



# 人工智能导论

## 从神经网络到深度学习

郭兰哲

南京大学 智能科学与技术学院

Homepage: [www.lamda.nju.edu.cn/guolz](http://www.lamda.nju.edu.cn/guolz)

Email: [guolz@nju.edu.cn](mailto:guolz@nju.edu.cn)

# 大纲

---

- 神经元模型到前馈神经网络
- 参数优化：BP算法
- 深度学习
- 卷积神经网络
- 循环神经网络
- 生成对抗网络

# 两条路线

自图灵提出“机器与智能”起，一直有两派观点

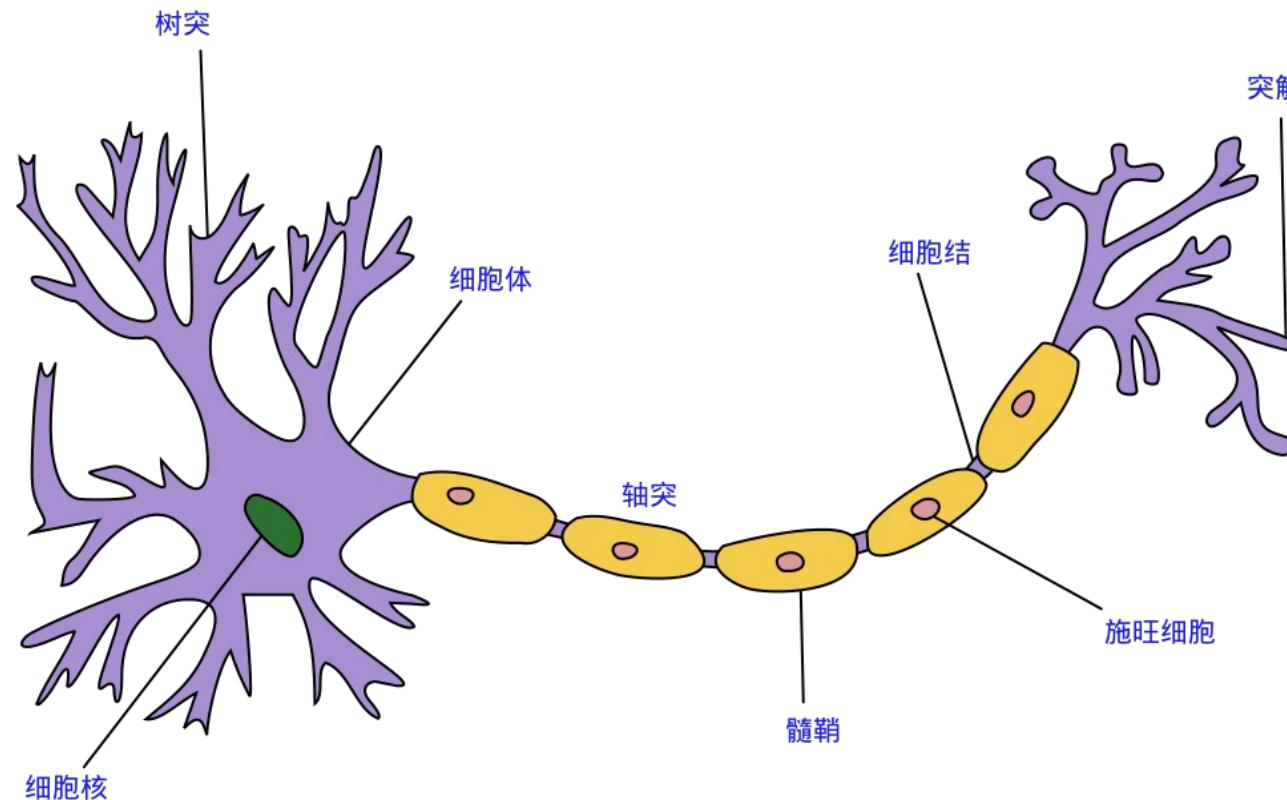
- 一派认为实现人工智能必须用逻辑和符号系统
- 另一派认为通过仿造大脑可以实现人工智能

想啥来啥 vs 吃啥补啥

“神经网络是由具有适应性的简单单元组成的广泛并行互连的网络，  
它的组织能够模拟生物神经系统对真实世界物体所作出的交互反应”

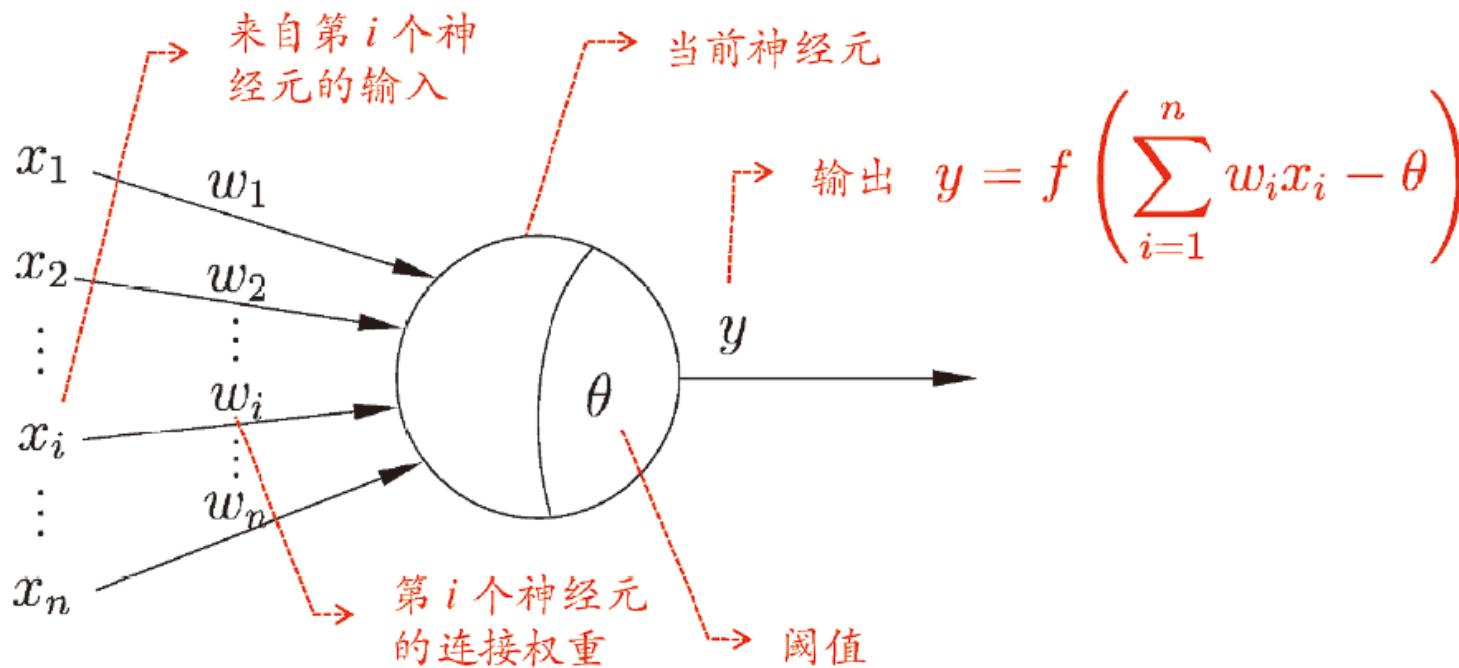
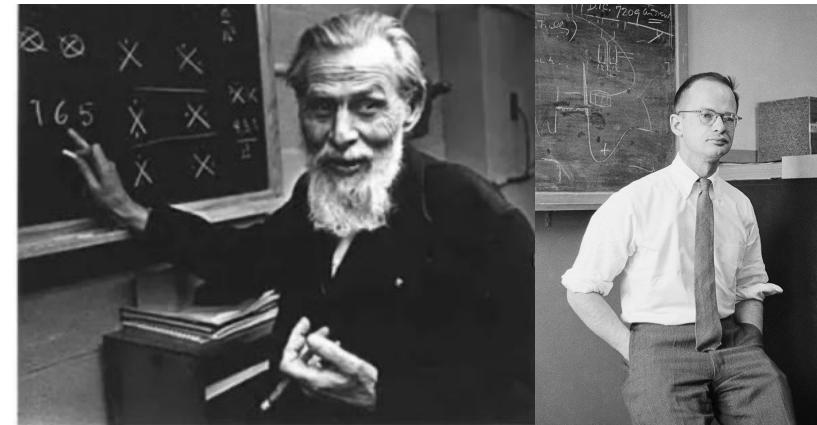
# 生物神经元结构

- 每个神经元与其他神经元相连，当它“兴奋”时，就会向相连的神经元发送化学物质，从而改变这些神经元内的电位
- 如果某神经元的电位超过一个“阈值”，那么它就会被激活，向其它神经元发送化学物质



# 神经元的数学模型

1943年，麦卡洛克(Warren McCulloch)和皮茨(Walter Pitts)提出了神经元的数学模型：M-P神经元模型

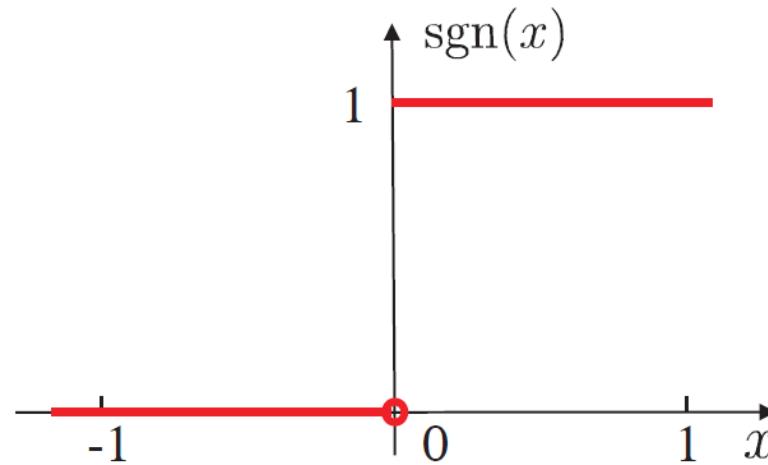


Warren McCulloch

Walter Pitts

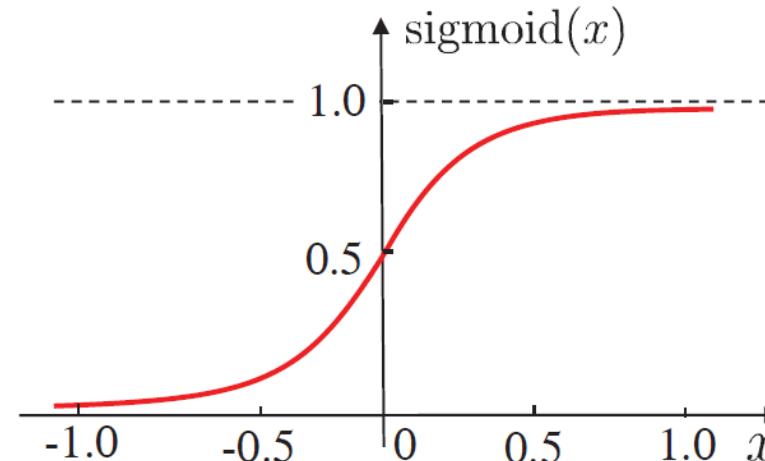
# 神经元的激活函数

- 理想激活函数是阶跃函数, 0表示抑制神经元而1表示激活神经元
- 阶跃函数具有不连续、不光滑等不好的性质, 常用的是 Sigmoid 函数



$$\text{sgn}(x) = \begin{cases} 1, & \text{if } x \geq 0; \\ 0, & \text{if } x < 0. \end{cases}$$

(a) 阶跃函数



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

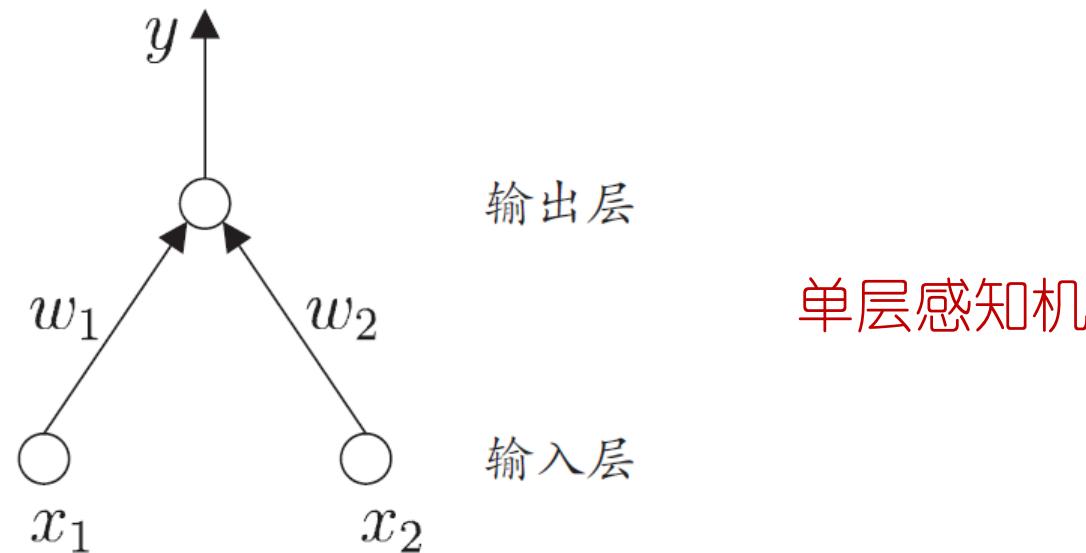
(b) Sigmoid 函数

# 感知机(Perceptron)

1957年，康奈尔大学的罗森布拉特(Frank Rosenblatt)  
在一台IBM-704计算机上模拟实现了感知机模型



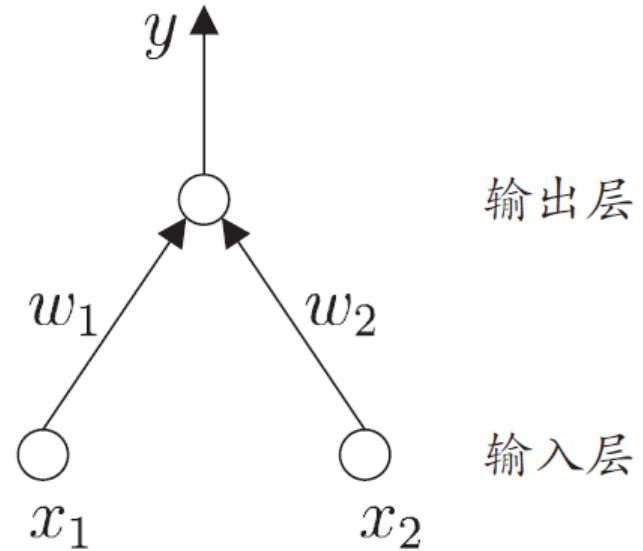
Frank Rosenblatt



罗森布拉特在理论上证明了单层感知机在处理线性可分的问题时，可以收敛，并以此为基础做了若干感知机有学习能力的实验

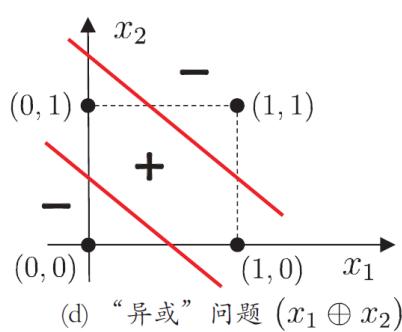
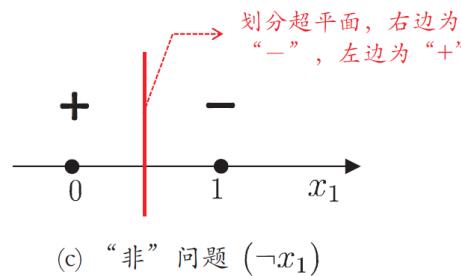
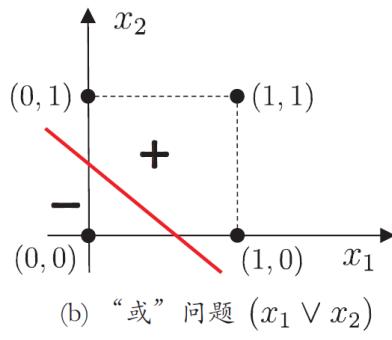
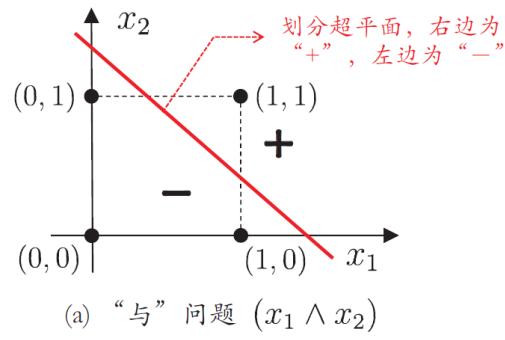
# 感知机(Perceptron)

- 感知机能够容易实现逻辑与、或、非运算
- 当 $w_1, w_2, \theta$ 为多少时，可以实现与运算 ( $x_1 \wedge x_2$ )
- 当 $w_1, w_2, \theta$ 为多少时，可以实现或运算 ( $x_1 \vee x_2$ )
- 当 $w_1, w_2, \theta$ 为多少时，可以实现非运算 ( $\neg x_1$ )



# 感知机(Perceptron)

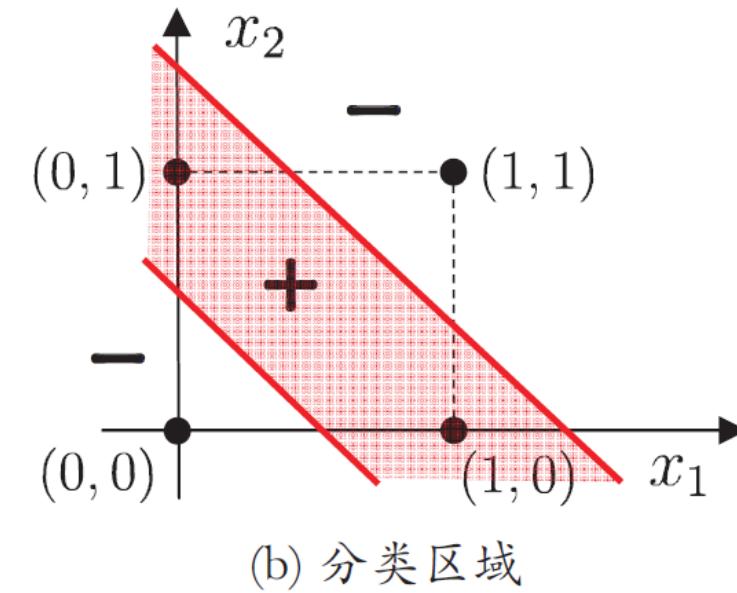
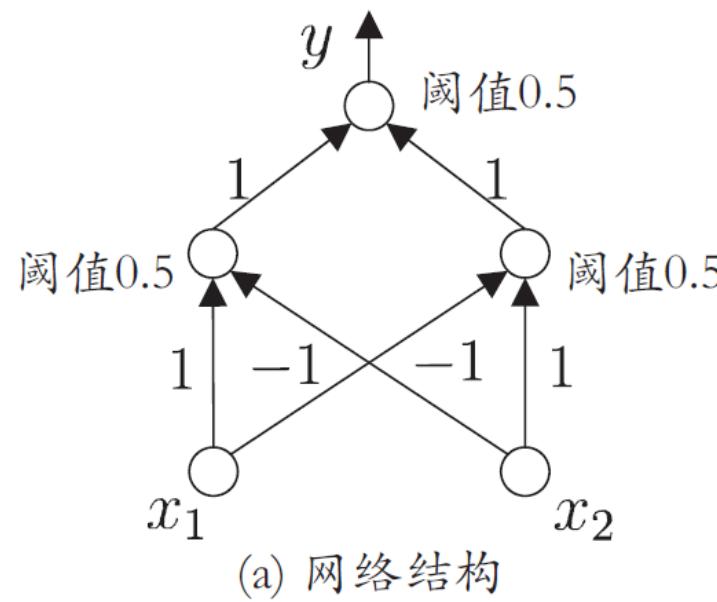
明斯基和佩铂特在《感知机：计算几何学》书中证明单层感知机不能解决异或XOR问题



- 当两类模式线性可分时，则感知机的学习过程一定会收敛；否则感知机的学习过程将会发生震荡 [Minsky and Papert, 1969]

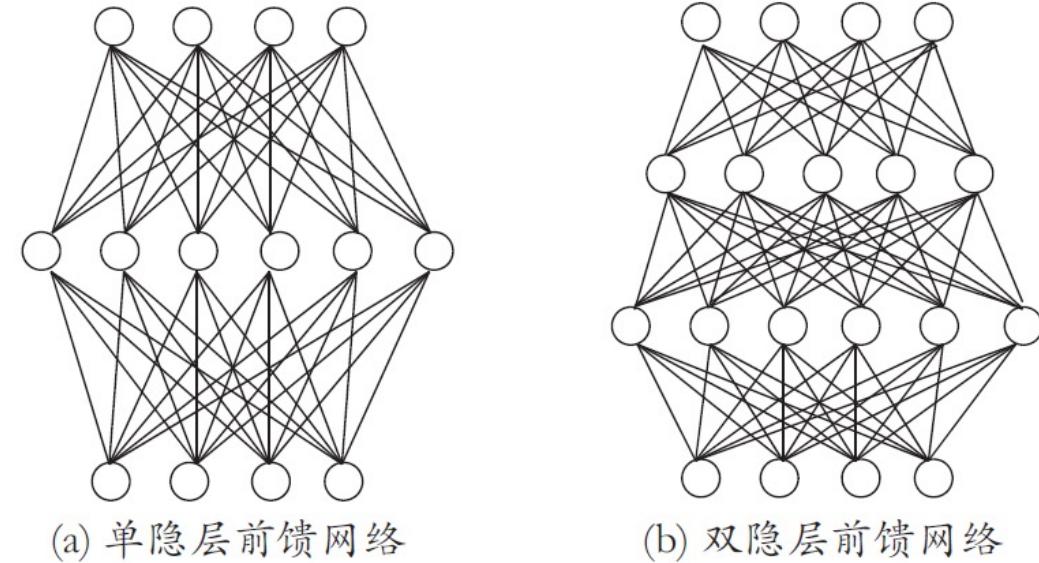
# 多层感知机

1974年哈佛大学沃波斯(Paul Werbos)的博士论文证明了在神经网络多加一层，可以解决XOR问题



# 前馈神经网络

- 将输出层与输入层之间的一层神经元，称为**隐层或隐含层**
- 隐含层和输出层神经元都是具有激活函数的功能神经元
- 前馈网络：各个神经元接受前一级的输入，并输出到下一级，层与层之间通过“全连接”进行链接，即两个相邻层之间的神经元完全成对连接，但层内的神经元不相互连接



多层次前馈网络有强大的表示能力（“**万有逼近性**”）

仅需一个包含足够多神经元的隐层，多层次前馈神经网络就能以任意精度逼近任意复杂度的连续函数 [Hornik et al., 1989]

# 大纲

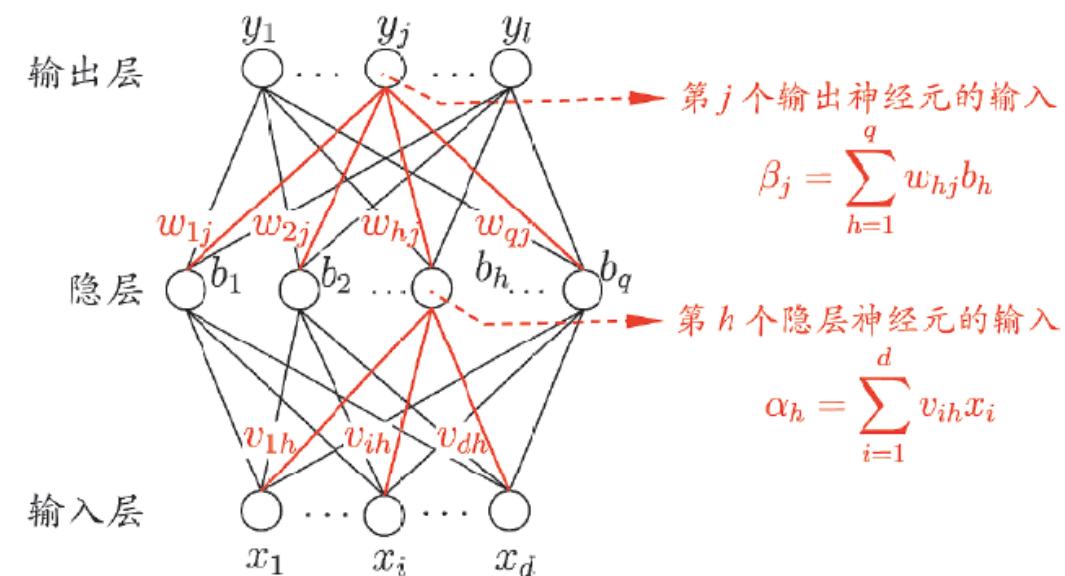
---

- 神经元模型到前馈神经网络
- **参数优化：BP算法**
- 深度学习
- 卷积神经网络
- 循环神经网络
- 生成对抗网络

# 网络参数优化

- 给定训练集  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ ,  $x_i \in \mathbb{R}^d$ ,  $y_i \in \mathbb{R}^l$

- 输入:  $d$  维特征向量
- 输出: 1 个输出值
- 隐层:  $q$  个隐层神经元
- 假定功能单元均使 Sigmoid 函数



- 需通过学习确定的参数数目:  $(d + l + 1)q + l$

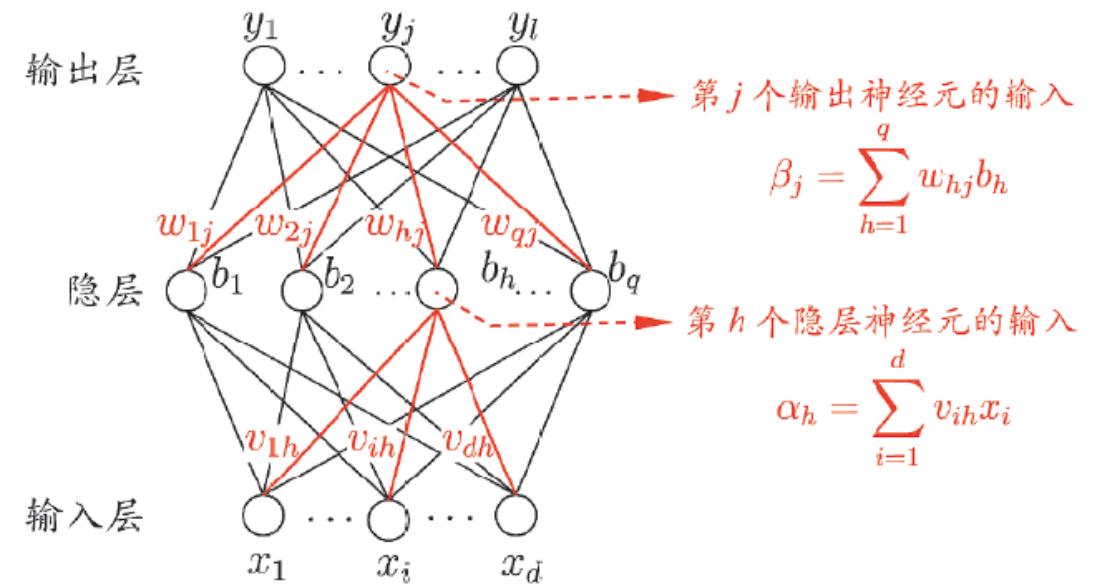
# 网络参数优化-前向计算

对于训练样本 $(x_k, y_k)$ , 假定网络的实际输出为 $\hat{y}_k = (\hat{y}_1^k, \hat{y}_2^k, \dots, \hat{y}_l^k)$

$$\hat{y}_j^k = f(\beta_j - \theta_j)$$

则网络在样本 $(x_k, y_k)$ 上的均方误差为:

$$E_k = \frac{1}{2} \sum_{j=1}^l (\hat{y}_j^k - y_j^k)^2$$



如何根据误差优化网络的参数?

# 梯度下降(Gradient Descent)

- 梯度下降算法是一种使得函数最小化的迭代式优化方法
- 函数 $f$ 在 $w$ 处的取值为 $f(w)$ , 梯度为 $\nabla f(w)$
- 参数更新方法:

$$w_i = w_{i-1} - \eta \nabla f(w_{i-1})$$

$\eta$ 一般被称为步长(step size)或者学习率(learning rate)

**Quiz:** 为什么要沿着梯度的方向?

学习率  $\eta \in (0, 1)$  不能太大、不能太小



# BP (BackPropagation:误差逆传播算法)

迄今最成功、最常用的神经网络算法，可用于多种任务（不仅限于分类）

P. Werbos在博士学位论文中正式完整描述：

P. Werbos. Beyond regression: New tools for prediction and analysis in the behavioral science. Ph.D dissertation, Harvard University, 1974

对于训练样本 $(x_k, y_k)$ ，假定网络的实际输出为 $\hat{y}_k = (\hat{y}_1^k, \hat{y}_2^k, \dots, \hat{y}_l^k)$

$$\hat{y}_j^k = f(\beta_j - \theta_j)$$

则网络在样本 $(x_k, y_k)$ 上的均方误差为：

$$E_k = \frac{1}{2} \sum_{j=1}^l (\hat{y}_j^k - y_j^k)^2$$

BP 是一个基于梯度下降的迭代学习算法，对于网络中的每一个参数 $w$ 在迭代的每一轮中更新方式如下：

$$w = w - \eta \frac{\partial E}{\partial w}$$

# BP (BackPropagation:误差逆传播算法)

以 $w_{hj}$ 为例

对误差  $E_k = \frac{1}{2} \sum_{j=1}^l (\hat{y}_j^k - y_j^k)^2$ ,

如何计算

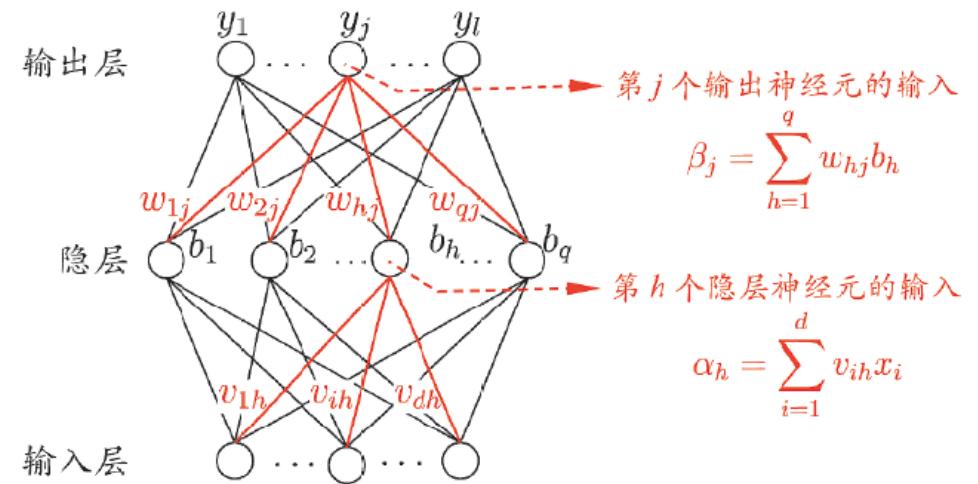
$$\frac{\partial E_k}{\partial w_{hj}}$$

注意到  $w_{hj}$  先影响到  $\beta_j$ ,

再影响到  $\hat{y}_j^k$ , 然后才影响到  $E_k$ , 有:

$$\frac{\partial E_k}{\partial w_{hj}} = \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{hj}}$$

“链式法则”



# BP (BackPropagation:误差逆传播算法)

$$\frac{\partial E_k}{\partial w_{hj}} = \boxed{\frac{\partial E_k}{\partial \hat{y}_j^k}} \cdot \boxed{\frac{\partial \hat{y}_j^k}{\partial \beta_j}} \cdot \boxed{\frac{\partial \beta_j}{\partial w_{hj}}} = b_h$$

$(\hat{y}_j^k - y_j^k)$

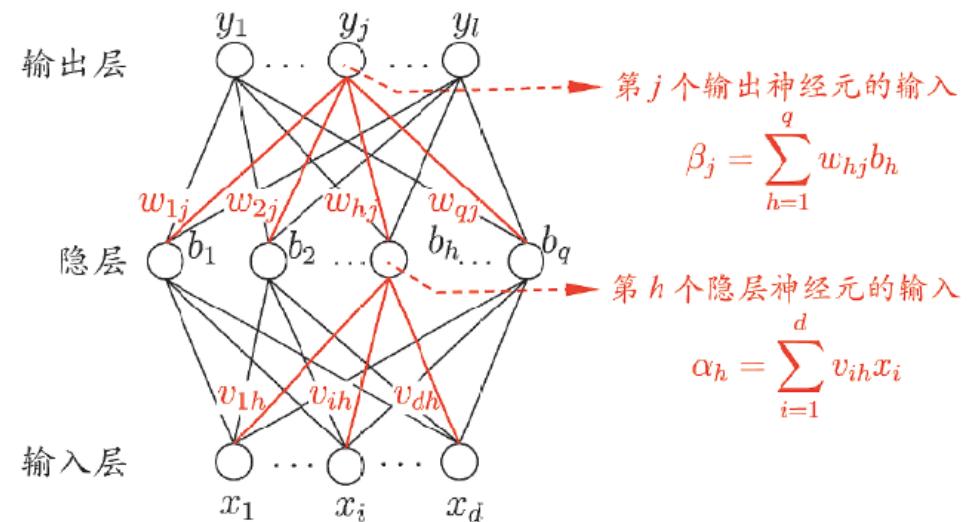
注意到

$$\hat{y}_j^k = f(\beta_j - \theta_j)$$

$f'(\beta_j - \theta_j)$

$\hat{y}_j^k(1 - \hat{y}_j^k)$

$$\begin{aligned} \Leftrightarrow g_j &= -\frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \\ &= \hat{y}_j^k(1 - \hat{y}_j^k)(y_j^k - \hat{y}_j^k) \end{aligned}$$



对  $\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$ , 有

$$f'(x) = f(x)(1 - f(x))$$

于是,

$$\Delta w_{hj} = -\eta \frac{\partial E_k}{\partial w_{hj}} = \eta g_j b_h$$

# BP (BackPropagation:误差逆传播算法)

类似地，有：

$$\Delta\theta_j = -\eta g_j$$

$$\Delta v_{ih} = \eta e_h x_i$$

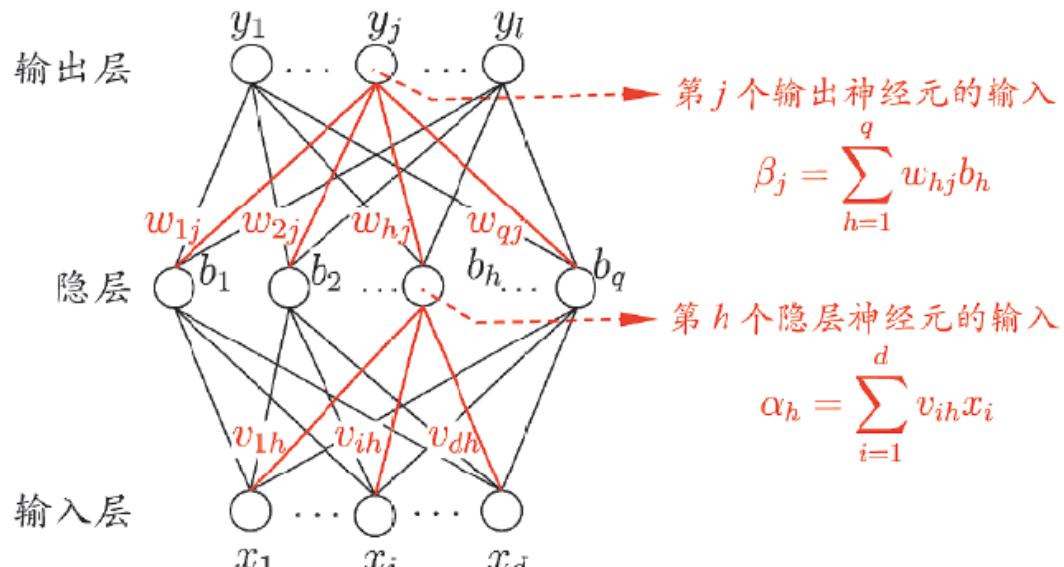
$$\Delta\gamma_h = -\eta e_h$$

其中：

$$e_h = -\frac{\partial E_k}{\partial b_h} \cdot \frac{\partial b_h}{\partial \alpha_h}$$

$$= -\sum_{j=1}^l \frac{\partial E_k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial b_h} f'(\alpha_h - \gamma_h) = \sum_{j=1}^l w_{hj} g_j f'(\alpha_h - \gamma_h)$$

$$= b_h(1 - b_h) \sum_{j=1}^l w_{hj} g_j$$



# 多种实现方式

## 批量梯度下降 (batch gradient descent)

- 每次在整个训练集上计算损失误差
- 每次需要读取完整数据，消耗内存，收敛速度较慢

## 随机梯度下降 (stochastic gradient descent)

- 每次随机选一个样本计算损失误差
- 参数更新频繁，不同样例可能抵消

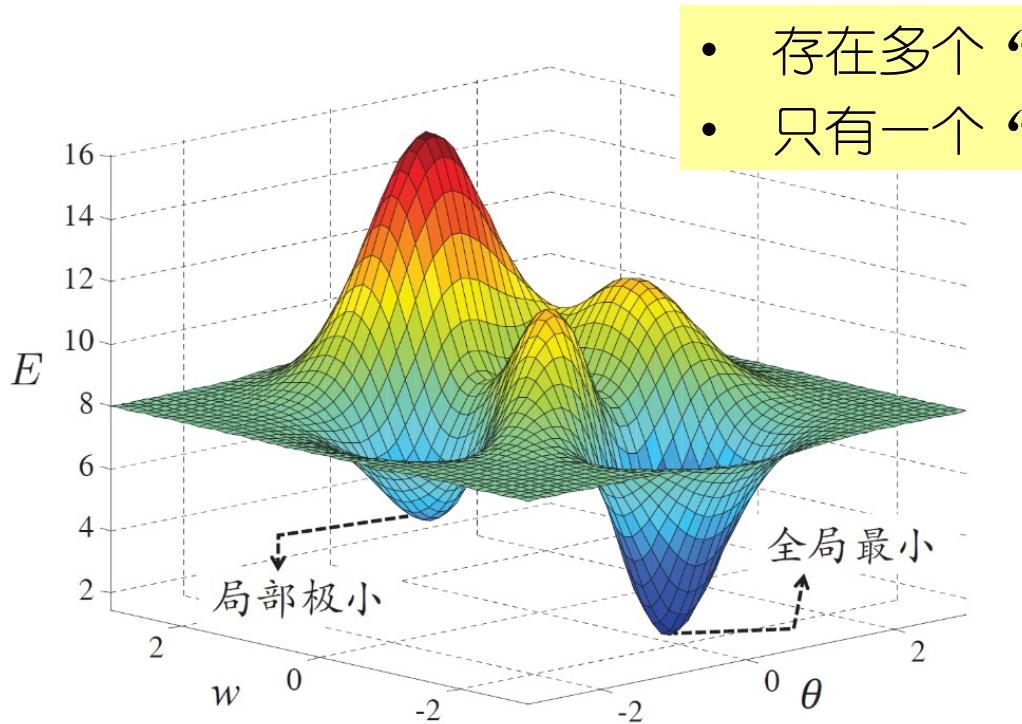
## 小批量梯度下降 (mini-batch gradient descent)

- 每次随机选一批样本计算损失误差读取整个训练集
- 训练过程更稳定，如何设置batch-size是个难题

# 全局最小 vs. 局部极小

神经网络的训练过程可看作一个参数寻优过程：

在参数空间中，寻找一组最优参数使得误差最小



- 存在多个“局部极小”
- 只有一个“全局最小”

“跳出”局部极小的常见策略：

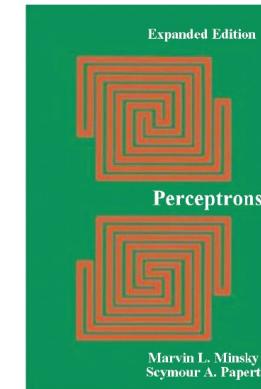
- ✓ 不同的初始参数
- ✓ 随机扰动
- ✓ 模拟退火
- ✓ 演化算法
- ✓ .....

# 神经网络发展回顾

1940年代-萌芽期：M-P模型 (1943), Hebb 学习规则 (1945)

1956左右-1969左右~繁荣期：感知机 (1958), Adaline (1960), ...

1969年：Minsky & Papert “Perceptrons”



马文·闵斯基  
(1927–2016)  
1969年图灵奖

1984左右 -1997左右~繁荣期：Hopfield (1983), BP (1986), ...

1997年左右：SVM文本分类成功 及 统计学习 兴起



2012-至今~繁荣期：深度学习

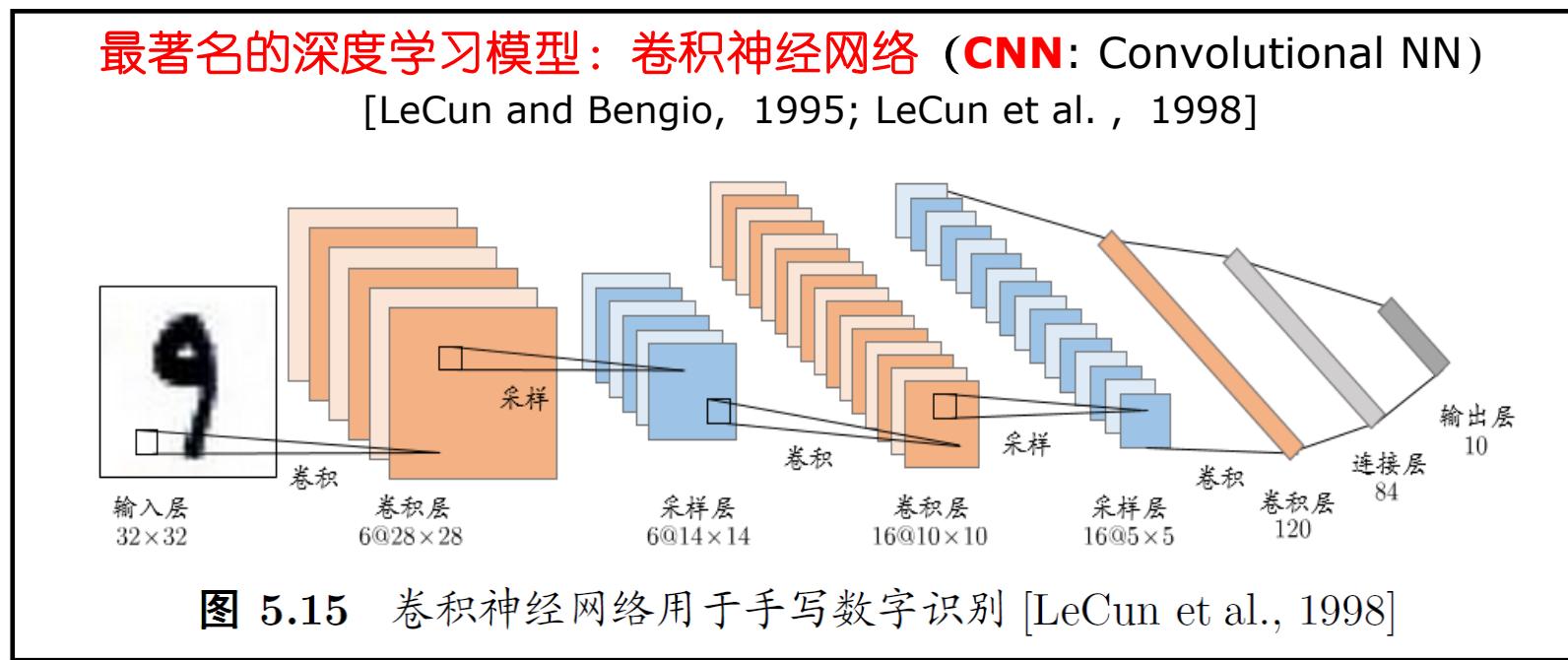
# 大纲

---

- 神经元模型到前馈神经网络
- 参数优化：BP算法
- 深度学习
- 卷积神经网络
- 循环神经网络
- 生成对抗网络

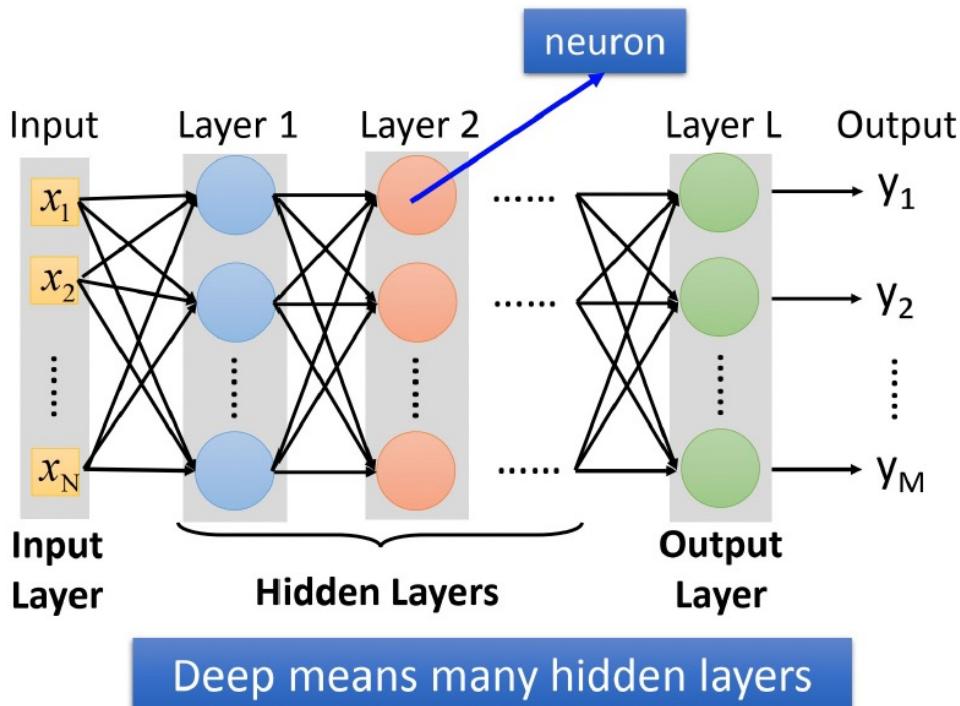
# 深度学习的兴起

- 2006年, Hinton 组发表深度学习的 Science 文章
- 2012年, Hinton 组参加ImageNet 竞赛, 使用 CNN 模型以超过第二名 10个百分点的成绩夺得当年竞赛的冠军
- 在计算机视觉、语音识别、机器翻译等领域取得巨大成功

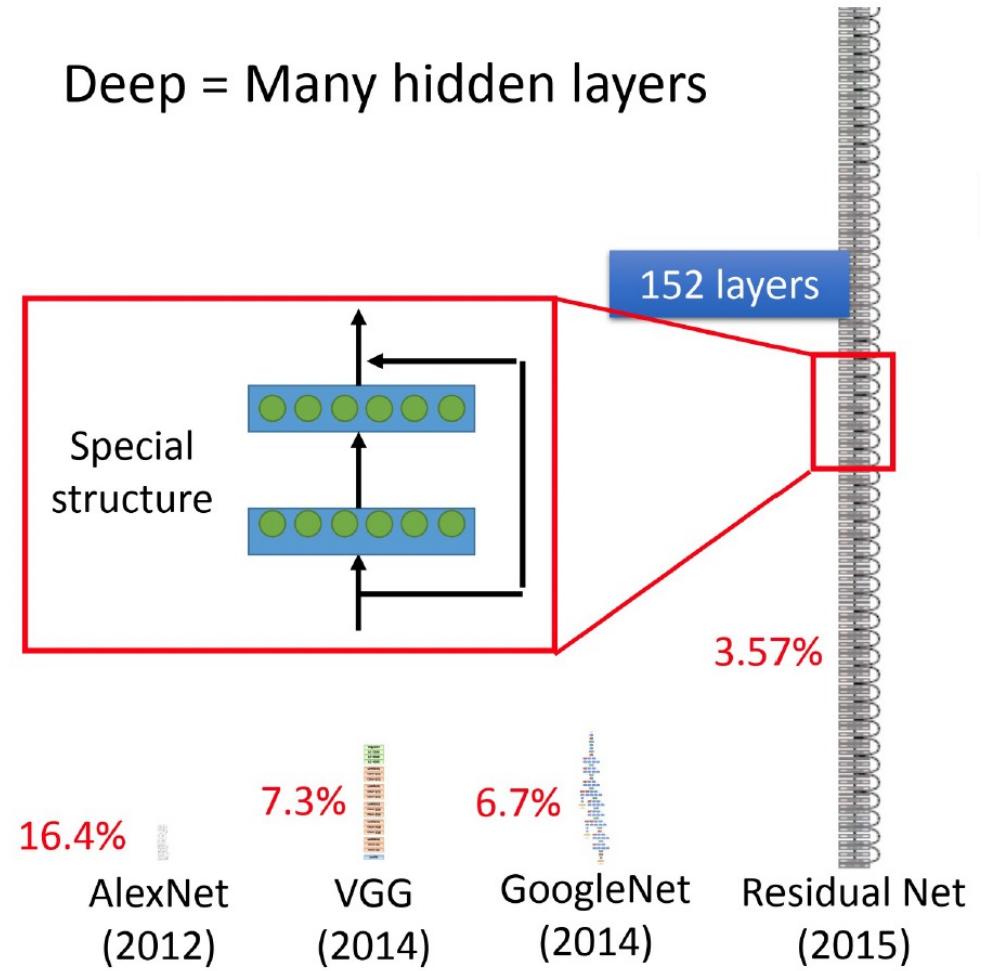


# 深度学习

- 深度学习模型就是具有很多个隐层的神经网络



Deep = Many hidden layers



# 深度学习

多层前馈网络有强大的表示能力（“万有逼近性”）

仅需一个包含足够多神经元的隐层，多层前馈神经网络就能以任意精度逼近任意复杂度的连续函数 [Hornik et al., 1989]

随着云计算的发展和大数据的涌现，一方面，计算能力的大幅提高缓解了训练效率，另一方面，训练数据的大幅增加降低了过拟合风险，因此，以“深度学习”(deep learning)为代表的复杂模型成为了合适的选择

# 深度学习

多层前馈网络有强大的表示能力（“万有逼近性”）

仅需一个包含足够多神经元的隐层，多层前馈神经网络就能以任意精度逼近任意复杂度的连续函数 [Hornik et al., 1989]

- 增加模型复杂程度的方式
  - 模型宽度：增加隐层神经元的数目
  - 模型深度：增加隐层数目

为何是深度学习不是宽度学习？

# 理解深度学习

## 从“特征工程”到“特征学习”或“表示学习”

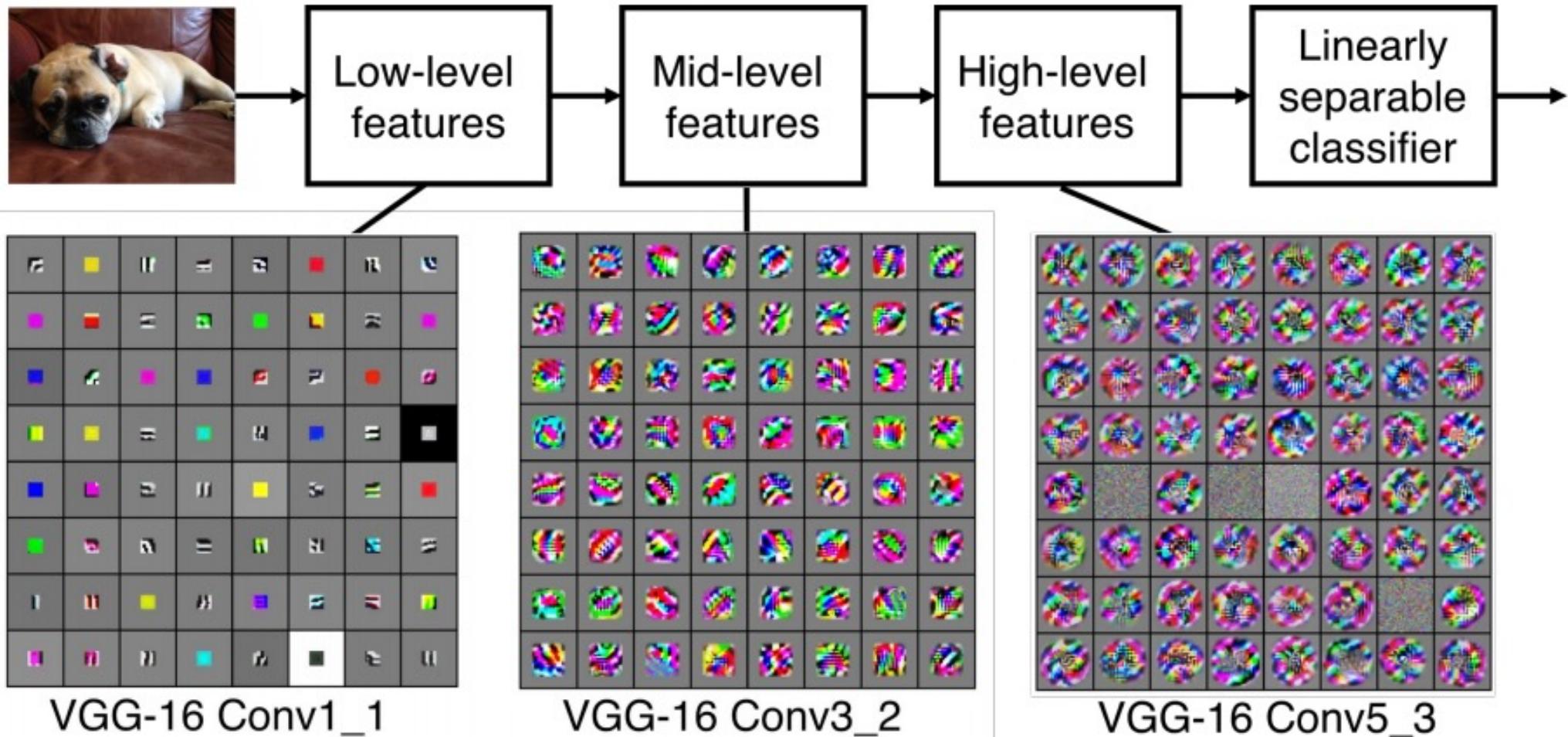
- 特征工程由人类专家根据现实任务来设计，特征提取与识别是分开的两个阶段



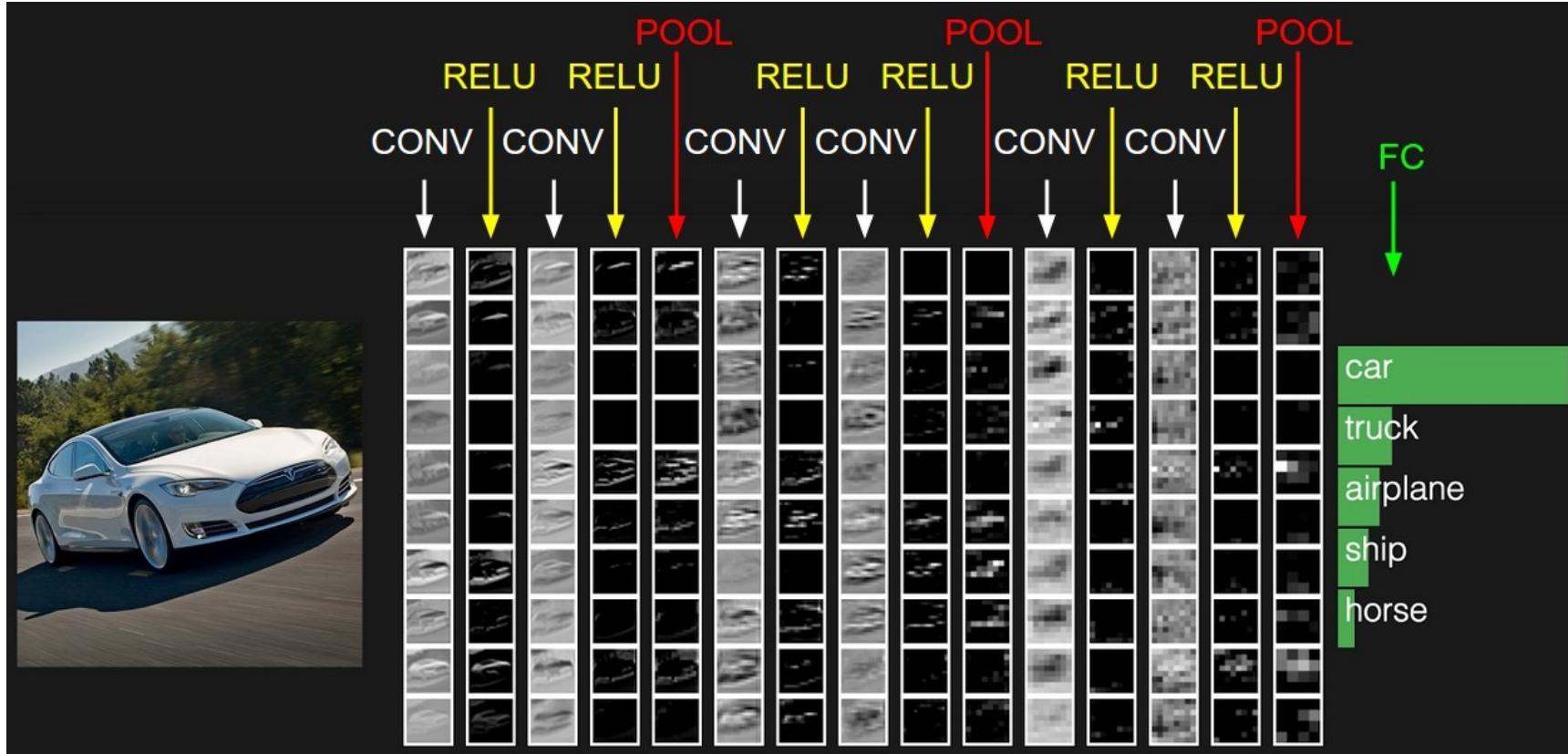
- 特征学习通过深度学习自动产生有益于分类的特征，是一个[端到端](#)的学习框架.



# 表示学习



# 表示学习



# 深度学习

---

- 复杂模型带来的困难
  - 深度网络难以直接用经典算法（例如BP算法）进行训练, 因为误差在多隐层内传播时会出现梯度消失问题（即梯度迅速为0），难以收敛到稳定状态
- 理论难以解决，依赖多种训练技巧(trick)

# 深度学习模型 - 训练技巧

## 预训练+微调

- **预训练**: 或称为监督逐层训练, 每次训练时将上一层隐层结点的输出作为输入, 本层隐结点的输出作为输出, 仅训练一层网络
- **微调**: 预训练全部完成后, 对整个网络进行微调训练, 一般采用BP算法

分析: 预训练+微调的做法可视为将大量参数进行分组, 局部先找到较好的设置, 然后再基于局部较优的结果进行全局寻优

# 深度学习模型 – 训练技巧

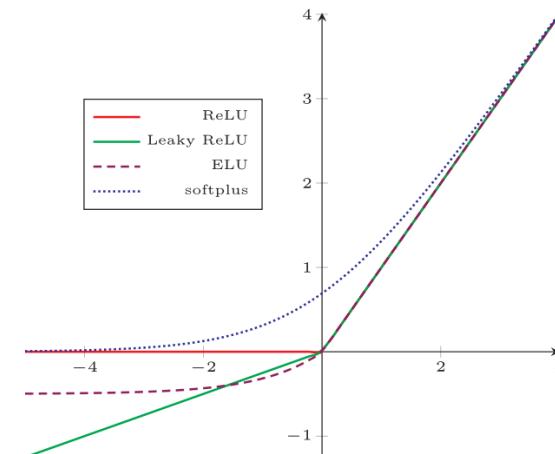
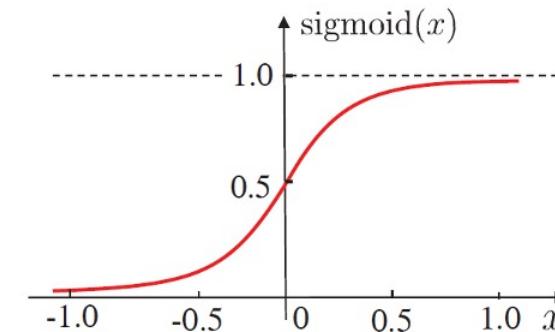
- 新型激活函数

$$\text{ReLU}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} = \max(0, x).$$

$$\text{Leaky ReLU}(x) = \begin{cases} x & \text{if } x > 0 \\ \gamma x & \text{if } x \leq 0 \end{cases} = \max(0, x) + \gamma \min(0, x)$$

$$\text{ELU}(x) = \begin{cases} x & \text{if } x > 0 \\ \gamma(\exp(x) - 1) & \text{if } x \leq 0 \end{cases} = \max(0, x) + \min(0, \gamma(\exp(x) - 1))$$

$$\text{softplus}(x) = \log(1 + \exp(x))$$

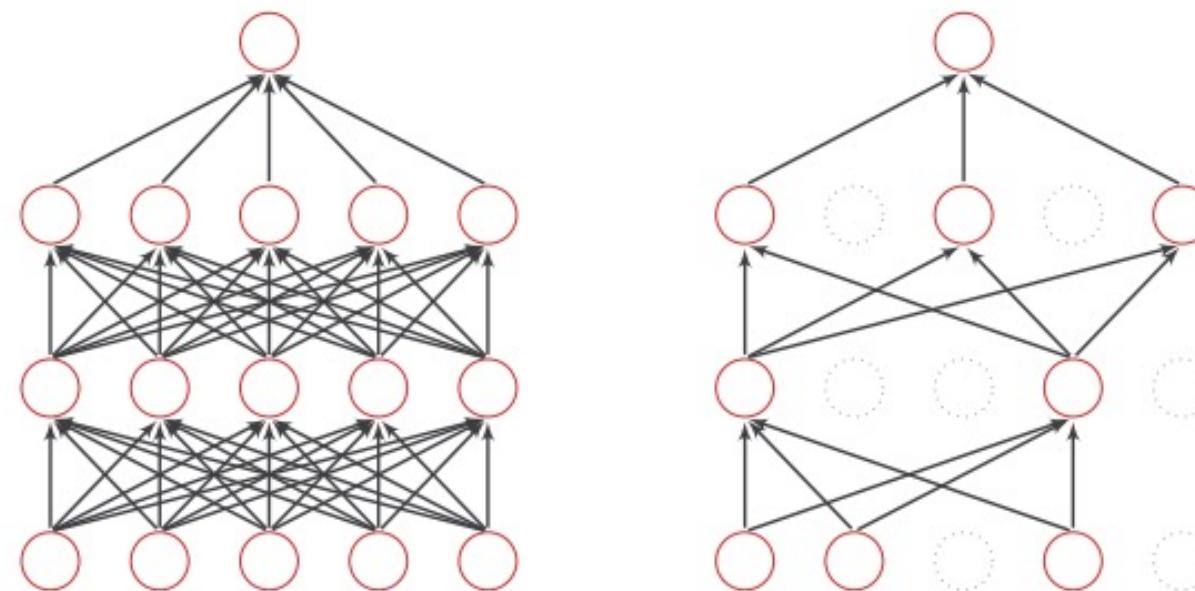


分析：求导更容易，缓解梯度消失

# 深度学习模型 - 训练技巧

- Dropout

当训练一个深度神经网络时，我们可以随机丢弃一部分神经元以及其对应的连接边



分析：降低Rademacher 复杂度

# 深度学习模型 - 训练技巧

- 权共享

- 一组神经元使用相同的连接权值
- 权共享策略在卷积神经网络(CNN)[LeCun and Bengio, 1995; LeCun et al. , 1998]中发挥了重要作用

分析：减少需要优化的参数数量

# 深度学习模型 - 训练技巧

- 交叉熵(cross-entropy)
- BP算法中以交叉熵  $-\frac{1}{m} \sum_{i=1}^m y_i \log \hat{y}_i$  代替均方误差  $\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2$

分析：更能体现分类任务的特性

# 深度学习是“模拟人脑”吗？

## 《IEEE 深度对话 Facebook 人工智能负责人 Yann LeCun》



**Yann LeCun**

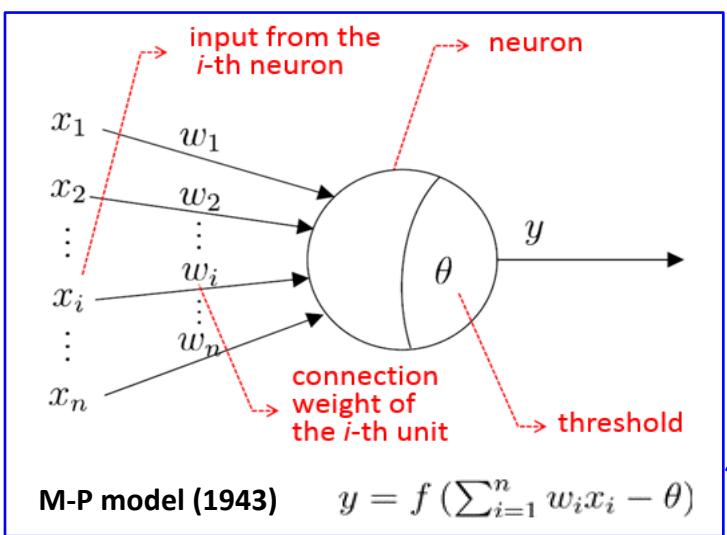
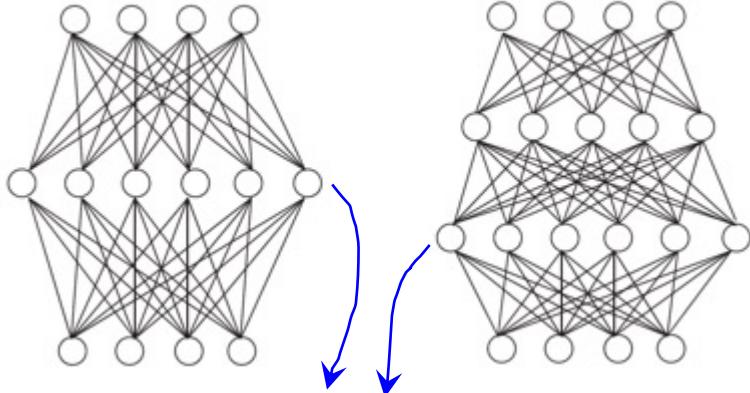
CNN的主要发明人  
深度学习“三架马车”之一  
2019年图灵奖得主

IEEE Spectrum：这些天我们看到了许多关于深度学习的新闻 .....

**Yann LeCun：**我最不喜欢的描述是「它像大脑一样工作」，我不喜欢人们这样说的原因是，虽然深度学习从生命的生物机理中获得灵感，但它与大脑的实际工作原理差别非常非常巨大。将它与大脑进行类比给它赋予了一些神奇的光环，这种描述是危险的。

# 深度神经网络

以往神经网络采用单或双隐层结构



例如, ImageNet 胜者:

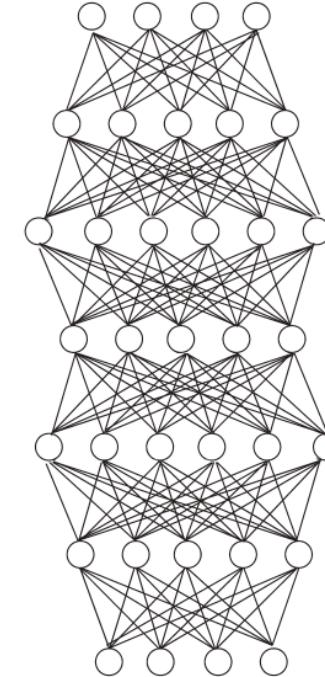
2012: 8 层

2015: 152 层

2016: 1207 层

deep

深度神经网络:  
很多层



神经网络实质上是多层次函数嵌套形成的数学模型

可以说受到了一点生物神经机制的“启发”，但远没有“受指导”

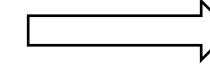
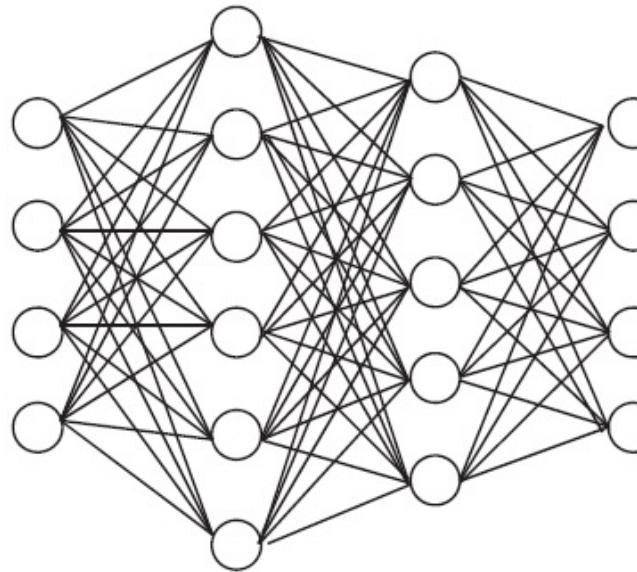
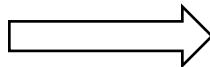
至今最常用的算法: BP [Rumelhart et al., 1986], 是完全从数学上推导出来的

# 大纲

---

- 神经元模型到前馈神经网络
- 参数优化：BP算法
- 深度学习
- 卷积神经网络**
- 循环神经网络
- 生成对抗网络

# 卷积神经网络(Convolutional Neural Network, CNN)



Cat  
Dog

- 使用全连接前馈网络处理图像的问题：
  - **参数太多**：如果输入数据为 $1000 \times 1000$ 的图像，需要转换为 $1,000,000$ 维的向量，如果第一个隐藏层与输入维度一致，则输入层到第一个隐藏层的待训练参数为 $10^{12}$ 个
  - **局部不变性特征**：自然图像中的物体都具有局部不变形特征，比如尺度缩放、平移、旋转不影响语义，前馈网络很难提取局部不变性特征

# 卷积神经网络(Convolutional Neural Network, CNN)

- 卷积神经网络：一般由卷积层、采样层和全连接层构成的前馈神经网络，具有**局部连接、权值共享、汇聚等特性**，使得卷积神经网络具有一定度的平移、缩放和旋转不变性，同时与前馈神经网络相比，参数更少
- 卷积(Convolution)

$$\begin{array}{|c|c|c|} \hline 1 & 1 & \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline \times -1 & \times 0 & \times 0 \\ \hline \end{array} \\ \hline -1 & 0 & \begin{array}{|c|c|c|} \hline -3 & 0 & 1 \\ \hline \times 0 & \times 0 & \times 0 \\ \hline \end{array} \\ \hline 2 & 1 & \begin{array}{|c|c|c|} \hline 1 & -1 & 0 \\ \hline \times 0 & \times 0 & \times 1 \\ \hline \end{array} \\ \hline 0 & -1 & 1 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \quad * \quad \begin{array}{|c|c|c|} \hline 1 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & -1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 0 & -2 & -1 \\ \hline 2 & 2 & 4 \\ \hline -1 & 0 & 0 \\ \hline \end{array}$$

# 卷积神经网络(Convolutional Neural Network, CNN)

- **卷积层**: 每个卷积层包含多个特征映射, 每个特征映射通过一种卷积滤波器提取一种数据的特征 (**特征提取**)

1 x1	1 x0	1 x1	0	0
0 x0	1 x1	1 x0	1	0
0 x1	0 x0	1 x1	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved  
Feature

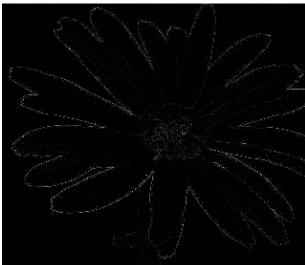
# 卷积神经网络(Convolutional Neural Network, CNN)

- **卷积层**: 每个卷积层包含多个特征映射, 每个特征映射通过一种卷积滤波器提取一种数据的特征 (**特征提取**)


$$\begin{array}{|c|c|c|} \hline \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \hline \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \hline \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \hline \end{array} =$$

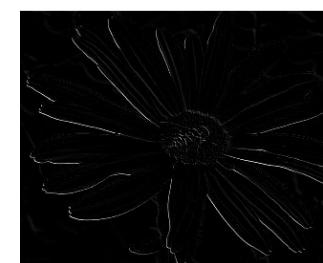


原始图像

$$* \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} =$$


$$\begin{array}{|c|c|c|} \hline 0 & 1 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & -1 & 0 \\ \hline \end{array}$$

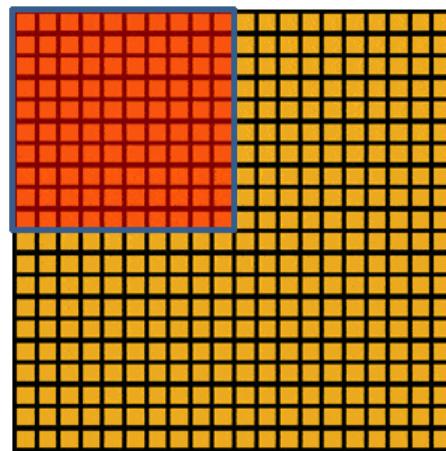
滤波器



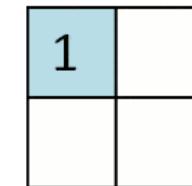
输出特征映射

# 卷积神经网络(Convolutional Neural Network, CNN)

- 采样层：亦称“池化层”，其作用是基于局部相关性原理进行采样，从而在减少数据量的同时保留有用信息（降低参数量级）

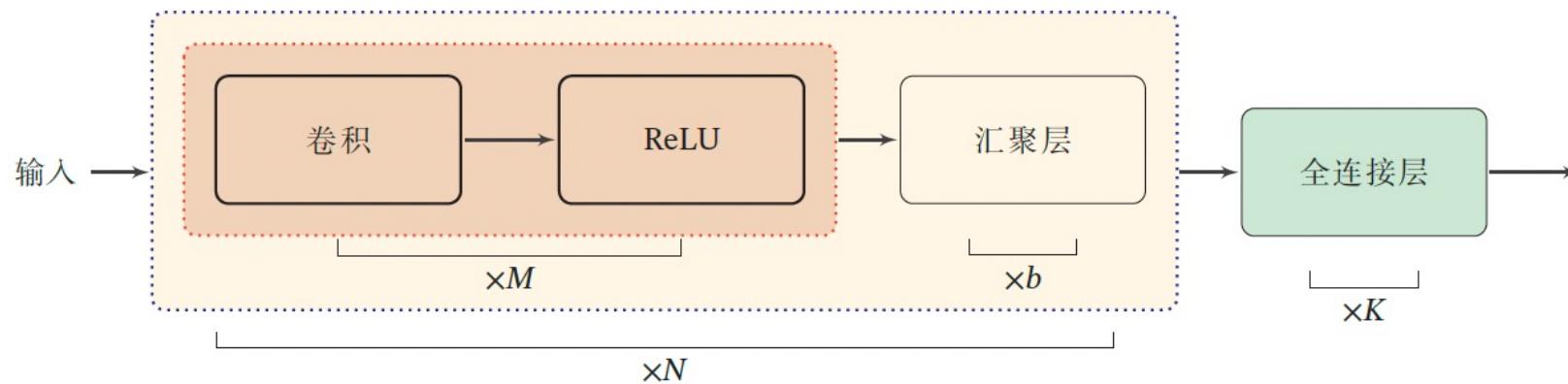


Convolved  
feature



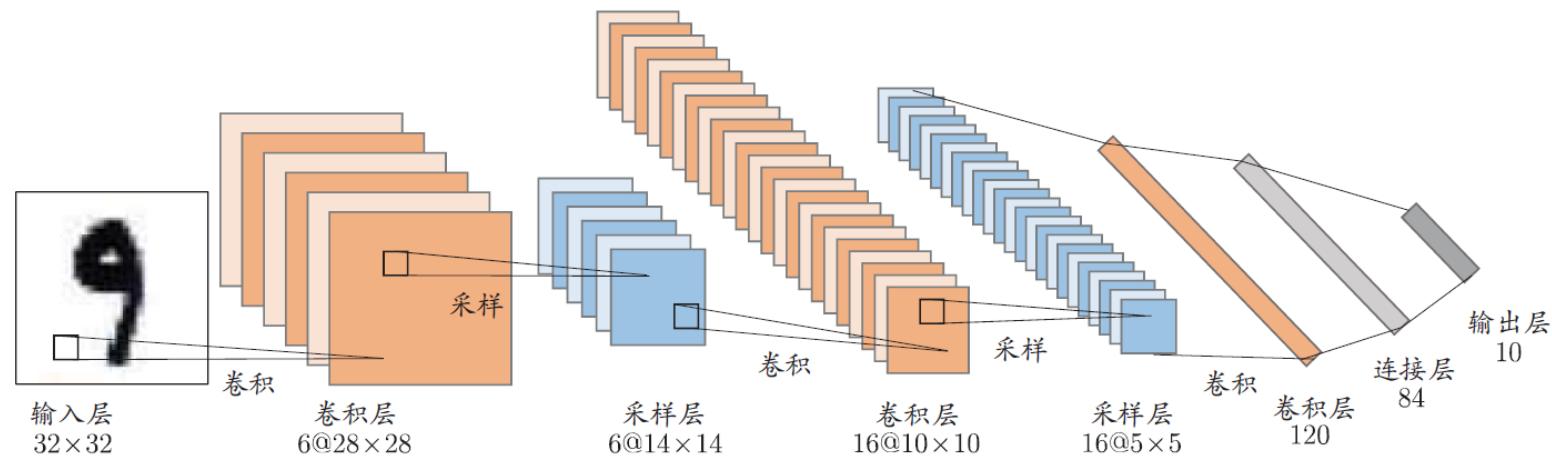
Pooled  
feature

# 卷积神经网络(Convolutional Neural Network, CNN)



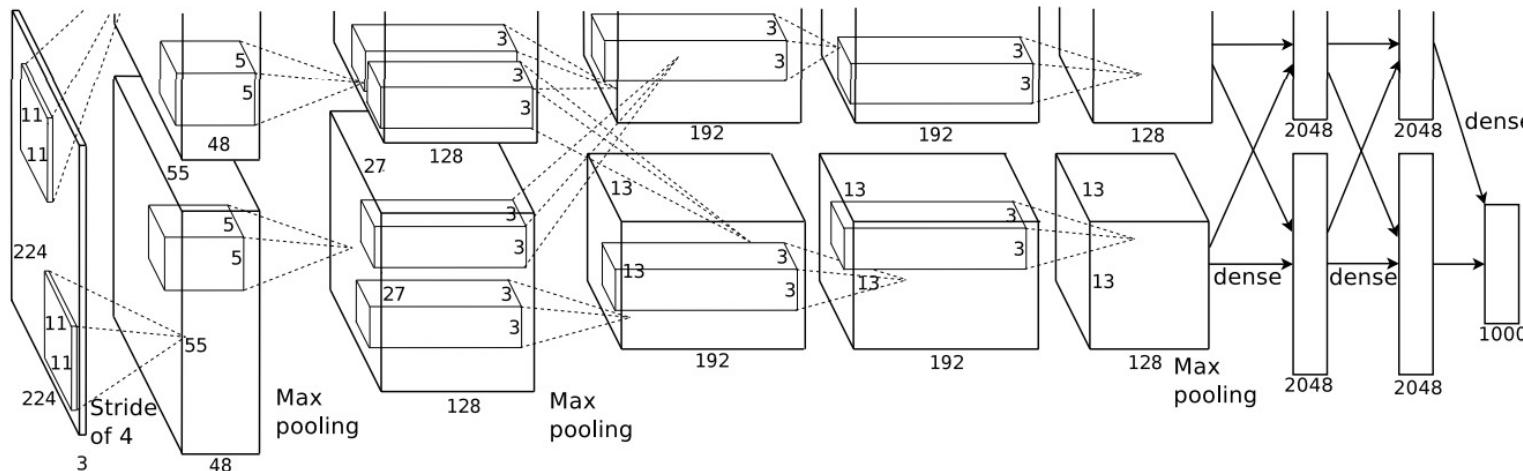
# 卷积神经网络(Convolutional Neural Network, CNN)

- LeNet 是一个非常成功的神经网络模型
  - 基于 LeNet 的手写数字识别系统在 90 年代被美国很多银行使用，用来识别支票上面的手写数字



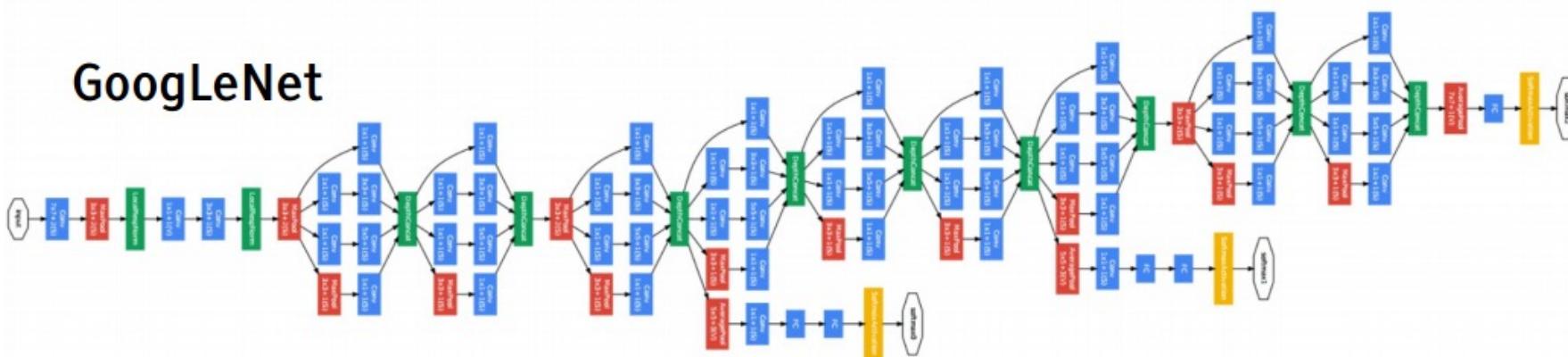
# 卷积神经网络-AlexNet

- 2012 ILSVRC winner
  - (top 5 error of 16% compared to runner-up with 26% error)
  - 第一个现代深度卷积网络模型
    - 首次使用了很多现代深度卷积网络的一些技术方法
      - 使用GPU进行并行训练，采用了ReLU作为非线性激活函数，使用Dropout防止过拟合，使用数据增强



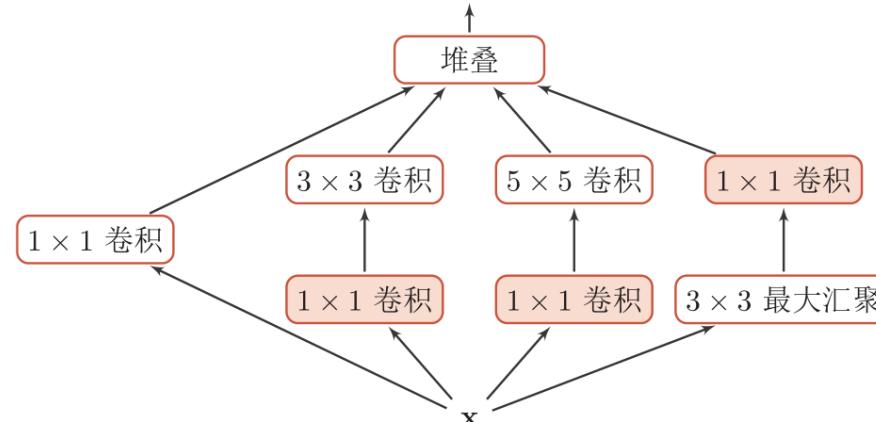
# 卷积神经网络-InceptionNet

- 2014 ILSVRC winner (22层)
  - 参数：GoogLeNet：4M VS AlexNet：60M
  - 错误率：6.7%
  - Inception网络是由多个inception模块和少量的汇聚层堆叠而成



# 卷积神经网络-InceptionNet

- 在卷积网络中，如何设置卷积层的卷积核大小是一个十分关键的问题
  - 在Inception网络中，一个卷积层包含多个不同大小的卷积操作，称为Inception模块
  - Inception模块同时使用 $1 \times 1$ 、 $3 \times 3$ 、 $5 \times 5$ 等不同大小的卷积核，并将得到的特征映射在深度上拼接（堆叠）起来作为输出特征映射

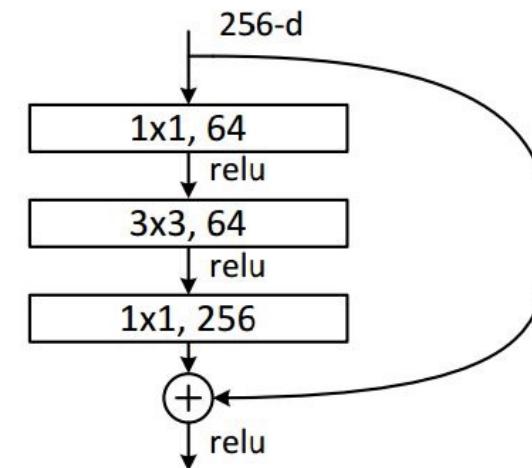
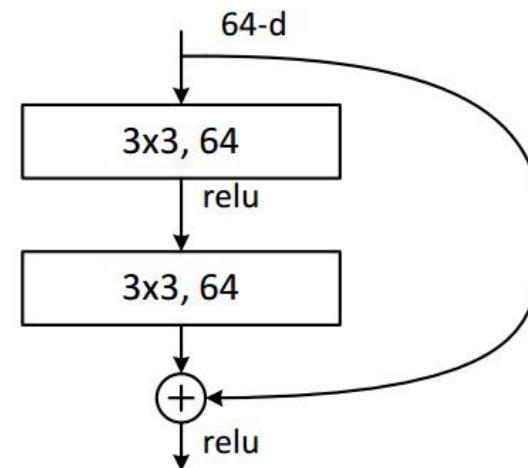
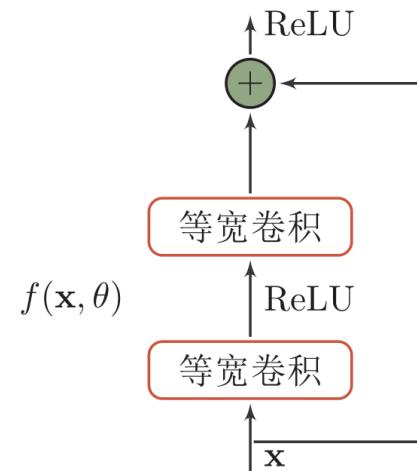


# 卷积神经网络-ResNet

- 残差网络（Residual Network, ResNet）是通过给非线性的卷积层增加直连边的方式来提高信息的传播效率
  - 假设在一个深度网络中，我们期望一个非线性单元（可以为一层或多层的卷积层） $f(x, \theta)$ 去逼近一个目标函数为 $h(x)$
  - 将目标函数拆分成两部分：恒等函数和残差函数

$$h(\mathbf{x}) = \underbrace{\mathbf{x}}_{\text{恒等函数}} + \underbrace{(h(\mathbf{x}) - \mathbf{x})}_{\text{残差函数}} \xrightarrow{} f(\mathbf{x}, \theta)$$

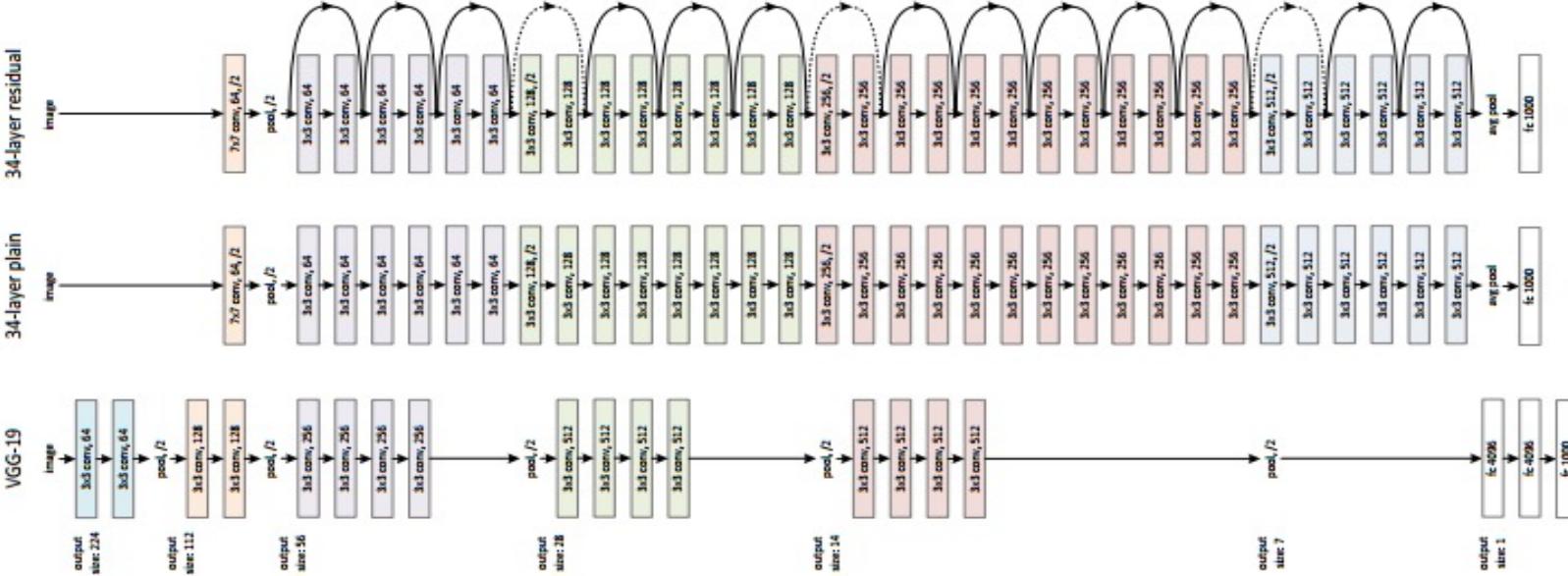
# 残差单元



# 卷积神经网络-ResNet

- 2015 ILSVRC winner (152层)

- 错误率： 3.57%



# 大纲

---

- 神经元模型到前馈神经网络
- 参数优化：BP算法
- 深度学习
- 卷积神经网络
- 循环神经网络**
- 生成对抗网络

# 循环神经网络(Recurrent Neural Network, RNN)

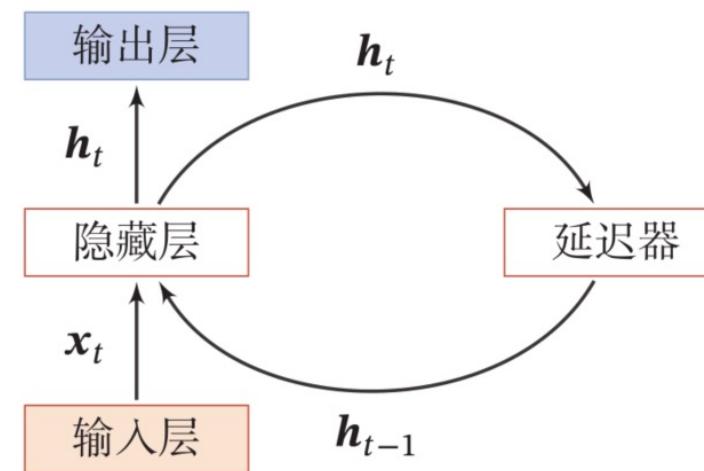
---

- 现实很多应用面临序列数据
  - 文本分析
  - 语音识别
  - 机器翻译
  - 视频动作识别
  - ...

# 循环神经网络(Recurrent Neural Network, RNN)

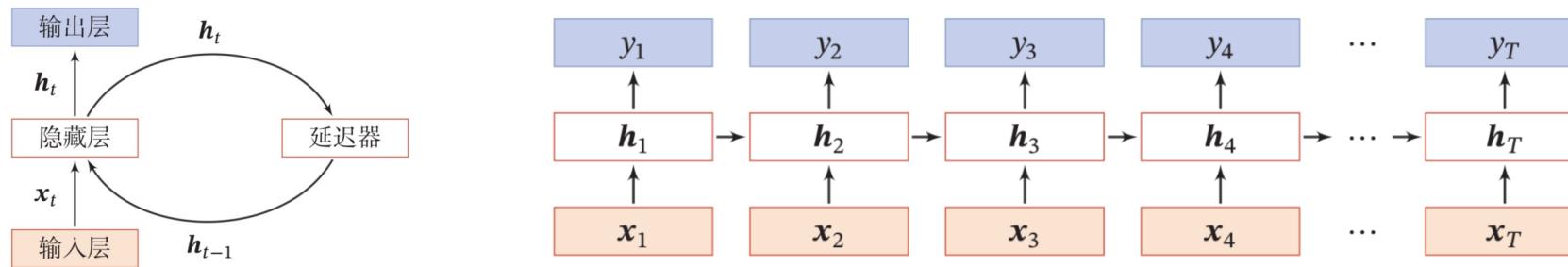
- 循环神经网络通过使用带自反馈的神经元，能够处理任意长度的时序数据

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, \mathbf{x}_t)$$



# 循环神经网络(Recurrent Neural Network, RNN)

- 循环神经网络通过使用带自反馈的神经元，能够处理任意长度的时序数据



- 循环神经网络已经被广泛应用在语音识别、语言模型以及自然语言生成等任务上

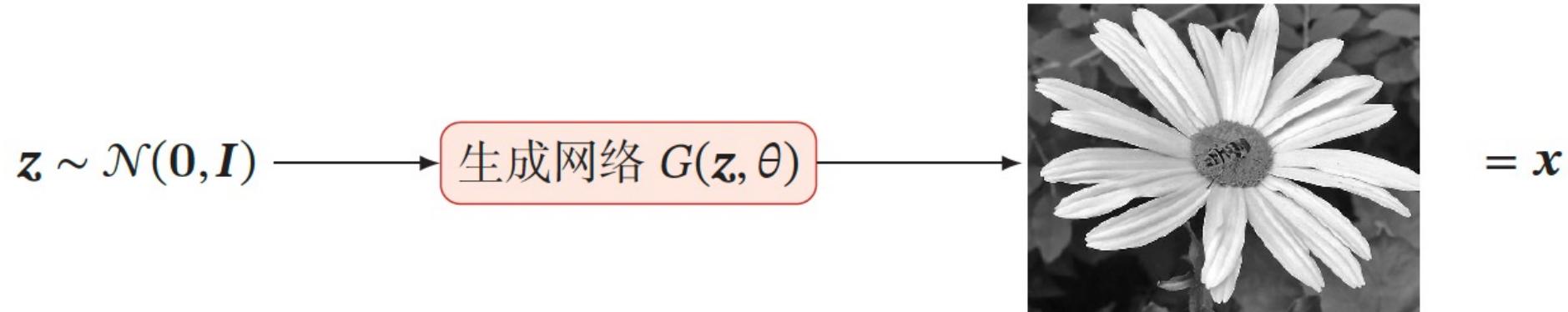
# 大纲

---

- 神经元模型到前馈神经网络
- 参数优化：BP算法
- 深度学习
- 卷积神经网络
- 循环神经网络
- 生成对抗网络

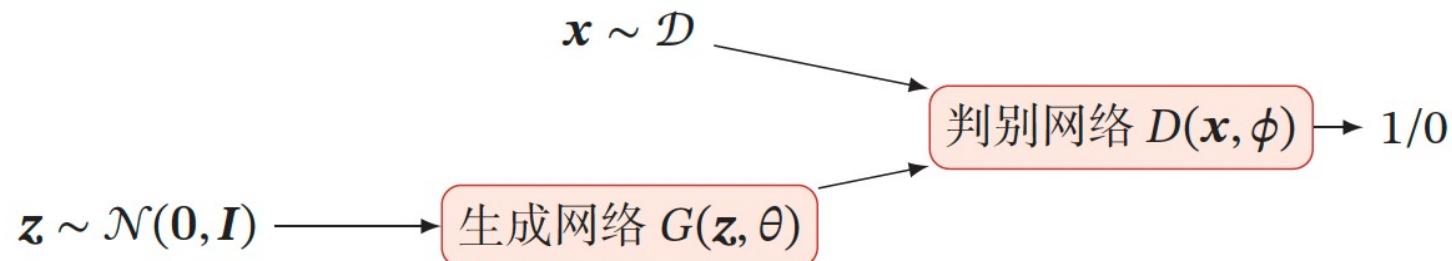
# 生成对抗网络(Generative Adversarial Networks, GAN)

是否可以构建出一个模型  $G: \mathcal{Z} \rightarrow \mathcal{X}$ , 使之可以生成符合数据分布  $p(x)$  的样本 ?



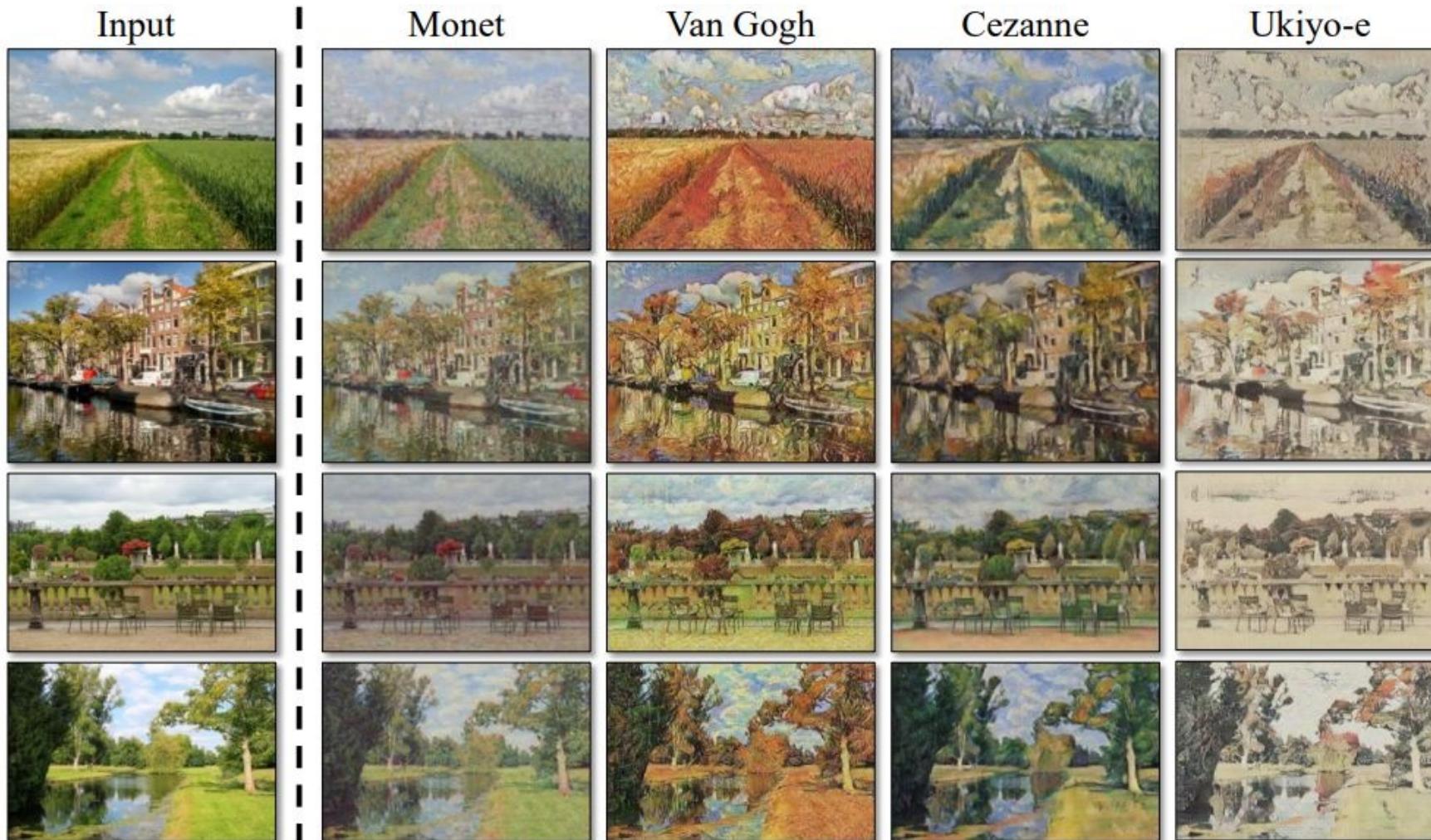
# 生成对抗网络(Generative Adversarial Networks, GAN)

- 生成对抗网络[Goodfellow et al., 2014]通过对抗训练的方式使得生成网络产生的样本服从真实数据分布
- 判别网络(Discriminator): 目标是尽量准确地判断一个样本是来自于真实数据还是由生成网络产生
- 生成网络(Geneartor): 目标是尽量生成判别网络无法区分来源的样本



# 生成对抗网络的典型应用

- 图像到图像(Image-to-Image)



# 生成对抗网络的典型应用

- 文本到图像(Text-to-Image)

this small bird has a pink breast and crown, and black primaries and secondaries.



the flower has petals that are bright pinkish purple with white stigma



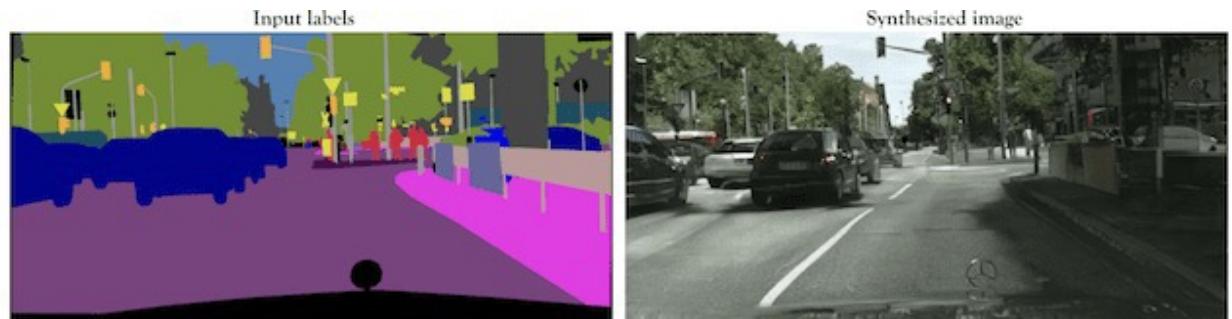
this magnificent fellow is almost all black with a red crest, and white cheek patch.



this white and yellow flower have thin white petals and a round yellow stamen

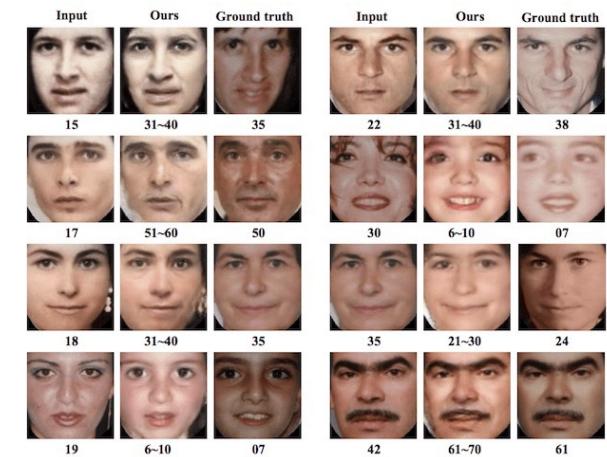
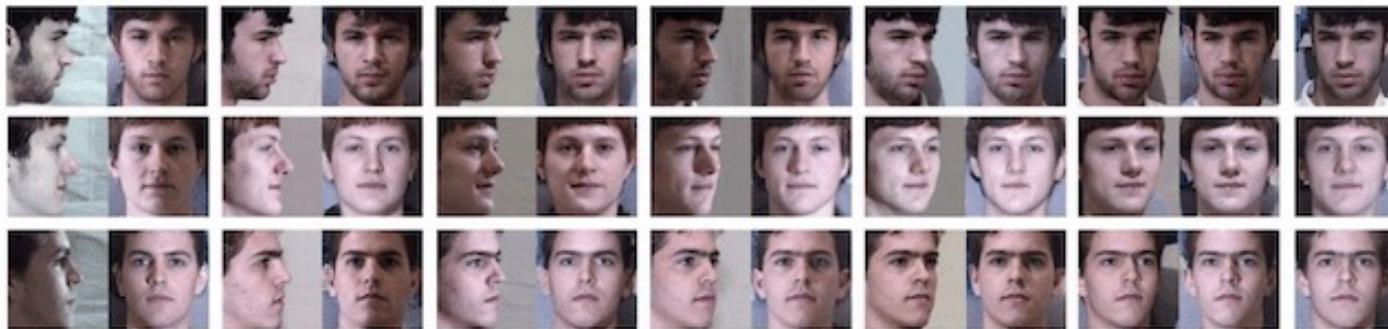


- 语义图到真实图像(Semantic Image-to-Photo)



# 生成对抗网络的典型应用

- 人脸生成(Face Frontal View Generation)



- 姿态生成(Human Pose Generation)



# 小结

---

- 神经元模型：熟悉
- 感知机：了解单层感知机不能解决异或问题
- 前馈神经网络：熟悉
- BP算法：熟悉推导过程
- 卷积神经网络：了解
- 循环神经网络：了解
- 生成对抗网络：了解