

高级机器学习

从神经网络到深度学习



大纲

- 神经元模型
 - 感知机与多层网络
 - 误差逆传播算法
 - 全局最小与局部极小
 - 深度学习
-

神经网络

“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”

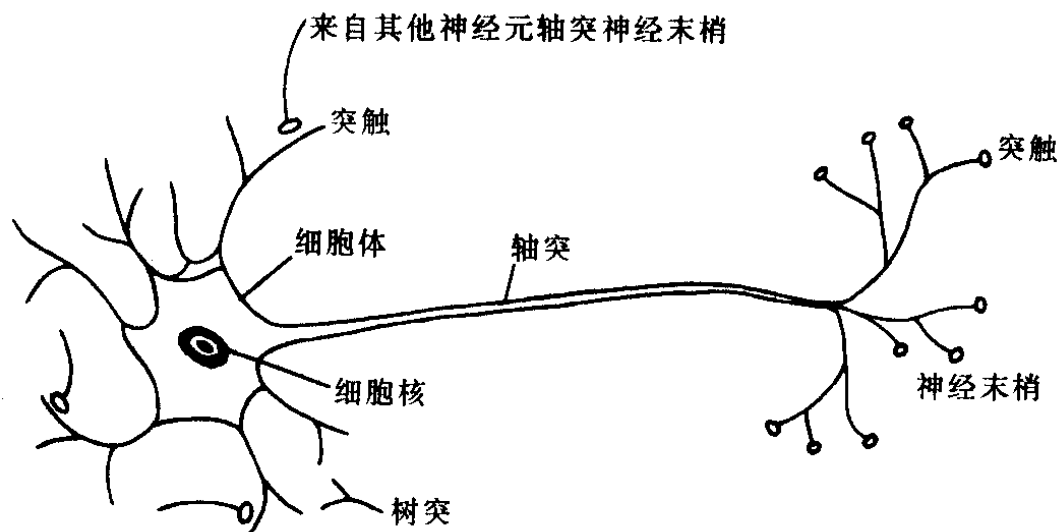
[T. Kohonen, 1988, *Neural Networks* 创刊号]

神经网络是一个很大的学科领域，神经网络与机器学习的交集，称为“神经网络学习”

亦称“连接主义(connectionism)” 学习

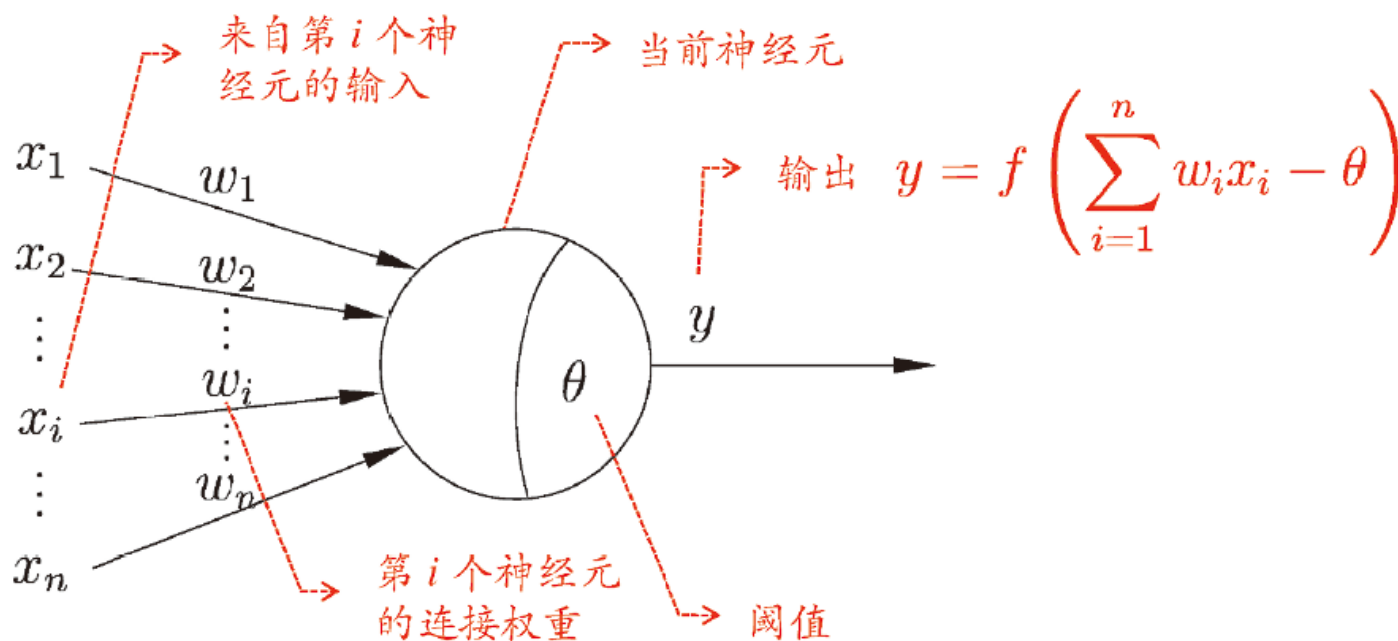
神经元模型

- 神经网络始于神经元模型，神经元模型是神经网络的基本成分
- 生物神经网络：每个神经元与其他神经元相连，当它“兴奋”时，就会向相连的神经云发送化学物质，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，即“兴奋”起来，向其它神经元发送化学物质



神经元模型

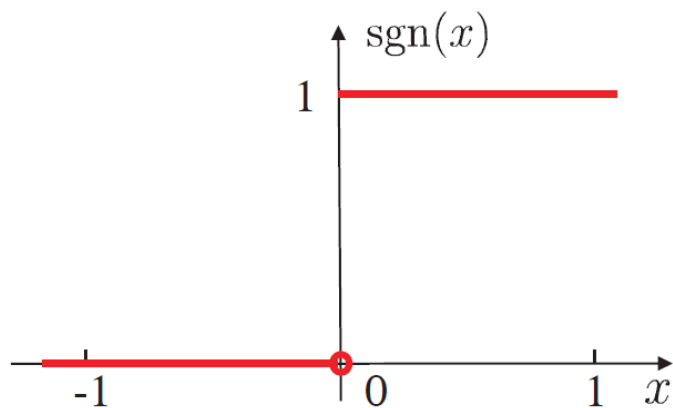
M-P 神经元模型 [McCulloch and Pitts, 1943]



神经网络学得的知识蕴含在连接权与阈值中

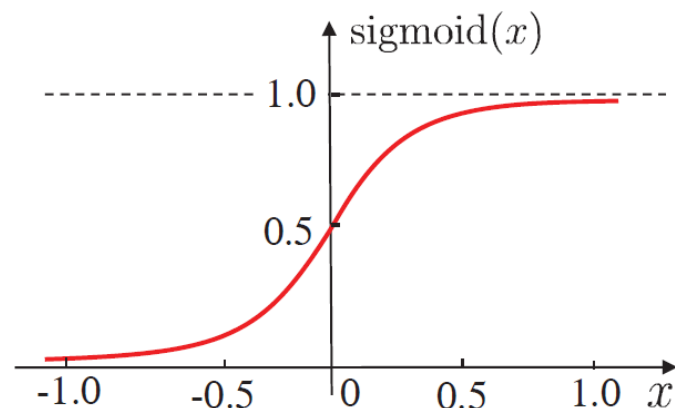
神经元的激活函数

- 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

图 5.2 典型的神经元激活函数

大纲

- 神经元模型
 - 感知机与多层网络
 - 误差逆传播算法
 - 全局最小与局部极小
 - 深度学习
-

感知机与多层神经网络

- 感知机由两层神经元组成，输入层接受外界输入信号传递给输出层，输出层是M-P神经元（阈值逻辑单元）

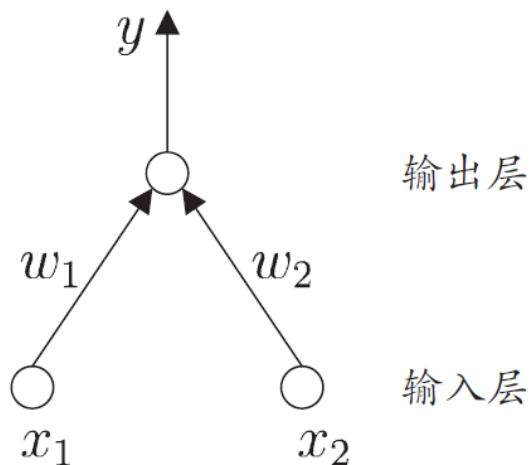


图 5.3 两个输入神经元的感知机网络结构示意图

感知机与多层神经网络

- 感知机能够容易实现逻辑与、或、非运算
 - “与” ($x_1 \wedge x_2$): 令 $w_1 = w_2 = 1$, $\theta = 2$, 则 $y = f(1 \cdot x_1 + 1 \cdot x_2 - 2)$, 仅在 $x_1 = x_2 = 1$ 时, $y = 1$;
 - “或” ($x_1 \vee x_2$): 令 $w_1 = w_2 = 1$, $\theta = 0.5$, 则 $y = f(1 \cdot x_1 + 1 \cdot x_2 - 0.5)$, 当 $x_1 = 1$ 或 $x_2 = 1$ 时, $y = 1$;
 - “非” ($\neg x_1$): 令 $w_1 = -0.6$, $w_2 = 0$, $\theta = -0.5$, 则 $y = f(-0.6 \cdot x_1 + 0 \cdot x_2 + 0.5)$, 当 $x_1 = 1$ 时, $y = 0$; 当 $x_1 = 0$ 时, $y = 1$.
-

感知机与多层神经网络

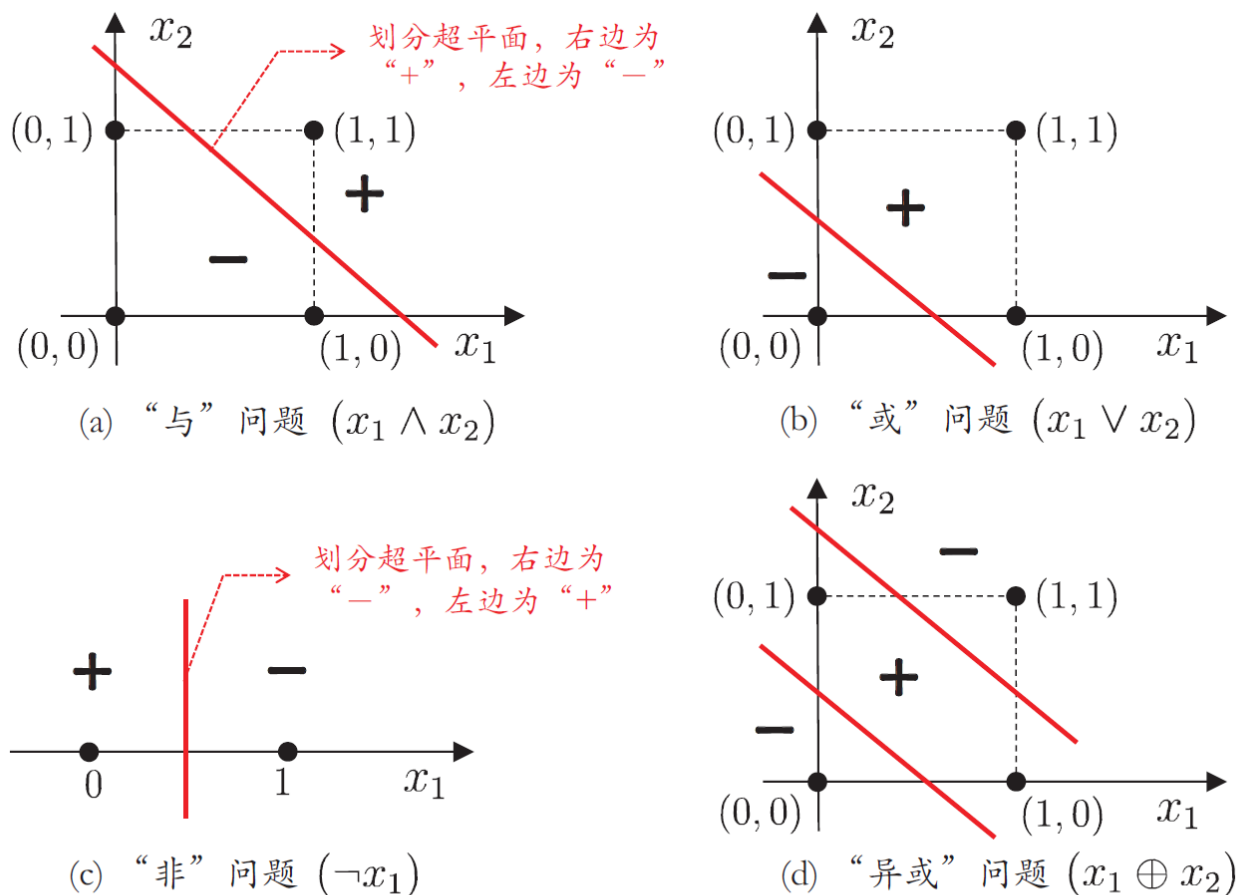


图 5.4 线性可分的“与”“或”“非”问题与非线性可分的“异或”问题

感知机与多层神经网络

- 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]
- 单层感知机的学习能力有限，只能解决线性可分问题
 - 例如，与、或、非问题线性可分，感知机能够求得适当的权值向量
 - 对于异或等不能够线性可分的问题，感知机不能求得合适解
 - 如何利用感知机处理非线性可分问题？

多层感知机

感知机与多层神经网络

- 构建两层感知机，求解异或问题

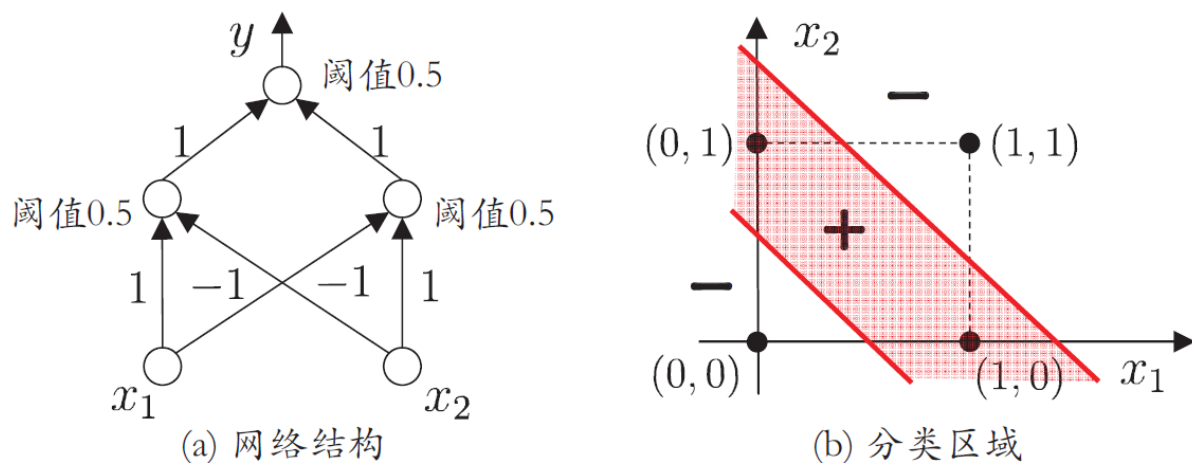


图 5.5 能解决异或问题的两层感知机

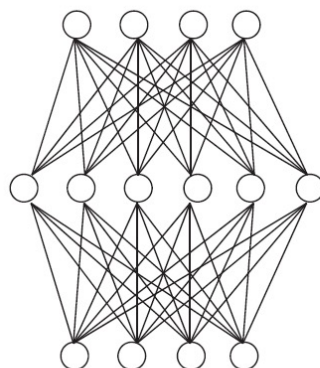
- 我们将输出层与输入层之间的一层神经元，称为**隐层或隐含层**
- 隐含层和输出层神经元都是具有激活函数的功能神经元

多层前馈神经网络

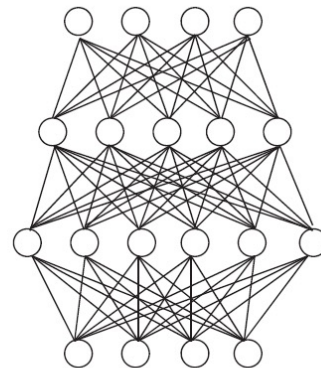
多层网络：包含隐层的网络

前馈网络：神经元之间不存在同层连接也不存在跨层连接

隐层和输出层神经元亦称
“功能单元” (functional unit)



(a) 单隐层前馈网络



(b) 双隐层前馈网络

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]

但是, 如何设置隐层神经元数是未决问题(Open Problem). 实际常用“试错法”

大纲

- 神经元模型
 - 感知机与多层网络
 - 误差逆传播算法
 - 全局最小与局部极小
 - 深度学习
-

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：

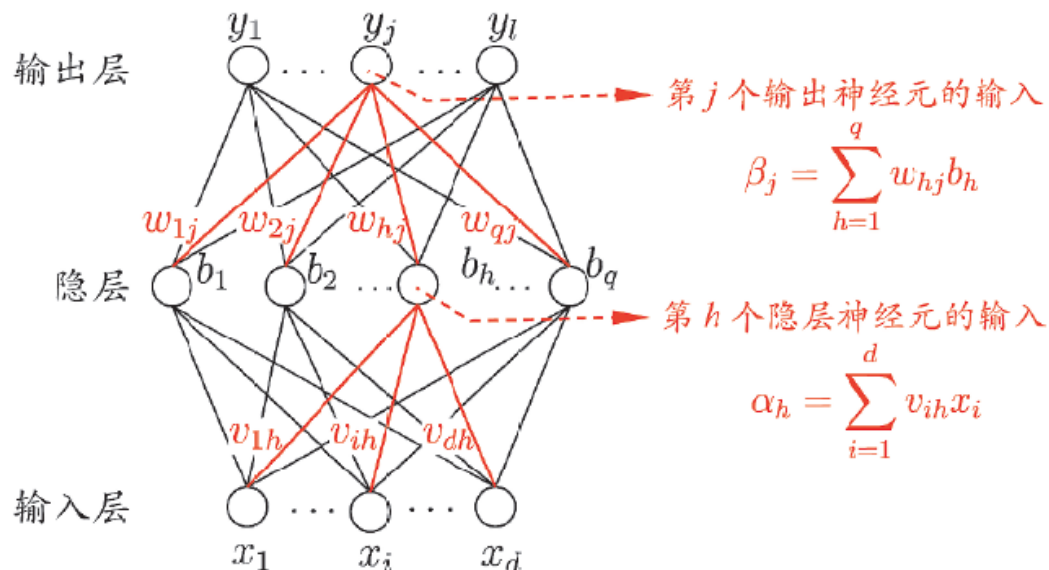
P. Werbos. Beyond regression: New tools for prediction and analysis in the behavioral science. Ph.D dissertation, Harvard University, 1974

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： l 个输出值

隐层：假定使用 q 个
隐层神经元

假定功能单元均使用
Sigmoid 函数



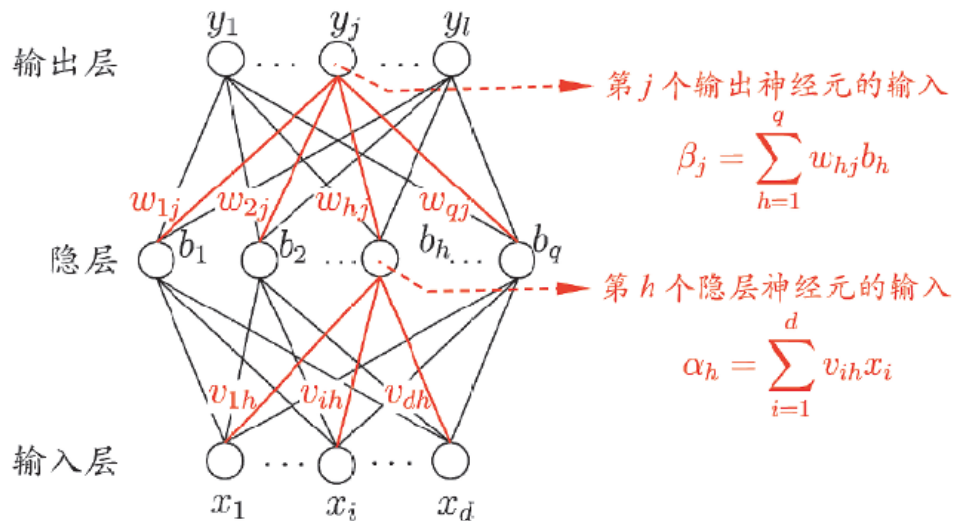
BP (BackPropagation:误差逆传播算法)

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

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

则网络在 $(\mathbf{x}_k, \mathbf{y}_k)$ 上的均方误差为:

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



需通过学习确定的参数数目: $(d + l + 1)q + l$

BP 是一个迭代学习算法, 在迭代的每一轮中采用广义感知机学习规则

$$v \leftarrow v + \boxed{\Delta v}.$$

BP (BackPropagation:误差逆传播算法)

BP 算法基于**梯度下降**策略，以目标的负梯度方向对参数进行调整

以 w_{hj} 为例

对误差 E_k ，给定学习率 η ，有：

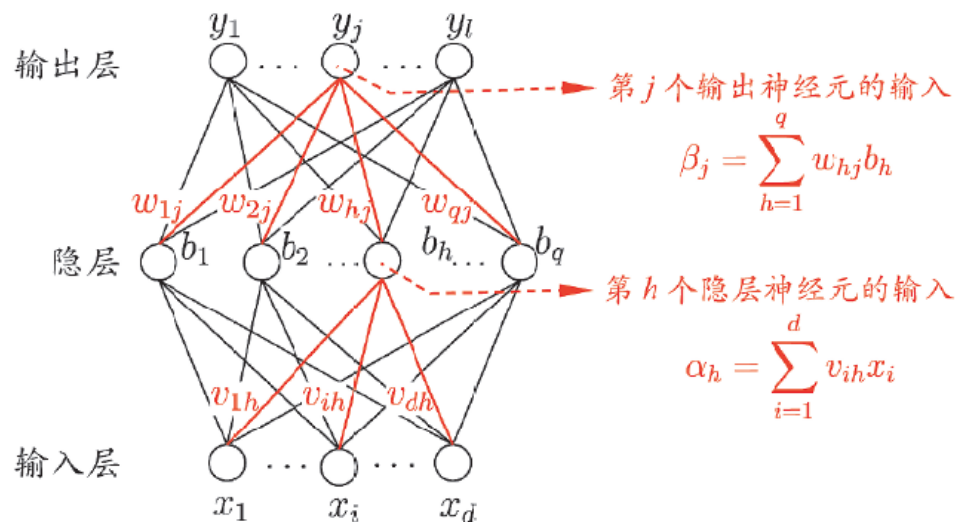
$$\Delta w_{hj} = -\eta \frac{\partial E_k}{\partial w_{hj}}$$

注意到 w_{hj} 先影响到 β_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)

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

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

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

$f'(\beta_j - \theta_j)$

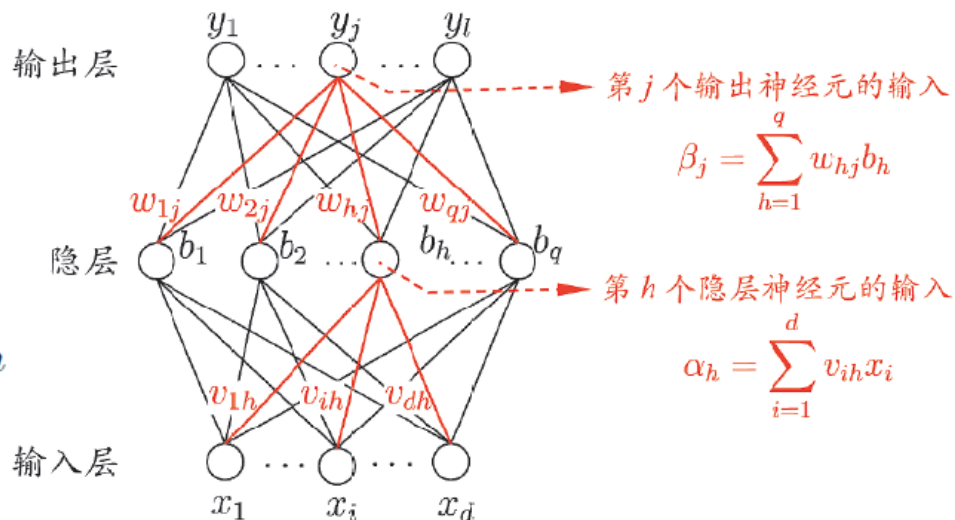
$\hat{y}_j^k (1 - \hat{y}_j^k)$

令 $g_j = -\frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j}$

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

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

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



$$\beta_j = \sum_{h=1}^q w_{hj} b_h$$

$$\alpha_h = \sum_{i=1}^d v_{ih} x_i$$

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：

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

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

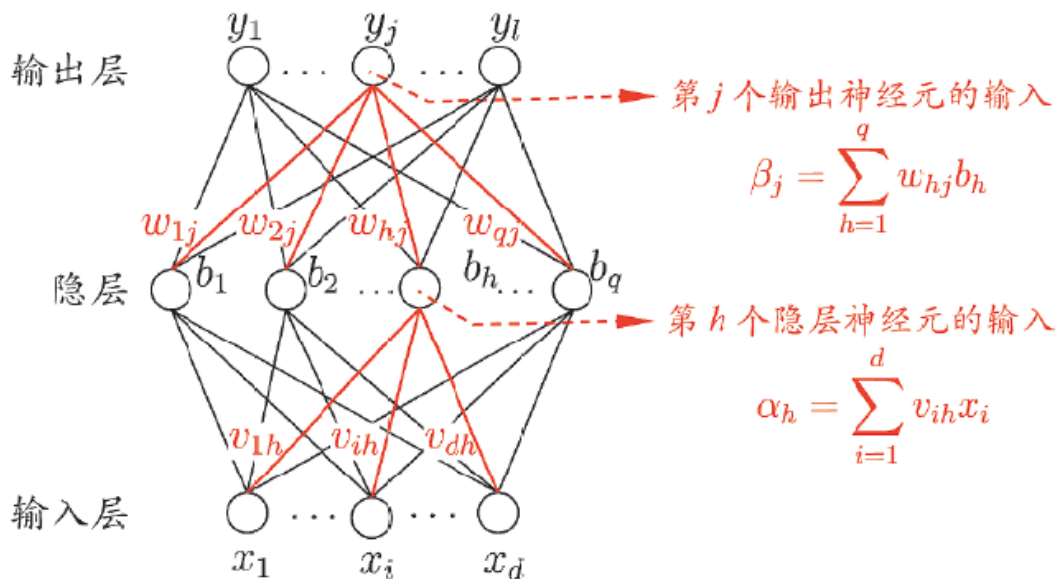
$$\Delta\gamma_h = -\eta e_h$$

其中：

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

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

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



学习率 $\eta \in (0, 1)$ 不能太大、不能太小

BP (BackPropagation:误差逆传播算法)

输入: 训练集 $D = \{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^m$;
学习率 η .

过程:

- 1: 在(0, 1)范围内随机初始化网络中所有连接权和阈值
- 2: **repeat**
- 3: **for all** $(\mathbf{x}_k, \mathbf{y}_k) \in D$ **do**
- 4: 根据当前参数和式(5.3) 计算当前样本的输出 $\hat{\mathbf{y}}_k$;
- 5: 根据式(5.10) 计算输出层神经元的梯度项 g_j ;
- 6: 根据式(5.15) 计算隐层神经元的梯度项 e_h ;
- 7: 根据式(5.11)-(5.14) 更新连接权 w_{hj} , v_{ih} 与阈值 θ_j , γ_h
- 8: **end for**
- 9: **until** 达到停止条件

输出: 连接权与阈值确定的多层前馈神经网络

图 5.8 误差逆传播算法

标准 BP 算法 vs. 累积 BP 算法

标准 BP 算法

- 每次针对单个训练样例更新权值与阈值
- 参数更新频繁, 不同样例可能抵消, 需要多次迭代

累积 BP 算法

- 其优化目标是最小化整个训练集上的累计误差
- 读取整个训练集一遍才对参数进行更新, 参数更新频率较低

在很多任务中, 累计误差下降到一定程度后, 进一步下降会非常缓慢, 这时标准BP算法往往会获得较好的解, 尤其当训练集非常大时效果更明显。

缓解过拟合

主要策略：

□ 早停(early stopping)

- 若训练误差连续 a 轮的变化小于 b , 则停止训练
- 使用验证集：若训练误差降低、验证误差升高, 则停止训练

□ 正则化 (regularization)

- 在误差目标函数中增加一项描述网络复杂度

例如
$$E = \lambda \frac{1}{m} \sum_{k=1}^m E_k + (1 - \lambda) \sum_i w_i^2$$

偏好比较小的连接权和阈值,
使网络输出更“光滑”

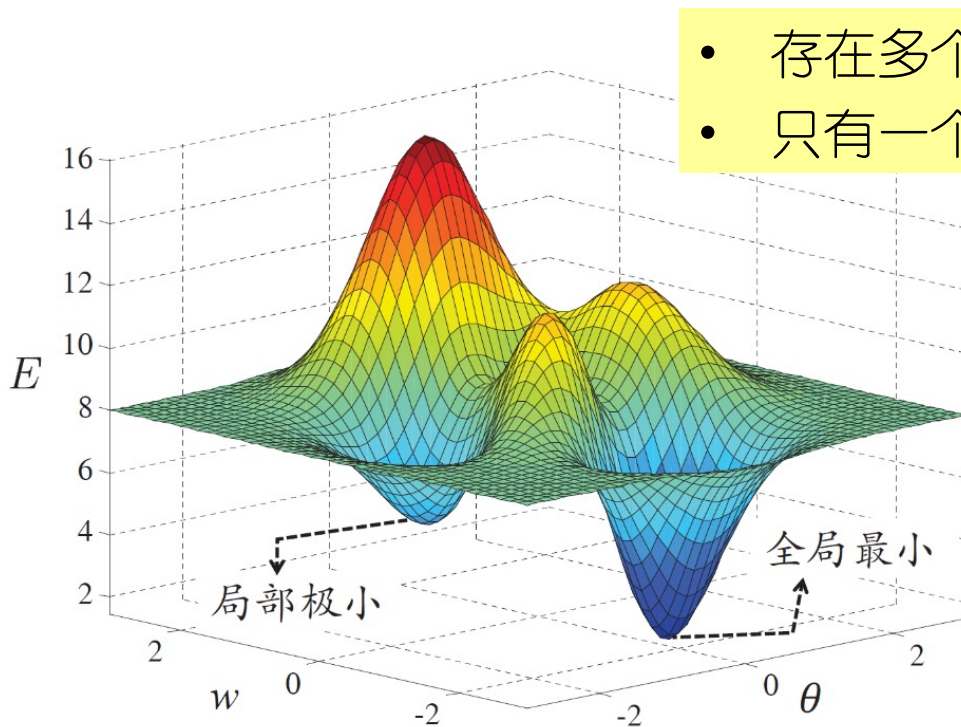
大纲

- 神经元模型
 - 感知机与多层网络
 - 误差逆传播算法
 - 全局最小与局部极小
 - 深度学习
-

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
- ✓ 模拟退火
- ✓ 随机扰动
- ✓ 演化算法
- ✓

大纲

- 神经元模型
 - 感知机与多层网络
 - 误差逆传播算法
 - 全局最小与局部极小
 - 深度学习
-

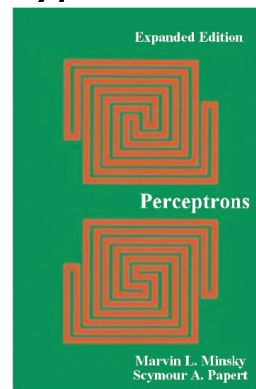
神经网络发展回顾

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"

冰河期



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

沉寂期

2012-至今~繁荣期：深度学习

交替模式：
热十三？年
冷十五？年

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

最著名的深度学习模型：卷积神经网络 (CNN: Convolutional NN)

[LeCun and Bengio, 1995; LeCun et al., 1998]

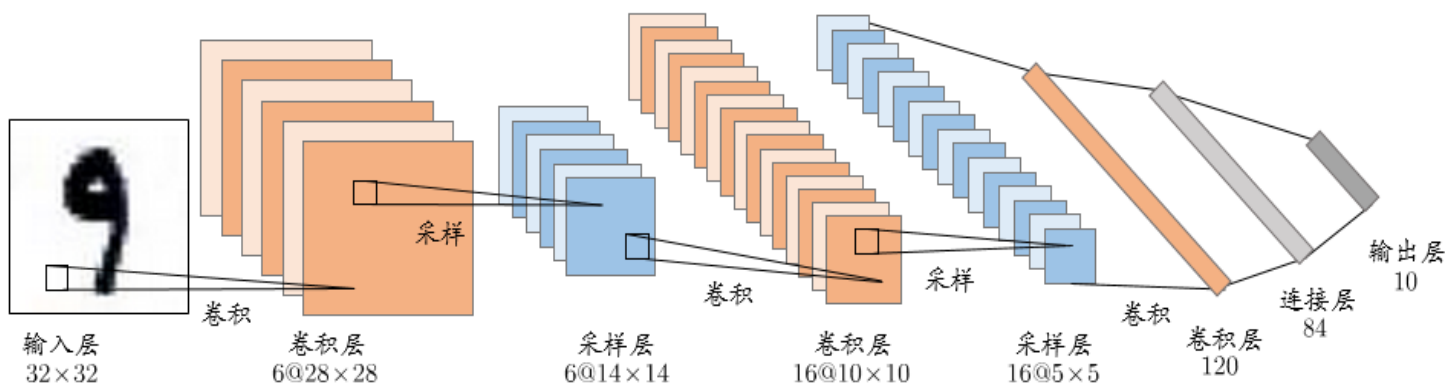


图 5.15 卷积神经网络用于手写数字识别 [LeCun et al., 1998]

深度学习模型

- 深度学习模型是具有很多个隐层的神经网络.
 - 【发展的需求】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
 - 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
 - 实际应用中，增加模型深度比增加宽度相对更有效
 - 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.
-

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。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。

例子：深度信念网络[Hinton et al., Nature 2006]，每层是个受限 Boltzmann机，采用训练方法为无监督预训练 + BP微调

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

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

$$= \max(0, x).$$

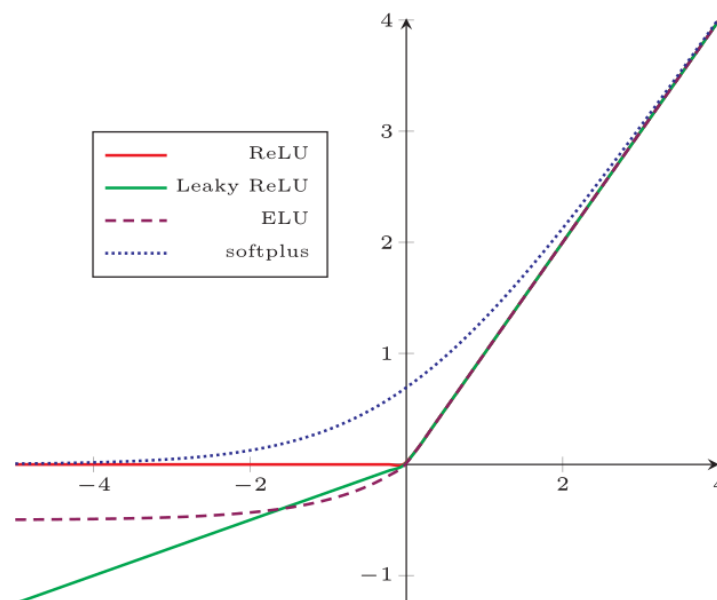
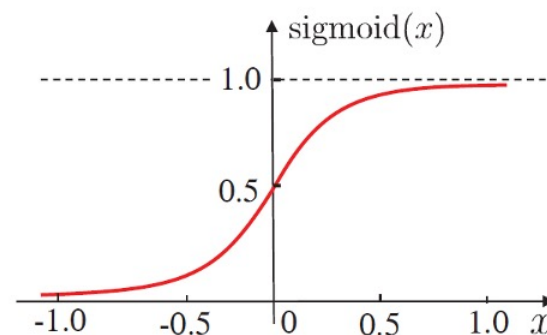
$$\text{LeakyReLU}(x) = \begin{cases} x & \text{if } x > 0 \\ \gamma x & \text{if } x \leq 0 \end{cases}$$

$$= \max(0, x) + \gamma \min(0, x)$$

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

$$= \max(0, x) + \min(0, \gamma(\exp(x) - 1))$$

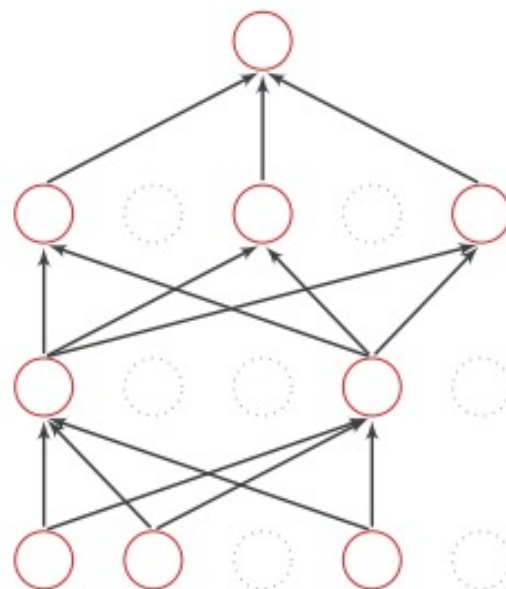
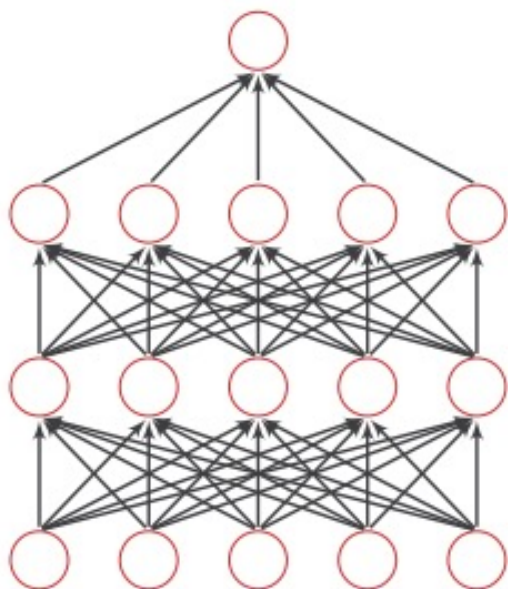
$$\text{softplus}(x) = \log(1 + \exp(x))$$



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

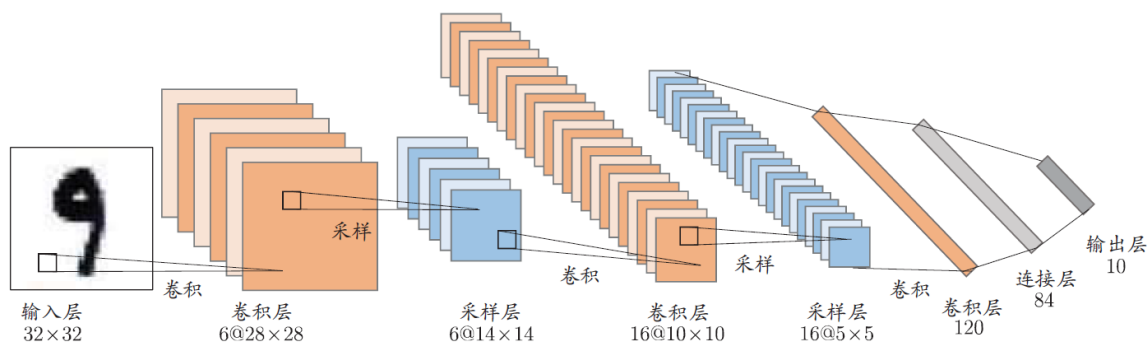


深度学习模型 – 训练技巧

- 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.

- 卷积神经网络

结构：CNN复合多个卷积层和采样层对输入信号进行加工，然后在连接层实现与输出目标之间的映射。



卷积神经网络

- **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）
 - **采样层**：亦称“池化层”，其作用是基于局部相关性原理进行亚采样，从而在减少数据量的同时保留有用信息（降低参数量级）
 - **连接层**：每个神经元被全连接到上一层每个神经元，本质就是传统的神经网络，其目的是通过连接层和输出层的连接完成识别任务
-

卷积神经网络

- **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

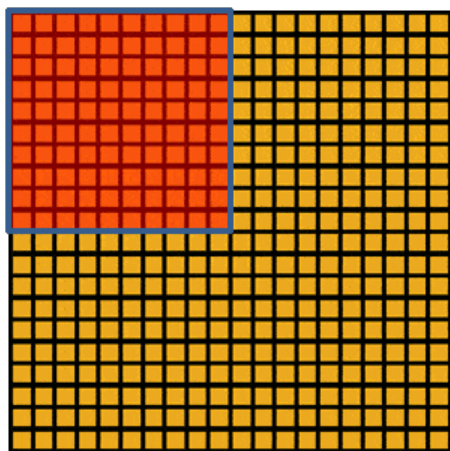
Image

4		

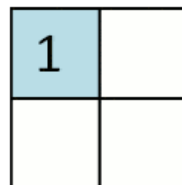
Convolved
Feature

卷积神经网络

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



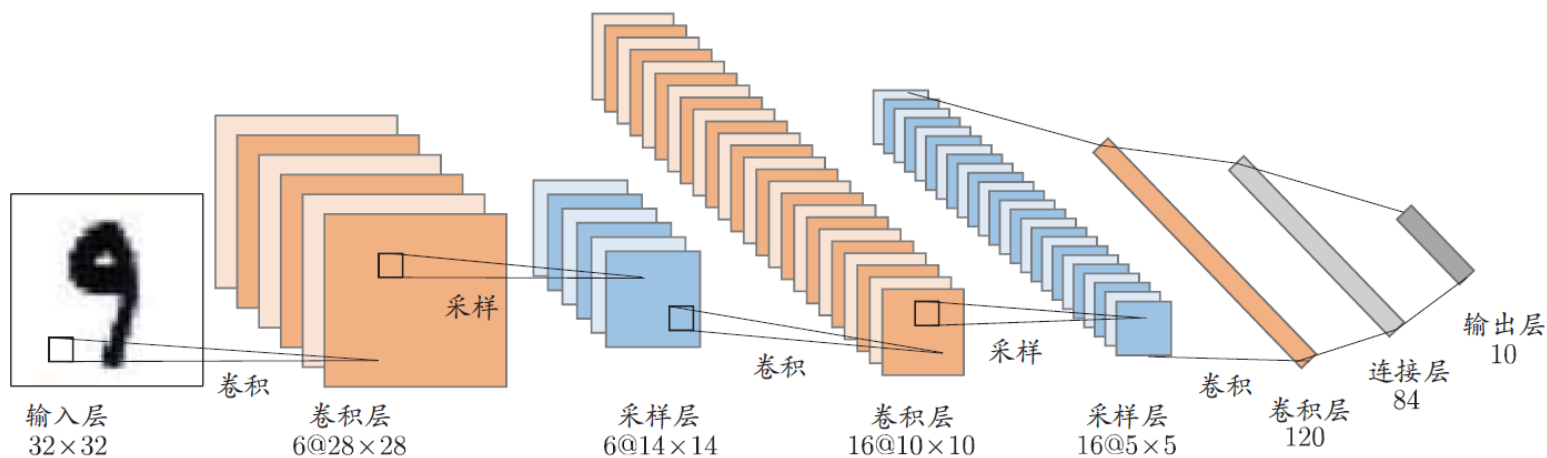
Convolved
feature



Pooled
feature

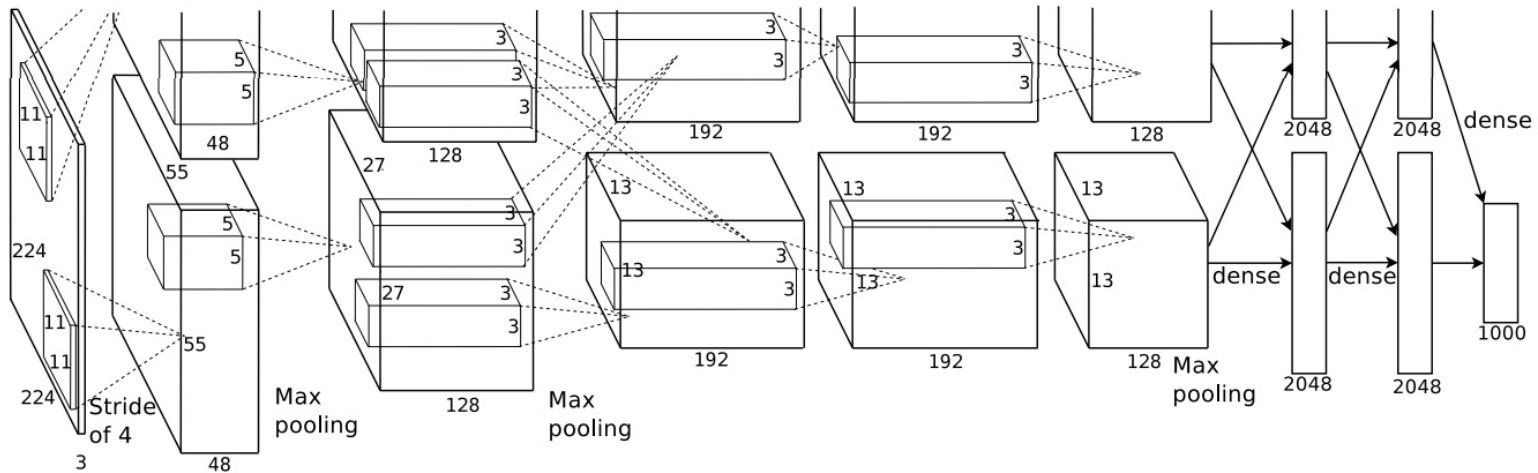
卷积神经网络-LeNet

- 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。



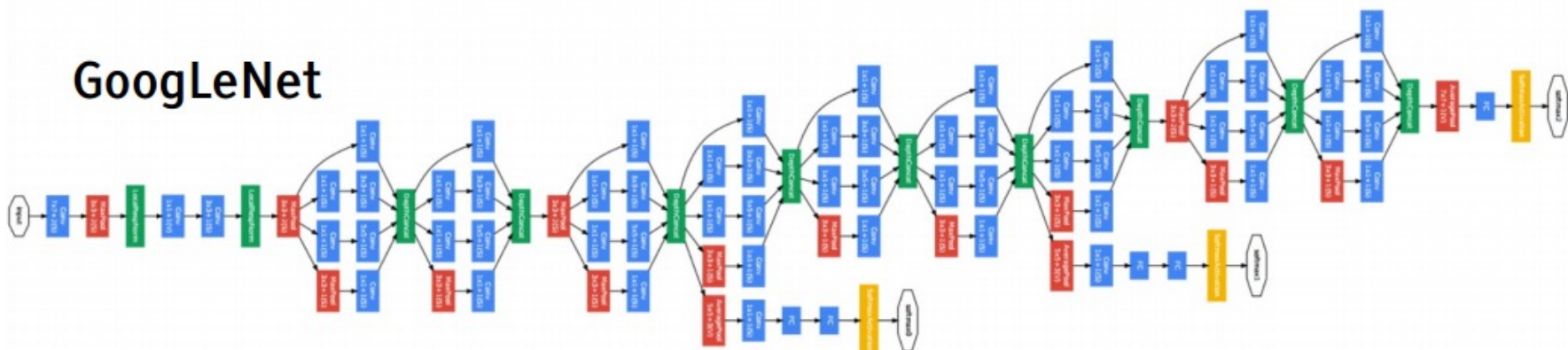
卷积神经网络-AlexNet

- **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



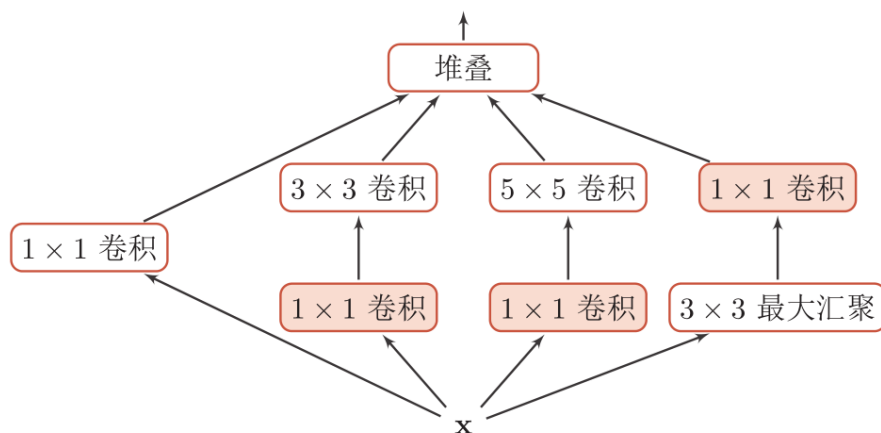
卷积神经网络-InceptionNet

- 2014 ILSVRC winner (22层)
- 参数：GoogLeNet：4M VS AlexNet：60M
- 错误率：6.7%
- Inception网络是由有多个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 1×1 、 3×3 、 5×5 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。

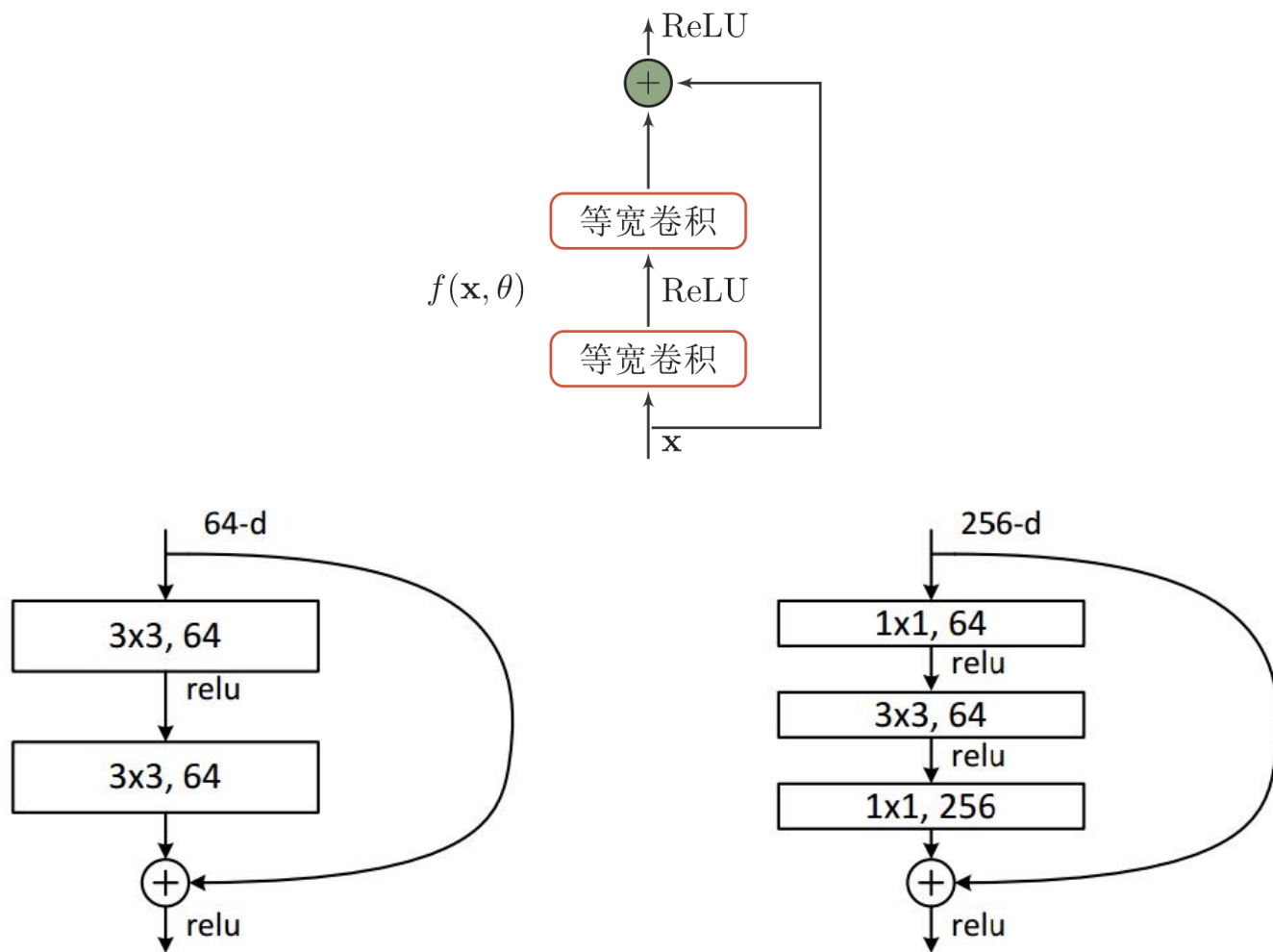


卷积神经网络-ResNet

- 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） $f(\mathbf{x}, \theta)$ 去逼近一个目标函数为 $h(\mathbf{x})$ 。
- 将目标函数拆分成两部分：恒等函数和残差函数

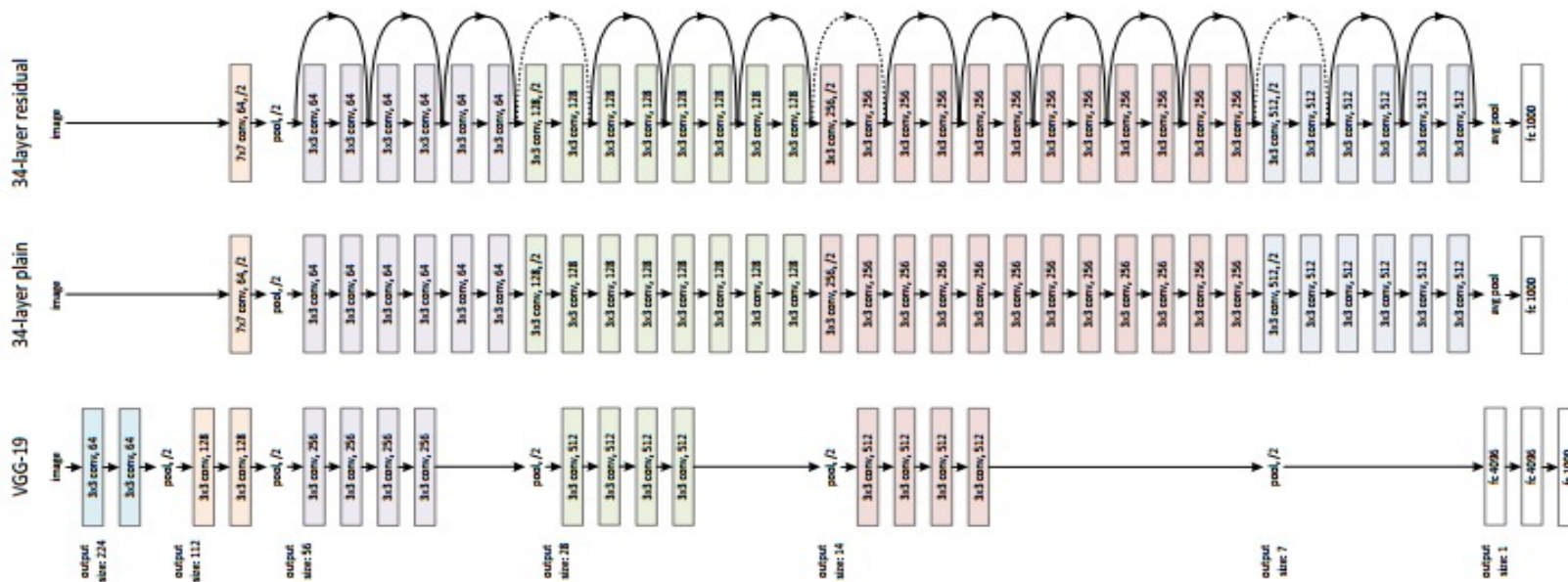
$$h(\mathbf{x}) = \underbrace{\mathbf{x}}_{\text{恒等函数}} + \underbrace{(h(\mathbf{x}) - \mathbf{x})}_{\text{残差函数}} \rightarrow f(\mathbf{x}, \theta)$$

残差单元



卷积神经网络-ResNet

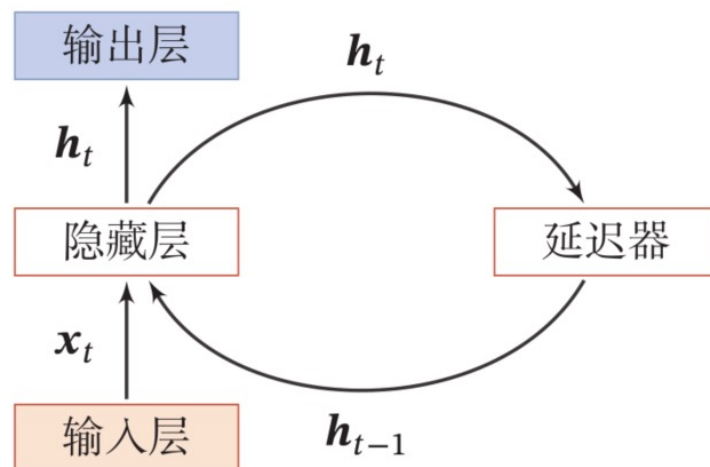
- 2015 ILSVRC winner (152层)
- 错误率：3.57%



循环神经网络

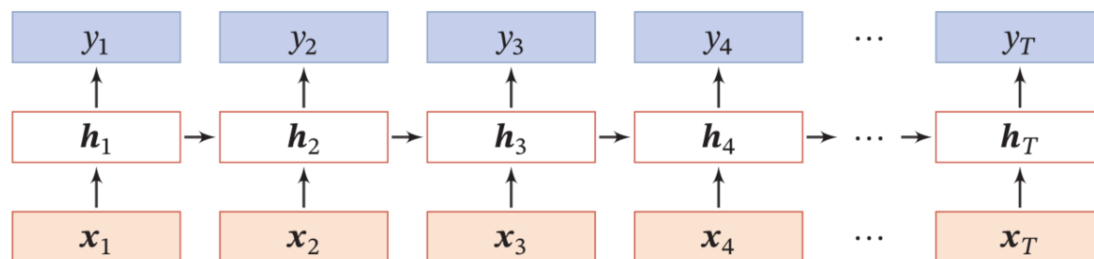
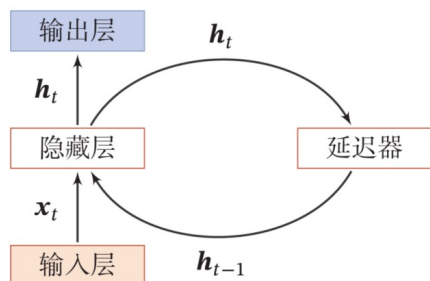
- 循环神经网络通过使用带自反馈的神经元，能够处理任意长度的时序数据。

$$h_t = f(h_{t-1}, x_t)$$



循环神经网络

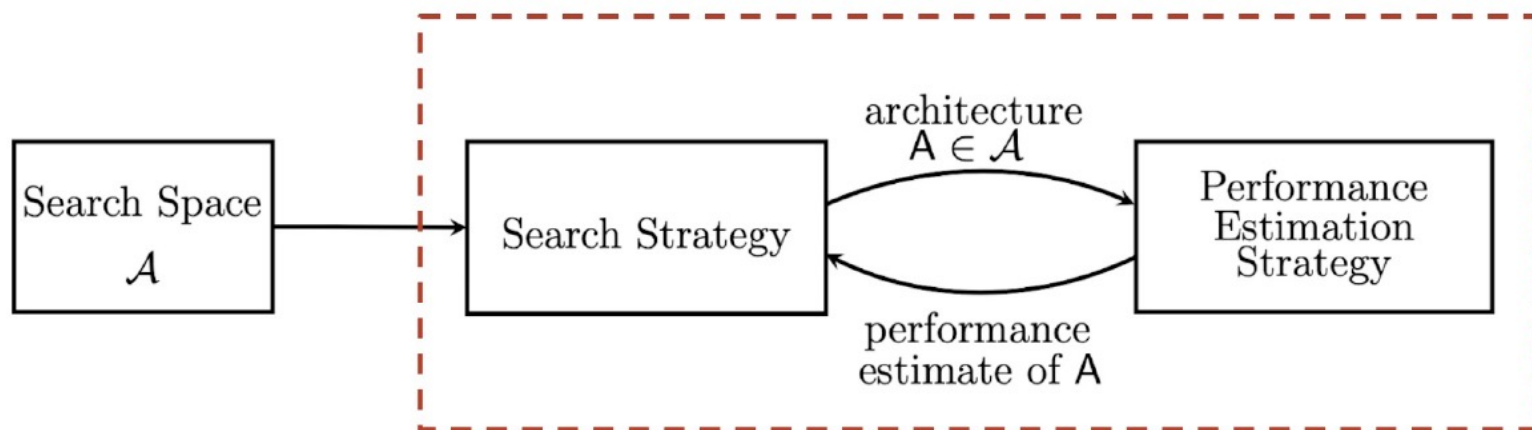
- 循环神经网络通过使用带自反馈的神经元，能够处理任意长度的时序数据。



- 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

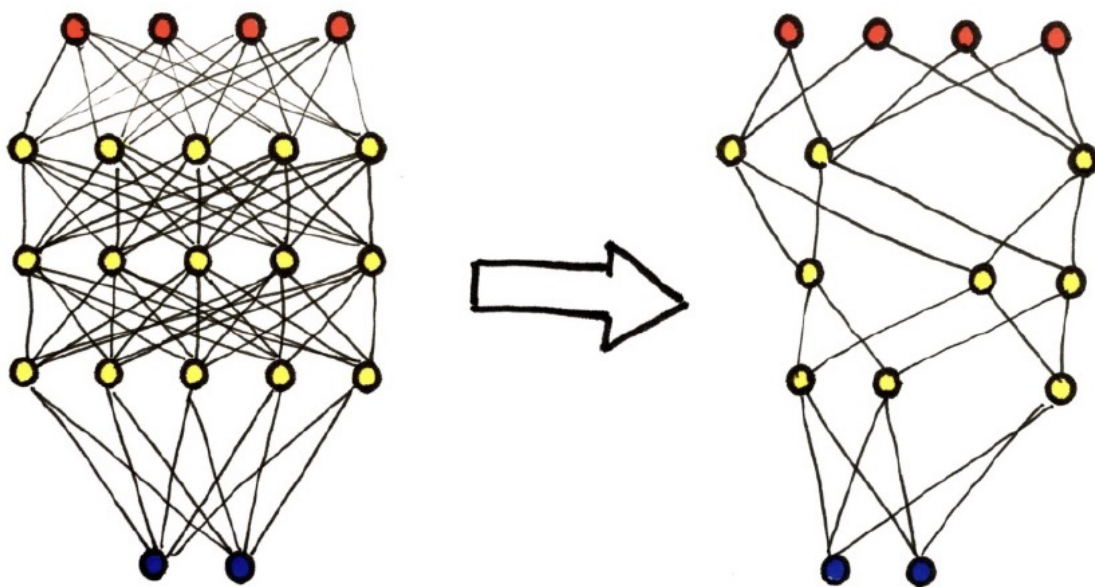
如何设计合适的网络架构？

□ 神经结构搜索（Neural Architecture Search）



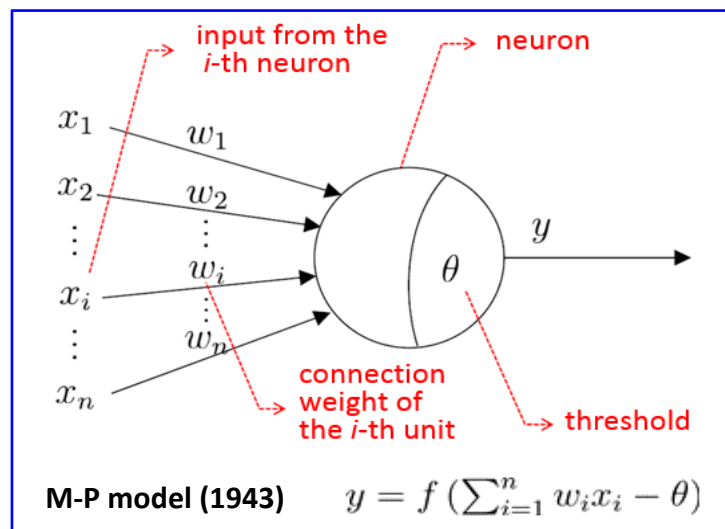
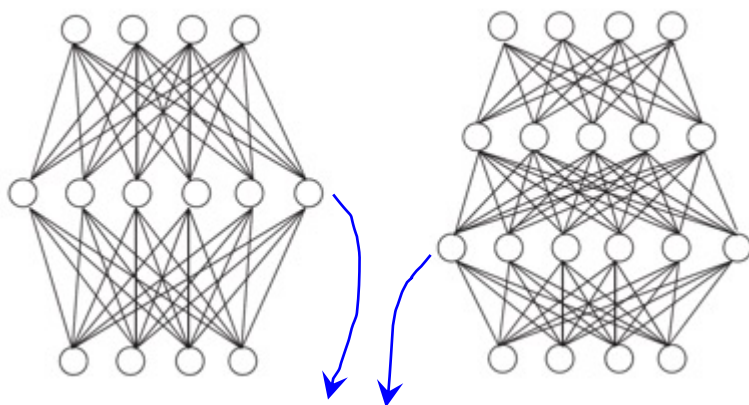
如何使神经网络变得更轻？

▣ 神经结构压缩 (Neural Network Compression)



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:

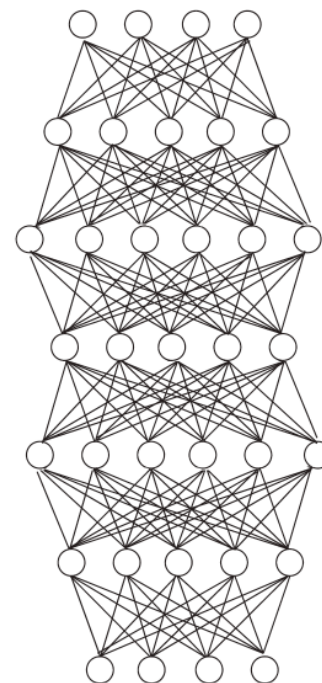
2012: 8 层

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

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》

IEEE Spectrum：这些天我们看到了许多关于深度学习的新闻



Yann LeCun

CNN的主要发明人
深度学习“三驾马车”之一
2019年图灵奖得主

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。

深度学习并非“突然出现”的
“颠覆性技术”，

而是经过了长期发展、
很多研究者做出贡献，

“冷板凳”坐“热”的结果

例如：卷积神经网络

引发深度学习热潮，被广泛应用

信号处理中的卷积 [最晚1903年已在文献中出现]

D. Hubel & T. Wiesel 关于
猫视皮层的研究 [1962]

福岛邦彦(Fukushima)
在神经网络中引入卷积 [1982]

G. Hinton研究组将8层CNN用于
ImageNet竞赛获胜 [2012]

30年

H. Lee et al. 引入无监督
逐层训练CNN [2009]

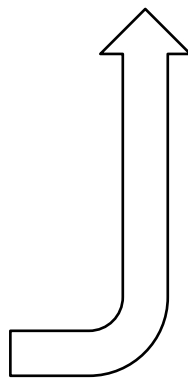
Y. LeCun 引入BP算法训练
卷积网络, CNN成型
[1989]

20年

G. Hinton通过无监督逐层
训练, 构建深层模型 [2006]

Y. LeCun and Y. Bengio,
完整描述CNN [1995]

Y. LeCun et al., CNN 用于
支票手写字符识别[1998]





2019年3月27日，ACM宣布：
Geoffrey Hinton, Yann LeCun, Yoshua Bengio
因对深度学习的卓越贡献获得图灵奖

科学的发展总是
“螺旋式上升”

三十年河东
三十年河西

坚持才能有结果



理解深度学习

从“特征工程” 到 “特征学习” 或 “表示学习”

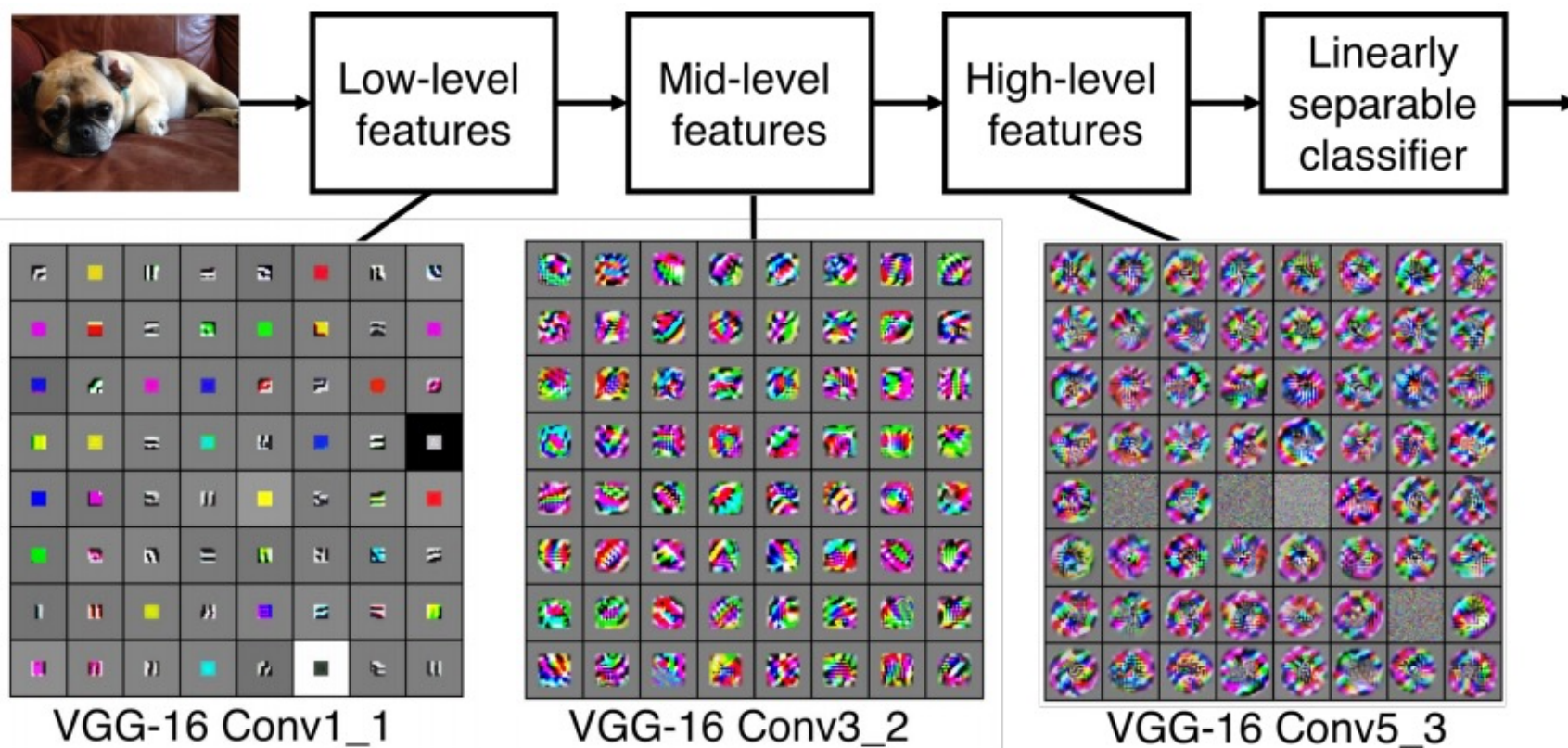
- 特征工程由人类专家根据现实任务来设计，特征提取与识别是分开的两个阶段



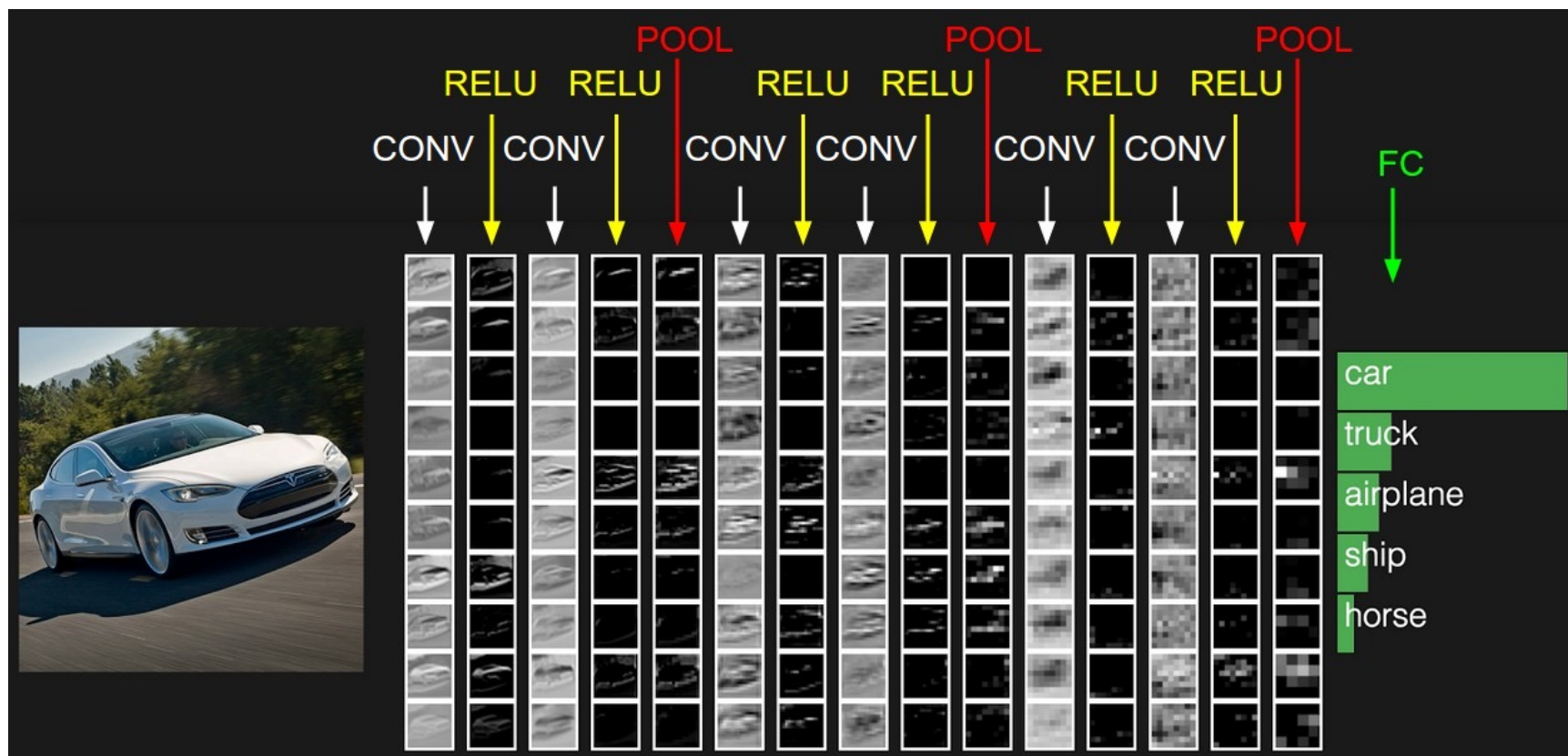
- 特征学习通过深度学习自动产生有益于分类的特征，是一个端到端的学习框架。



表示学习



表示学习



小结

- 神经网络历史
 - 神经元模型：熟悉
 - 感知机与多层网络：熟悉
 - 误差逆传播算法：熟悉
 - 全局最小与局部最小：了解
 - 深度学习：熟悉
-