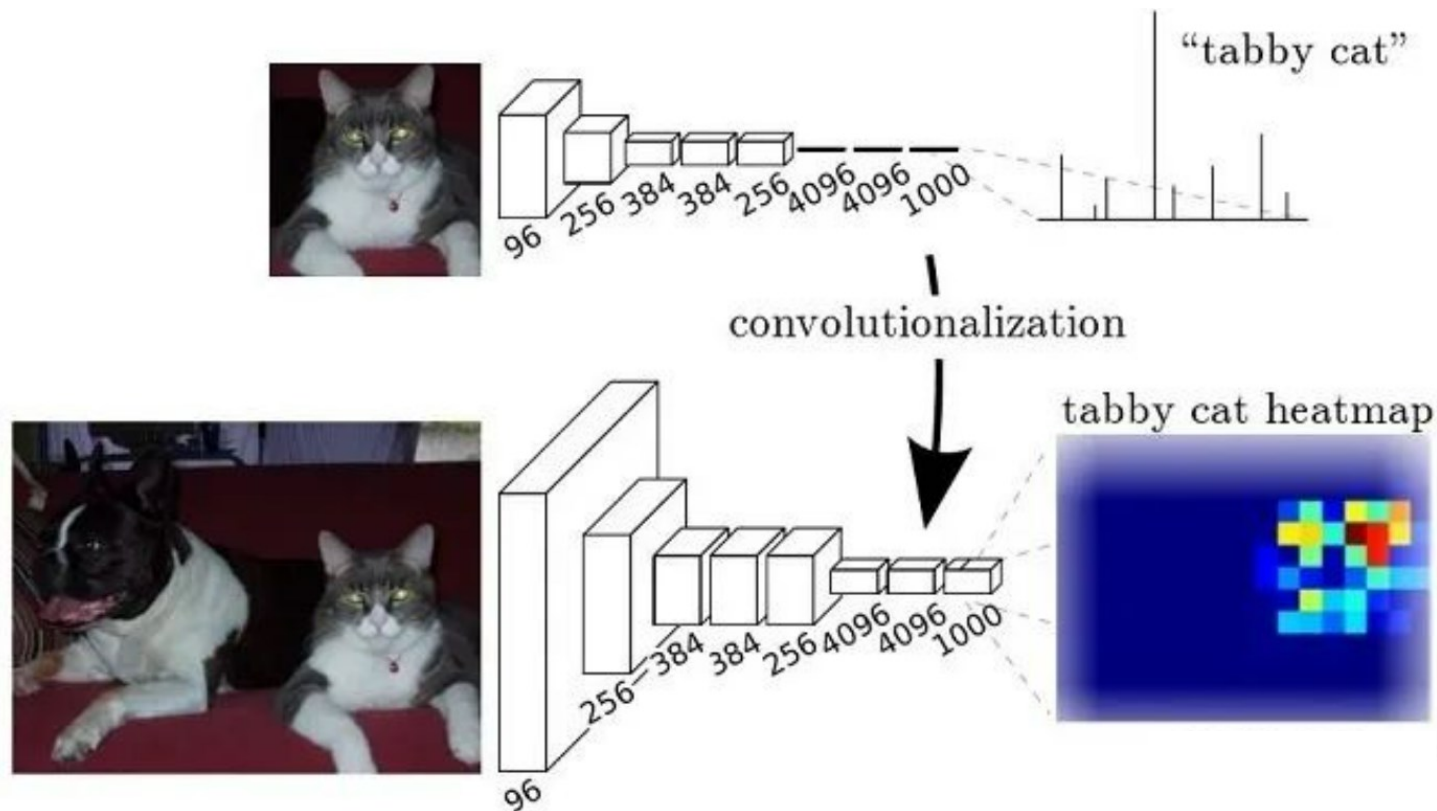
DenseNet

[Huang, CVPR, 2017]





# 经典CNN模型 -- Fully Convolutional Networks

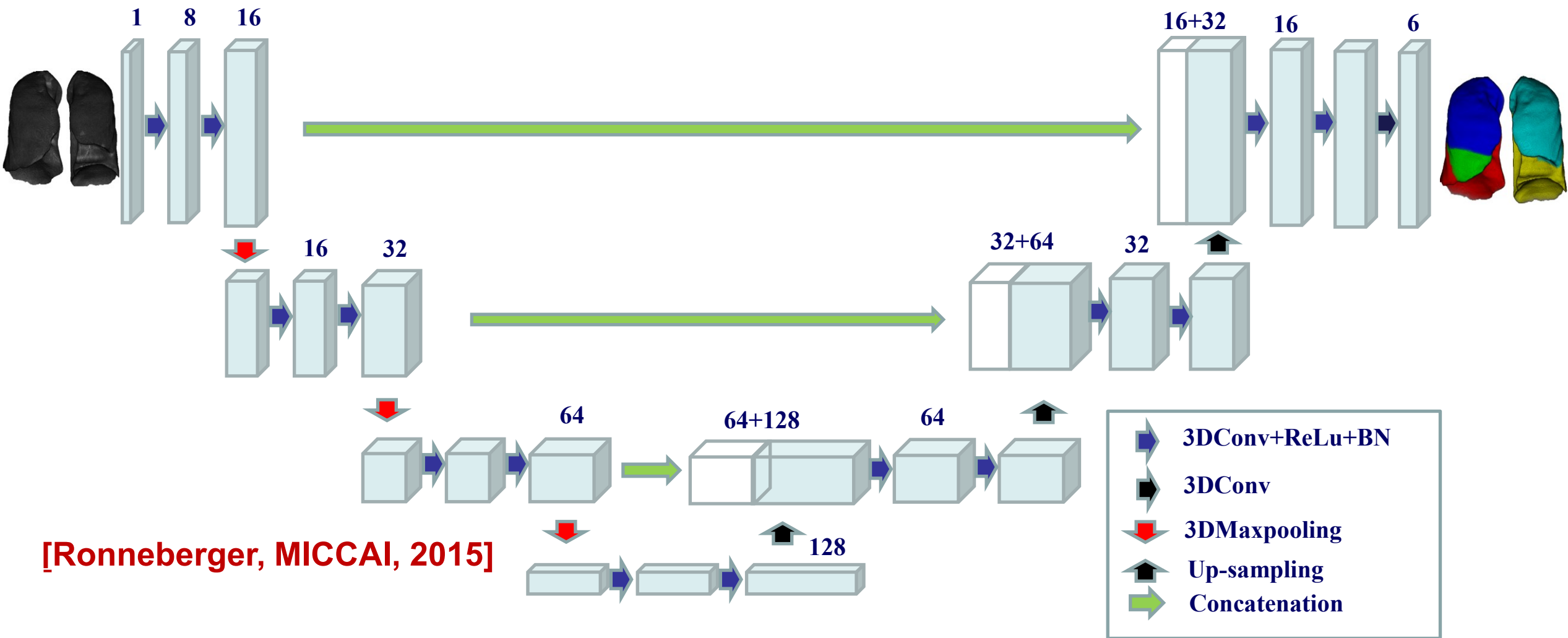


- 全卷积网络 -- 可适应任意尺寸输入
- 上采样 -- 增大图像尺寸，输出精细结果
- 忽略考虑像素之间的关系

[Long, CVPR, 2015]



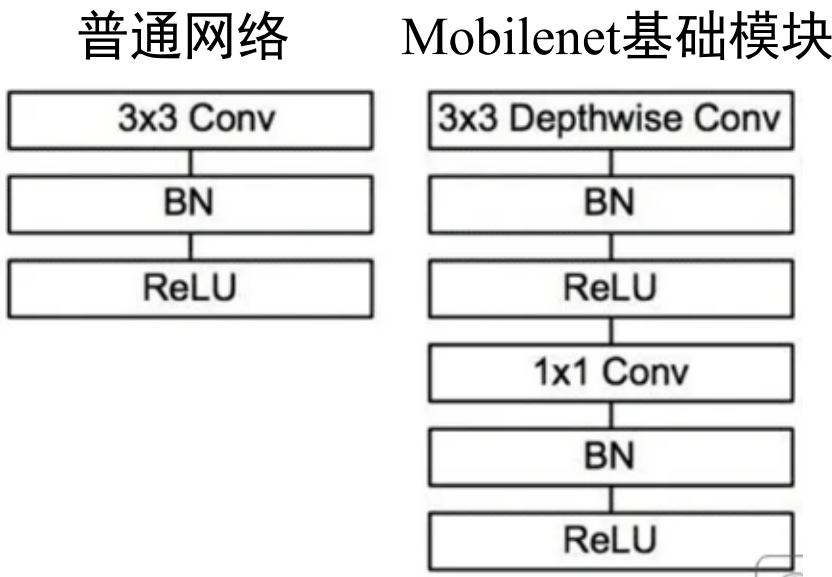
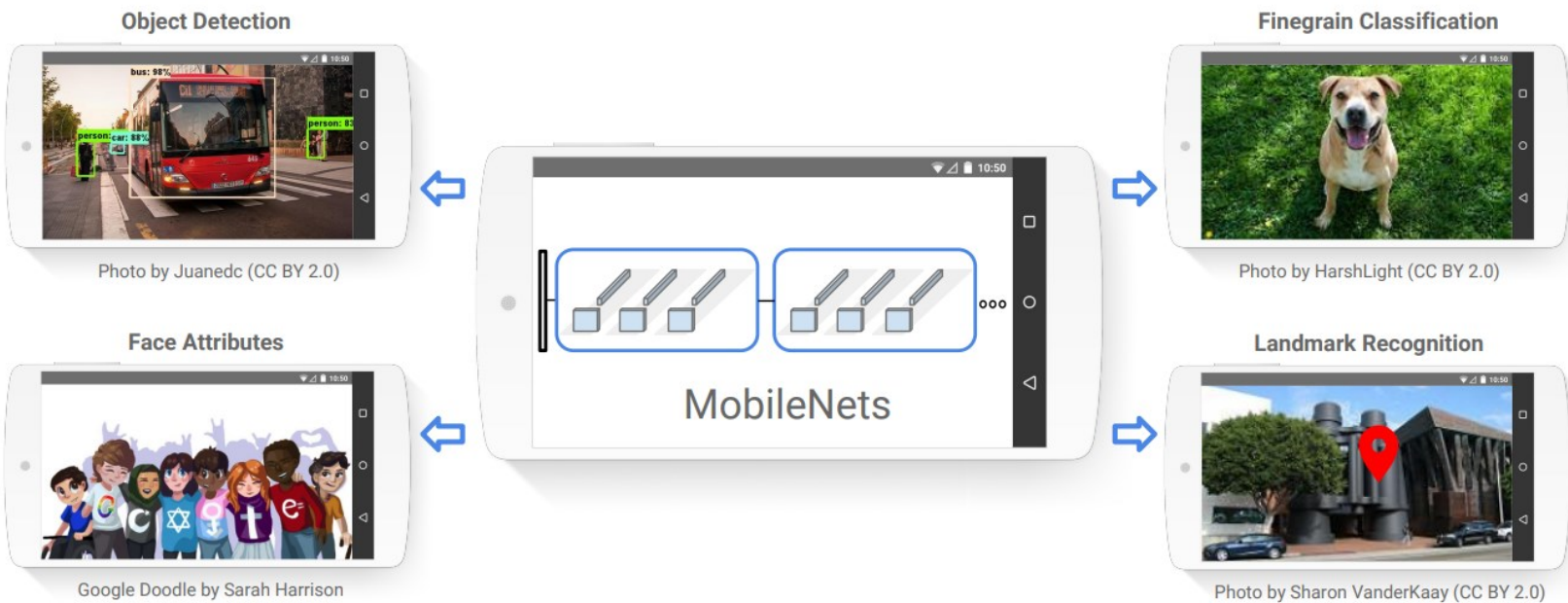
# 经典CNN模型-UNet分割网络



UNet: 下采样和上采样过程组成, 相当于对每个像素进行分类



# 轻量级CNN网络模型-Mob i l eNet

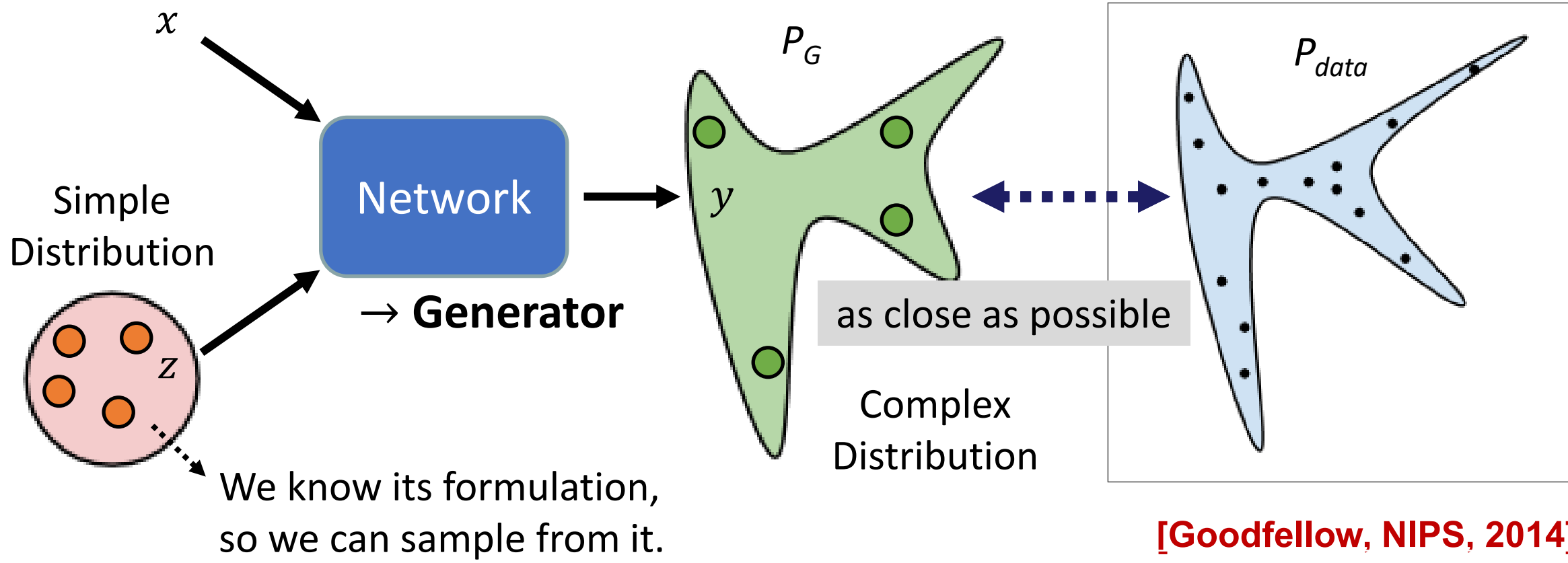


专注于CNN网络的移动端部署

[Howard, arXiv, 2017]



# 对抗生成网络模型-Generative Adversarial Network

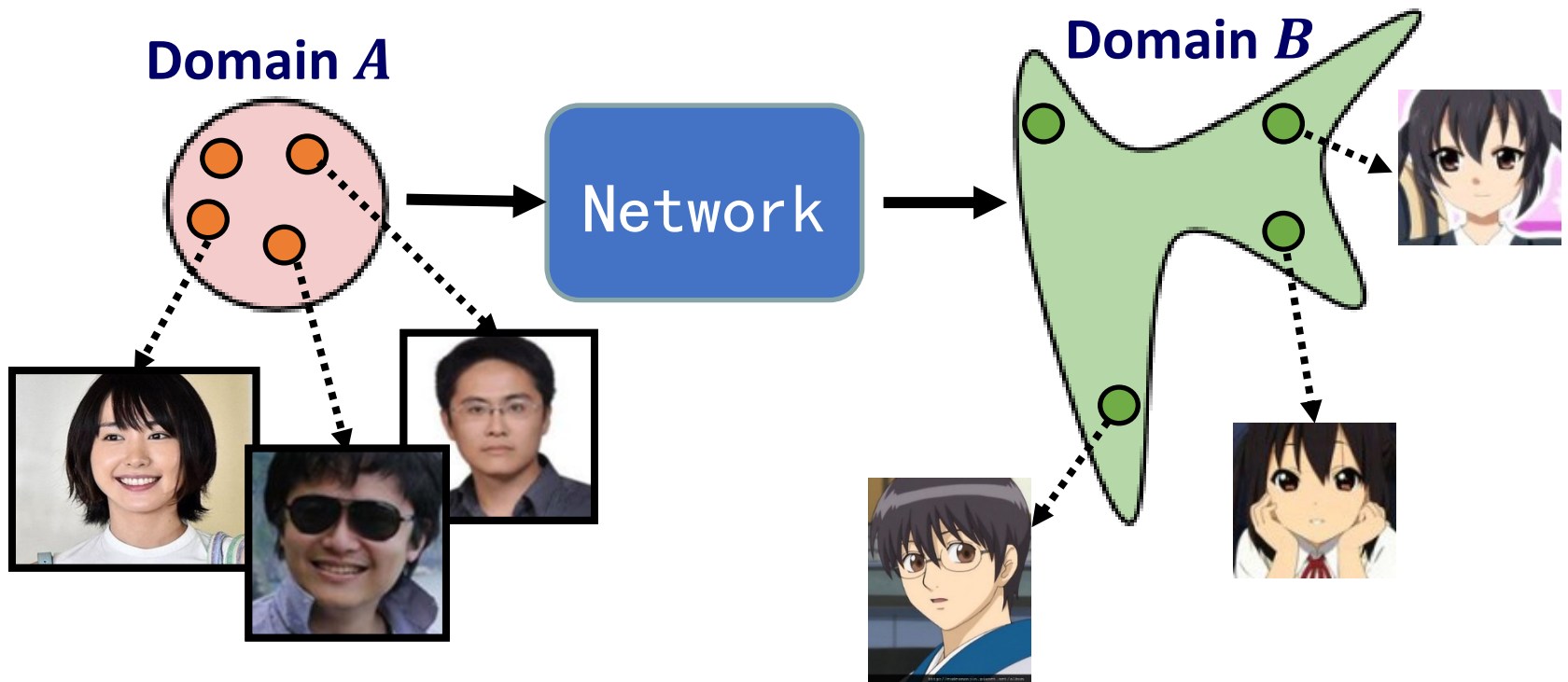


[Goodfellow, NIPS, 2014]

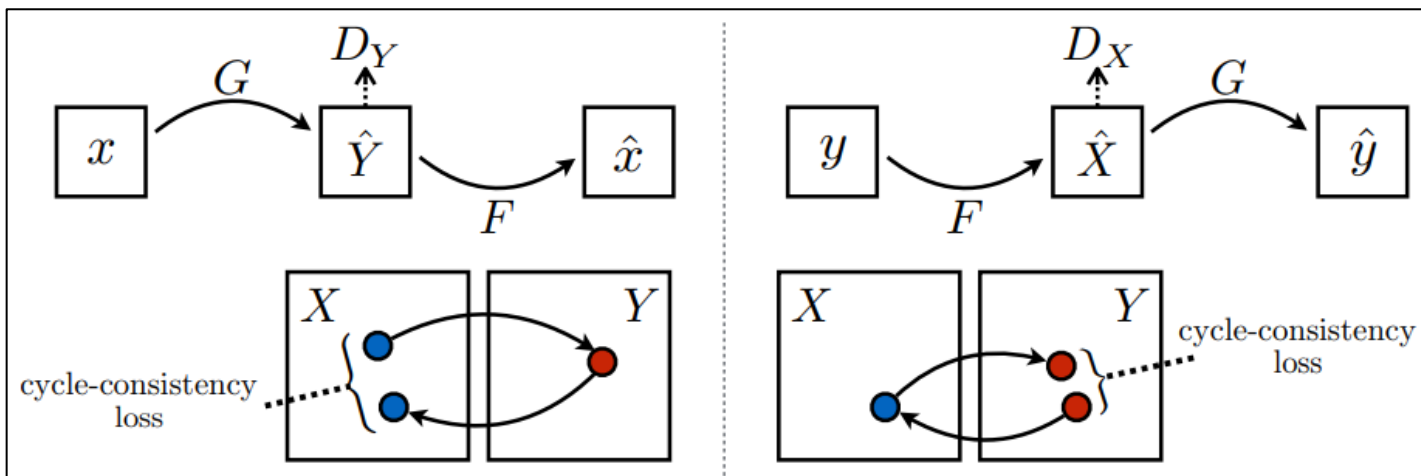
$$G^* = \arg \min_G \underline{Div}(P_G, P_{data})$$



# 对抗生成网络-CycleGAN

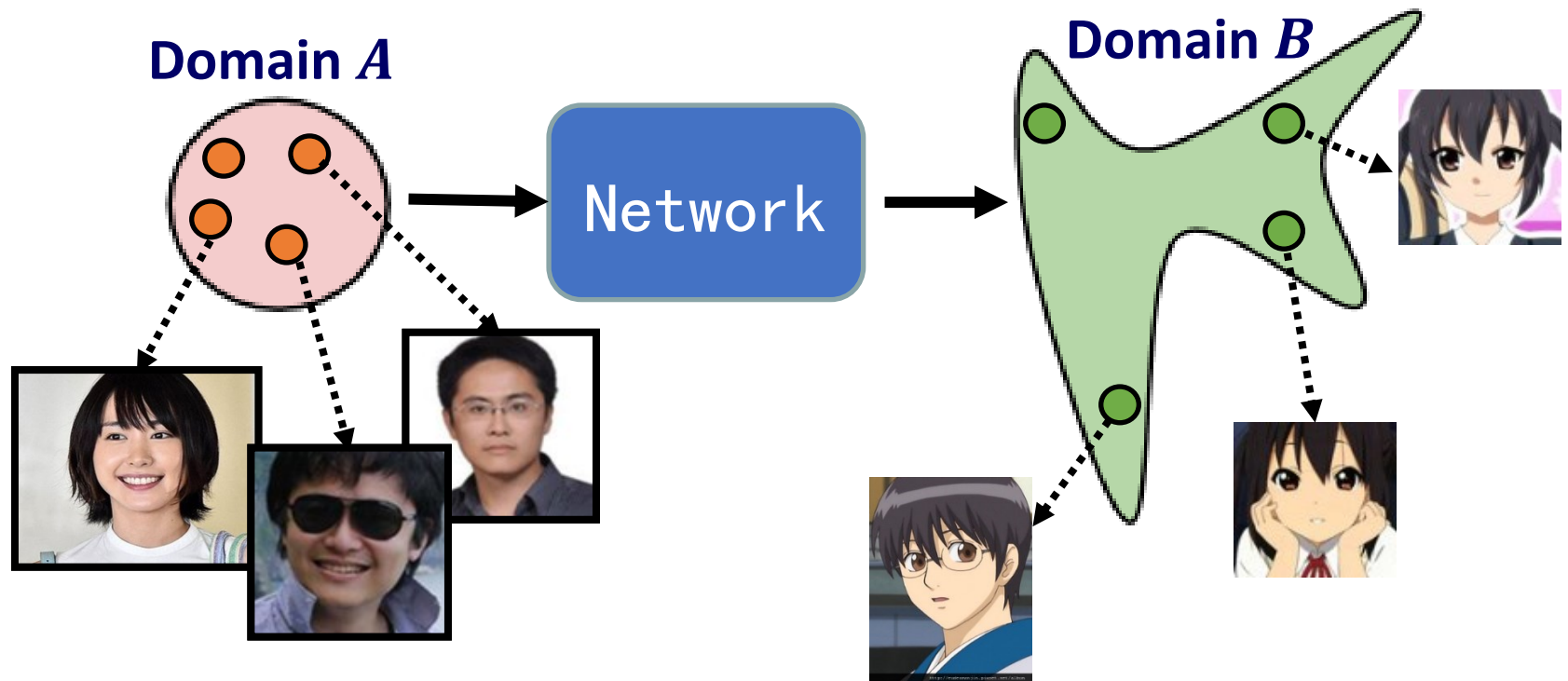


[Zhu, ICCV, 2017]





# 对抗生成网络-CycleGAN



[Zhu, ICCV, 2017]

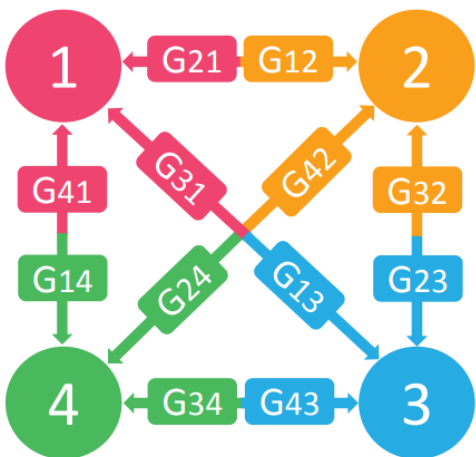
专注于未配对的图像到图像的转换



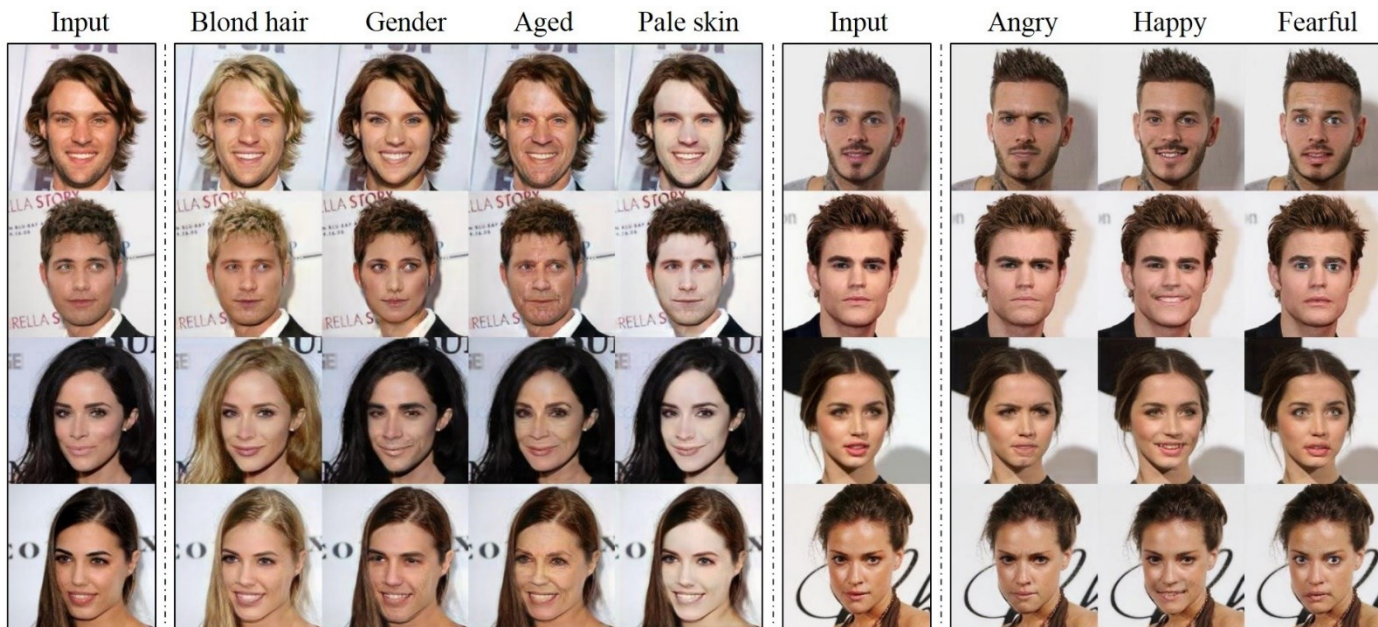
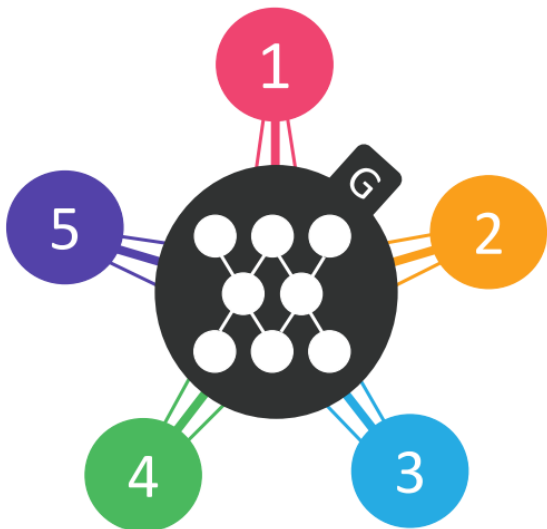


# 对抗生成网络-StarGAN

(a) Cross-domain models



(b) StarGAN



[Choi, CVPR, 2018]

专注于多域图像生成任务

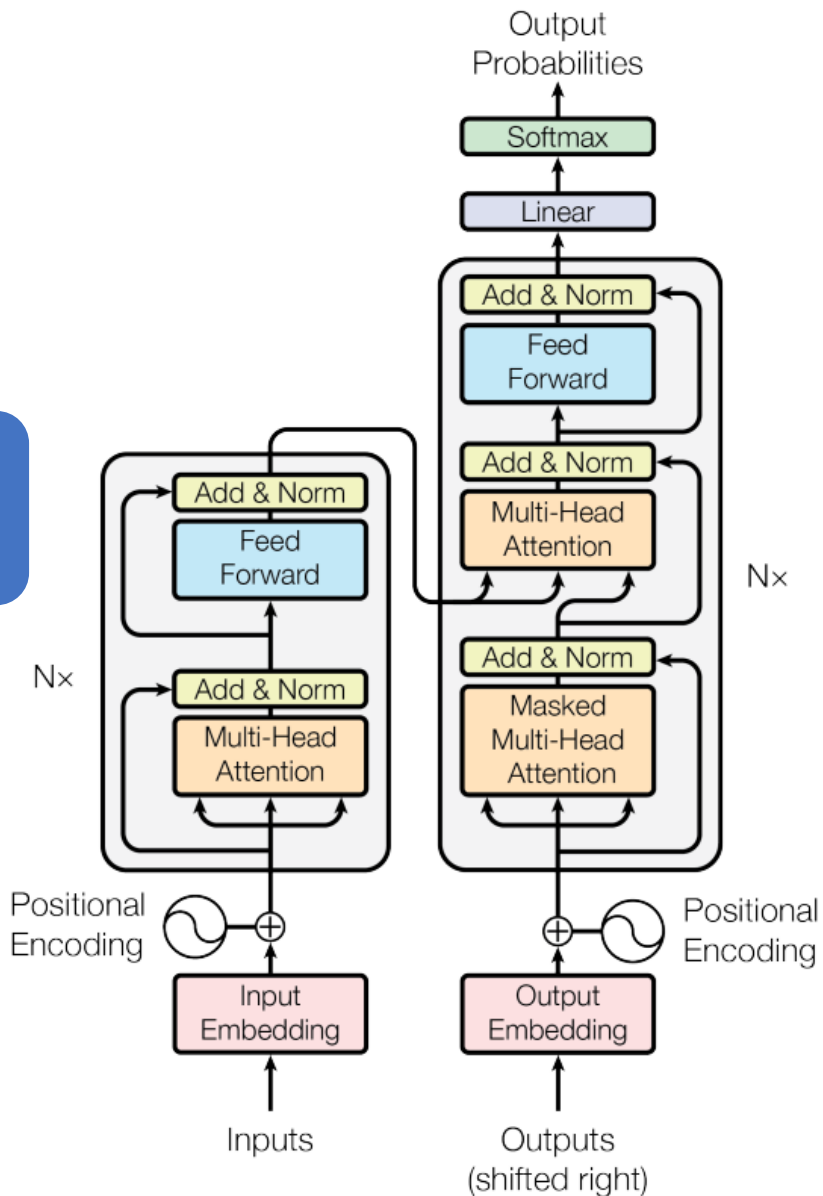


# 注意力机制



Attention is all you need!

编码器-解码器的结构

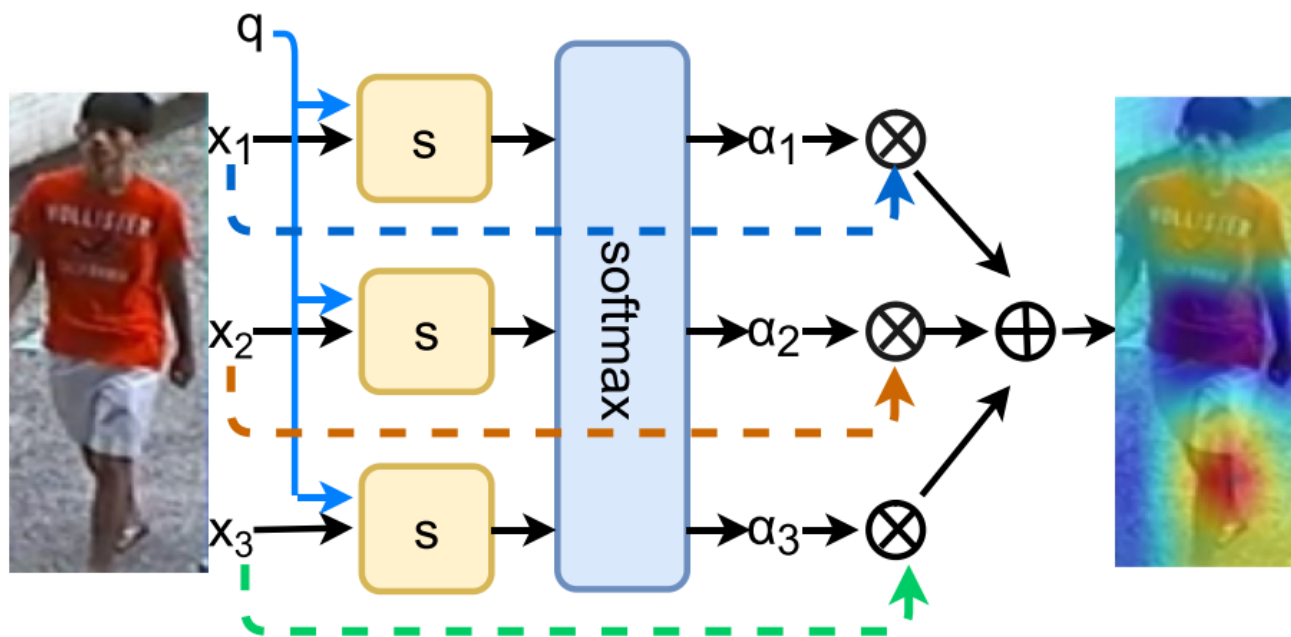
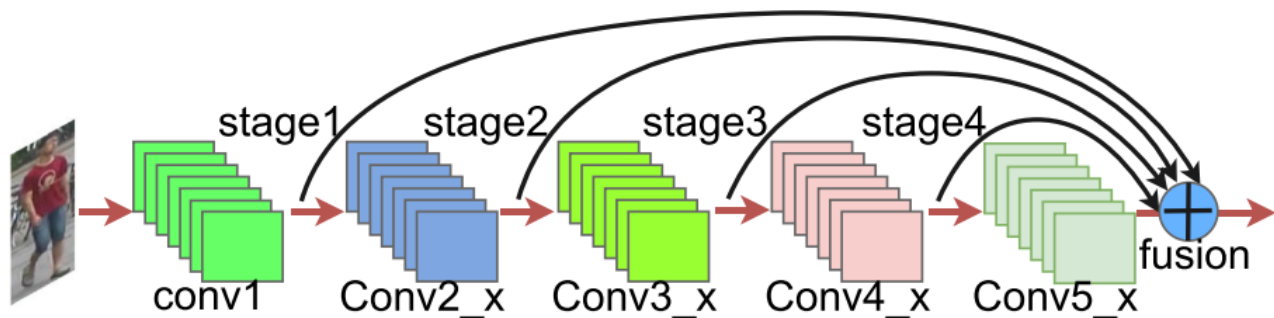


[Vaswani, NIPS, 2017]





# 图像检索中的注意力机制



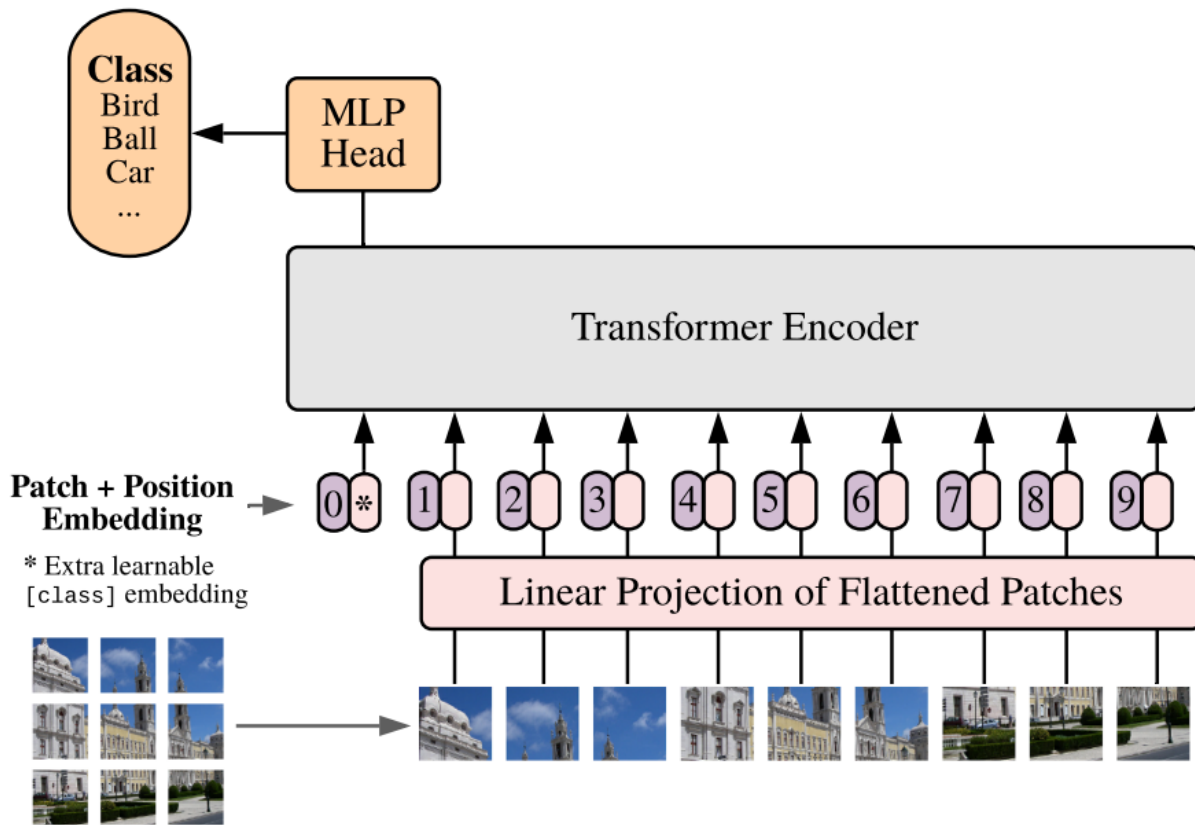
基于注意力机制的多尺度特征融合

[Xiang, MTA, 2020]

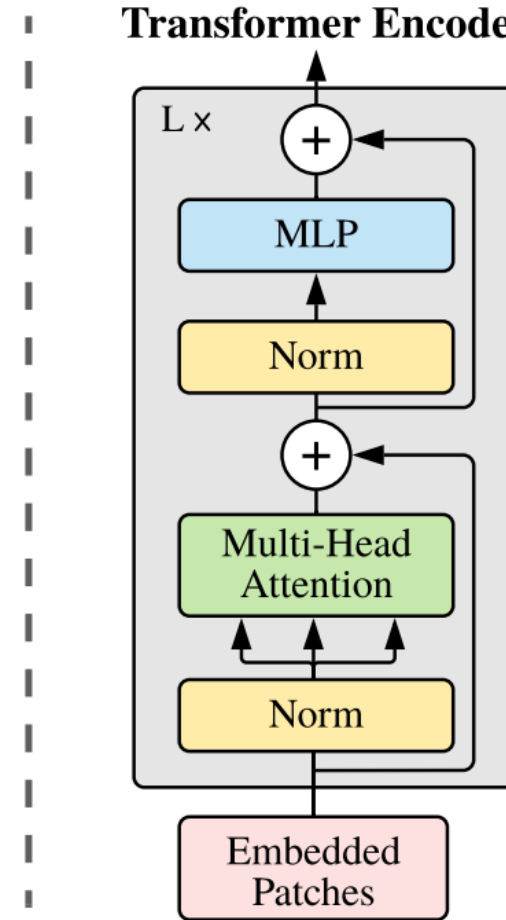


# Vision Transformer

Vision Transformer (ViT)



Transformer Encoder

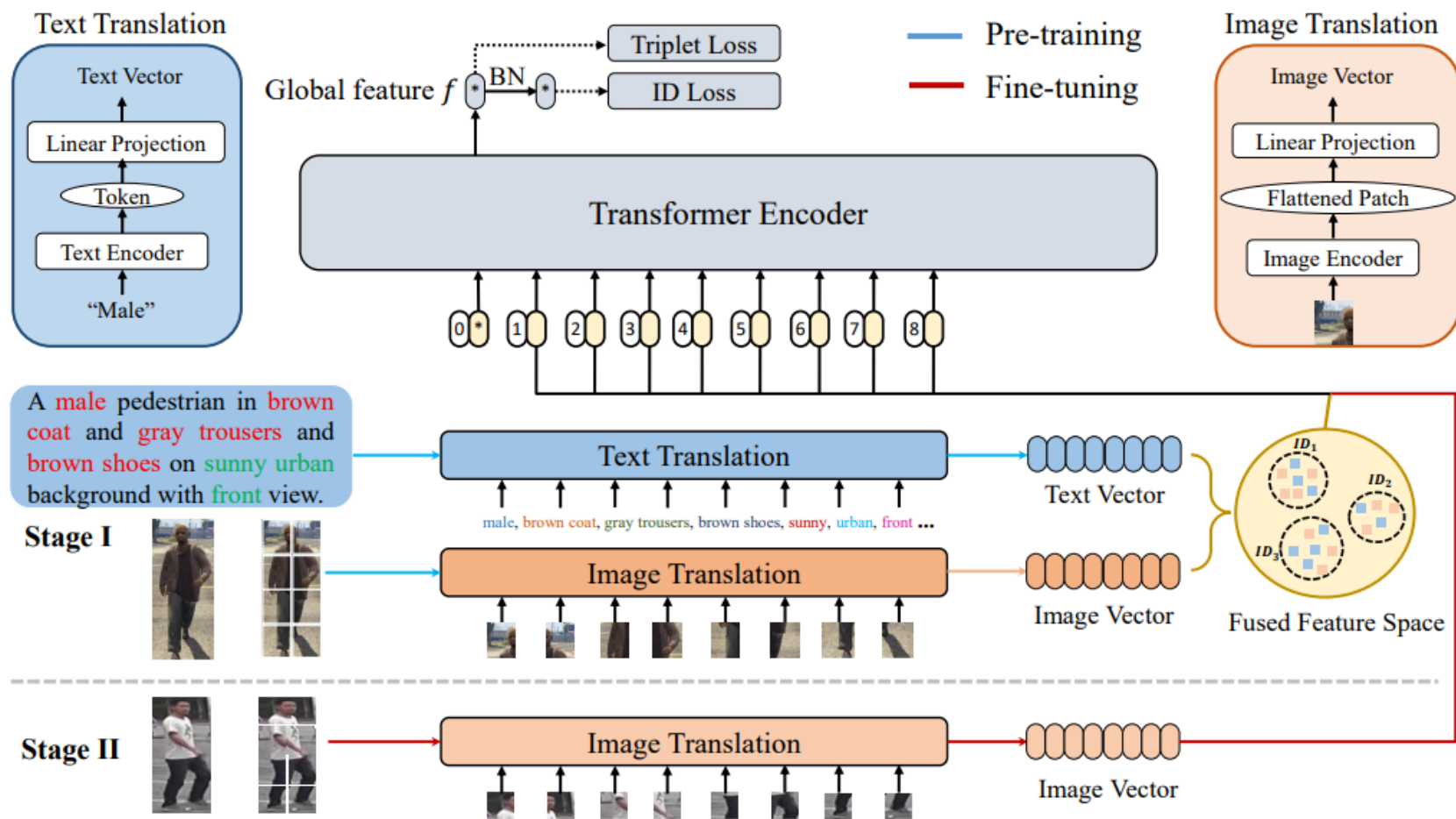


致力于打造视觉-文本的大一统模型

[Dosovitskiy, ICLR, 2021]



# Vision & Text with Transformer



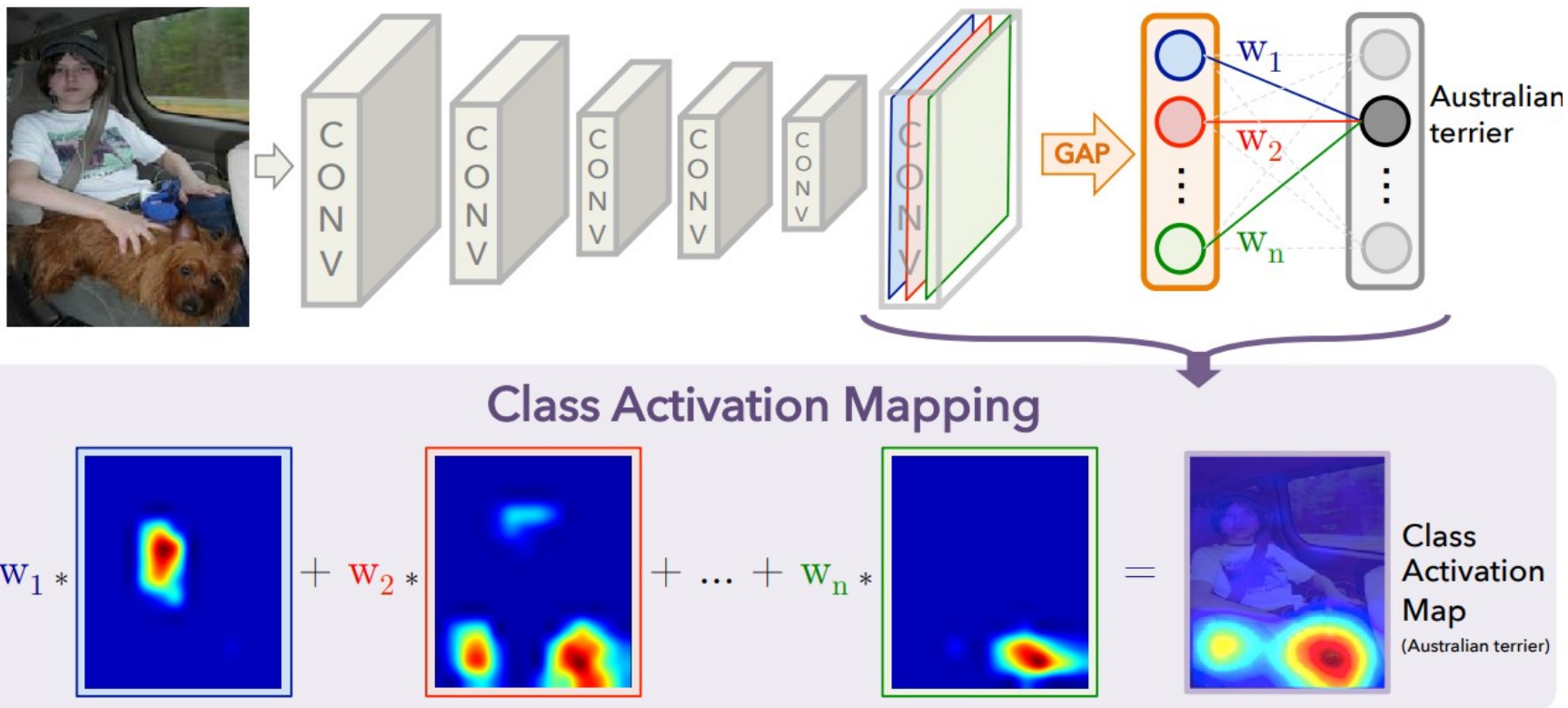
[Xiang, arXiv, 2022]

连接图像与文本的多模态融合模型



# CNN可解释性—热力图

Gradient-weighted Class Activation Mapping



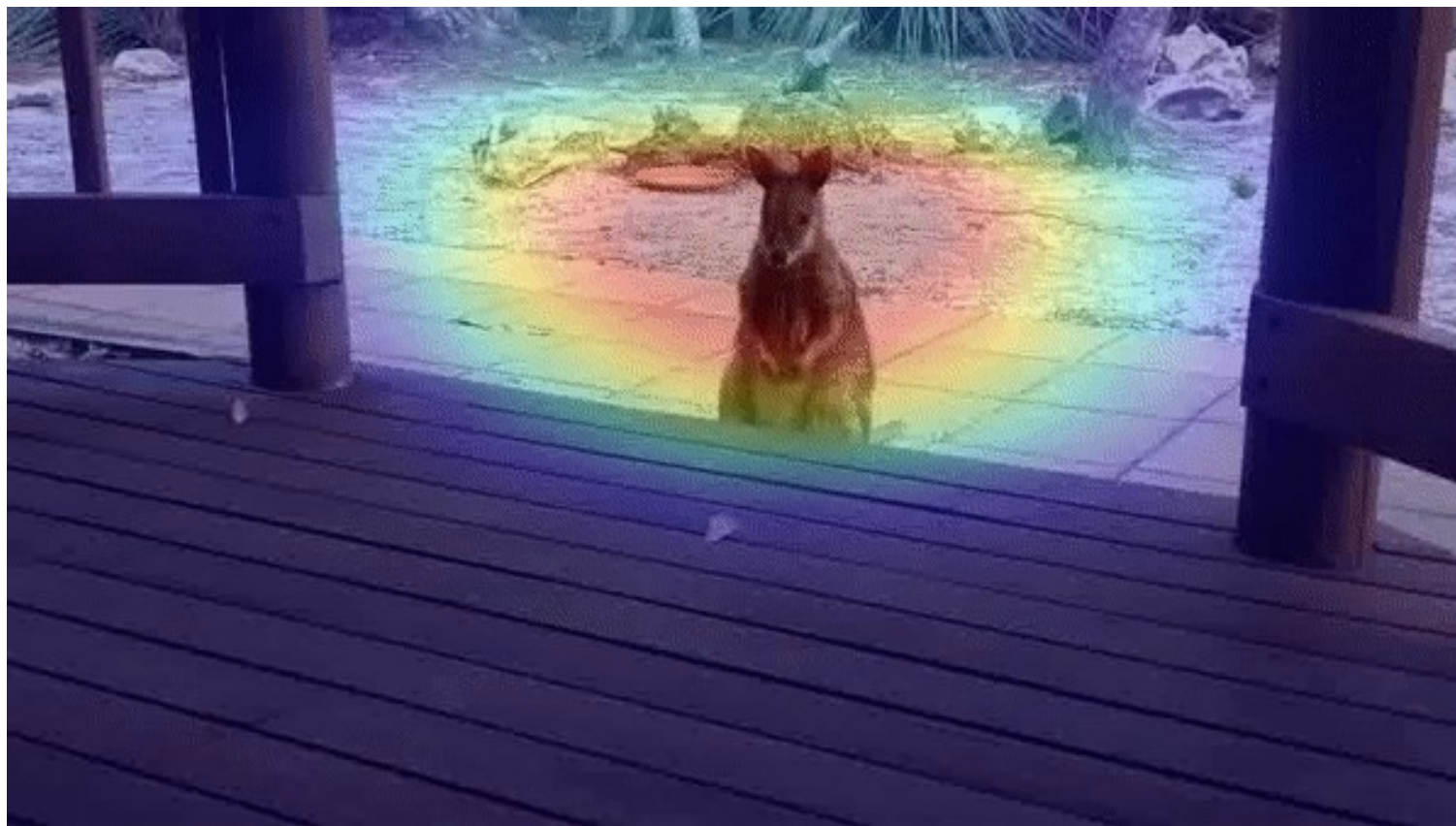
□ 优点：一定程度上对模型内部工作机制进行了可视化

[Zhou, CVPR, 2016]

□ 缺点：在医学领域作用有限，只能做主观解释



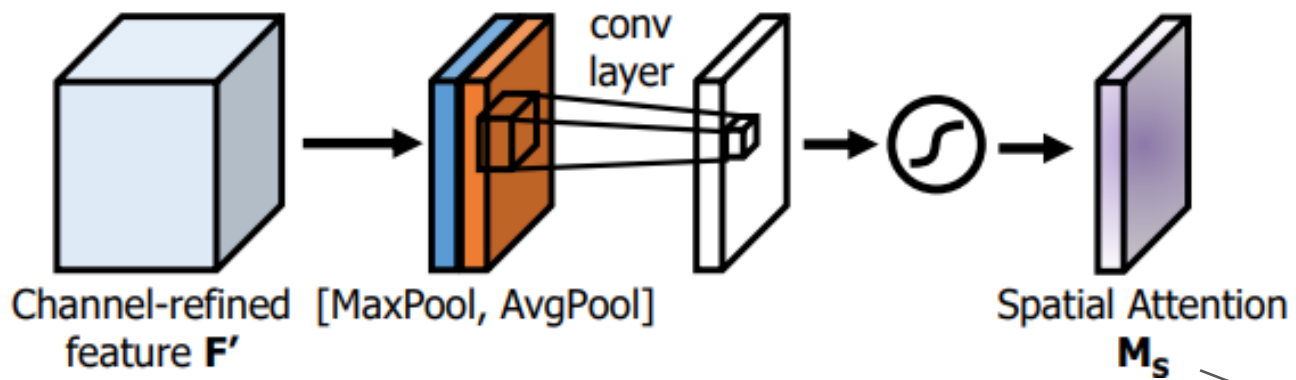
# 模型可视化



Activation maps created by Layer-CAM with a pretrained ResNet-18



# 模型可视化



[Mohammed, IJCNN, 2020]



# 总结

- 计算机视觉应用
- 机器学习/深度学习
- 计算机视觉深度网络