第五讲 神经网络

高级机器学习

大纲

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

神经网络发展史

第一阶段（1943~1969年，感知机为代表）

- 1943年, McCulloch和Pitts 提出第一个神经元数学模型, 即M-P模型, 并从原理上证明了人工神经网络能够计算任何算数和逻辑函数
- 1949年, Hebb 发表《The Organization of Behavior》一书, 提出生物神经元学习的机理, 即Hebb学习规则
- 1958年, Rosenblatt 提出感知机网络（Perceptron）模型和其学习规则
- 1960年, Widrow和Hoff提出自适应线性神经元（Adaline）模型和最小均方学习算法
- 1969年, Minsky和Papert 发表《Perceptrons》一书, 指出单层神经网路不能解决非线性问题, 多层网络的训练算法尚无希望. 这个论断导致神经网络进入低谷

神经网络发展史

第二阶段（1982~2000，反向传播算法为代表）

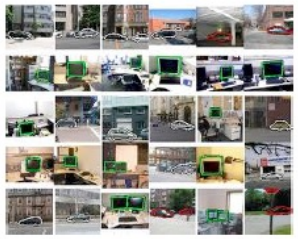
- 1982年, 物理学家Hopfield提出了一种具有联想记忆、优化计算能力的递归网络模型, 即Hopfield 网络
- 1986年, Rumelhart 等编辑的著作《Parallel Distributed Processing: Explorations in the Microstructures of Cognition》报告了反向传播算法
- 1987年, IEEE 在美国加州圣地亚哥召开第一届神经网络国际会议（ICNN）
- 90年代初, 伴随统计学习理论和SVM的兴起, 神经网络由于理论不够清楚, 试错性强, 难以训练, 再次进入低谷

神经网络发展史

第三阶段（2006~迄今，深度网络为代表）

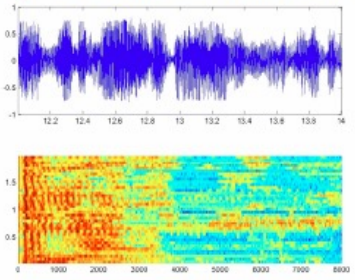
- 2006年, Hinton提出了深度信念网络(DBN), 通过“预训练+微调”使得深度模型的最优化变得相对容易
- 2012年, Hinton 组参加ImageNet 竞赛, 使用 CNN 模型以超过第二名10个百分点的成绩夺得当年竞赛的冠军
- 伴随云计算、大数据时代的到来，计算能力的大幅提升，使得深度学习模型在计算机视觉、自然语言处理、语音识别等众多领域都取得了较大的成功

Images & Video

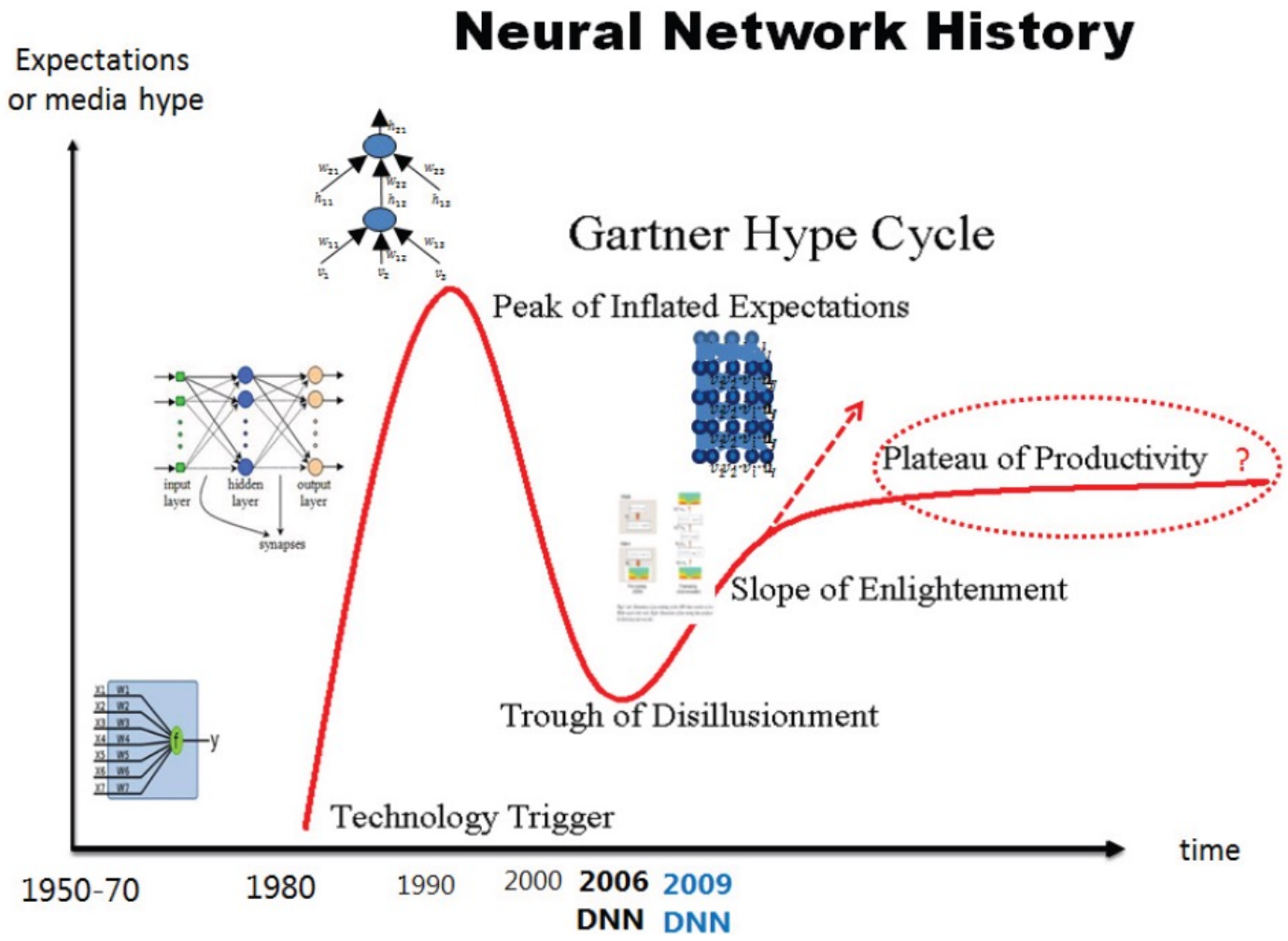


Text & Language

Speech & Audio



神经网络发展史

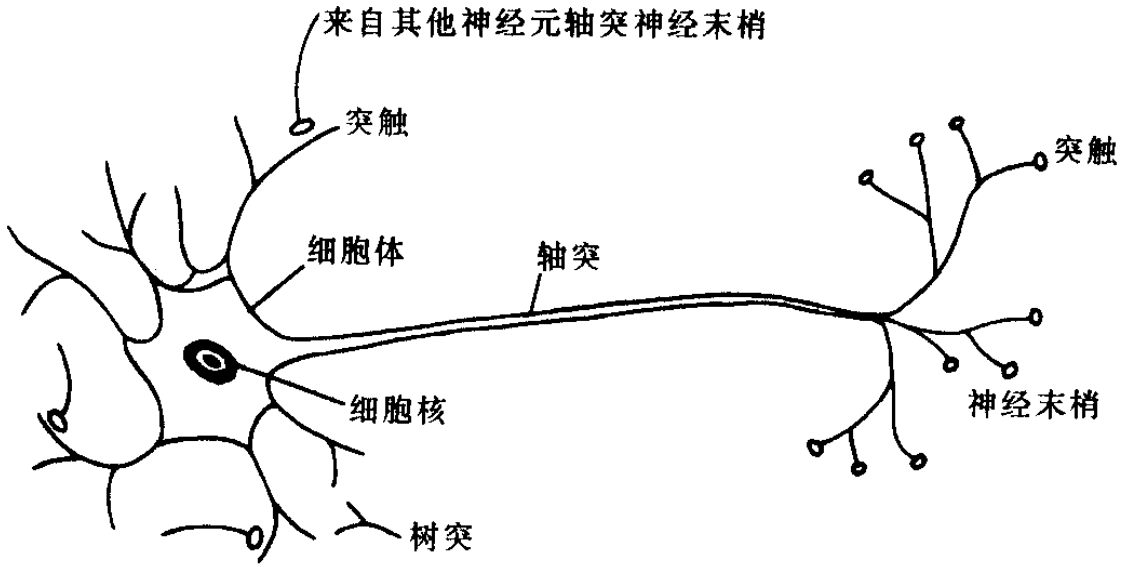

Neural Network History
Expectations or media hype
Gartner Hype Cycle
Peak of Inflated Expectations
Plateau of Productivity
?
Slope of Enlightenment
Trough of Disillusionment
Technology Trigger
input layer
hidden layer
output layer
synapses
time
1950-70
1980
1990
2000
2006 DNN
2009 DNN

大纲

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

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]

- **输入**：来自其他 n个神经元传递过来的信号
- **处理**：通过带权重连接进行传递, 总值与神经元的阈值比较
- **输出**：通过激活函数得到输出

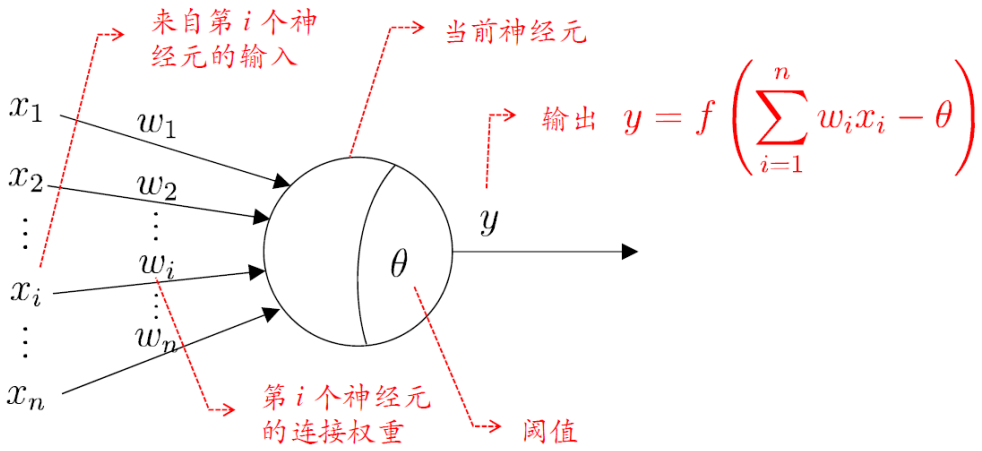
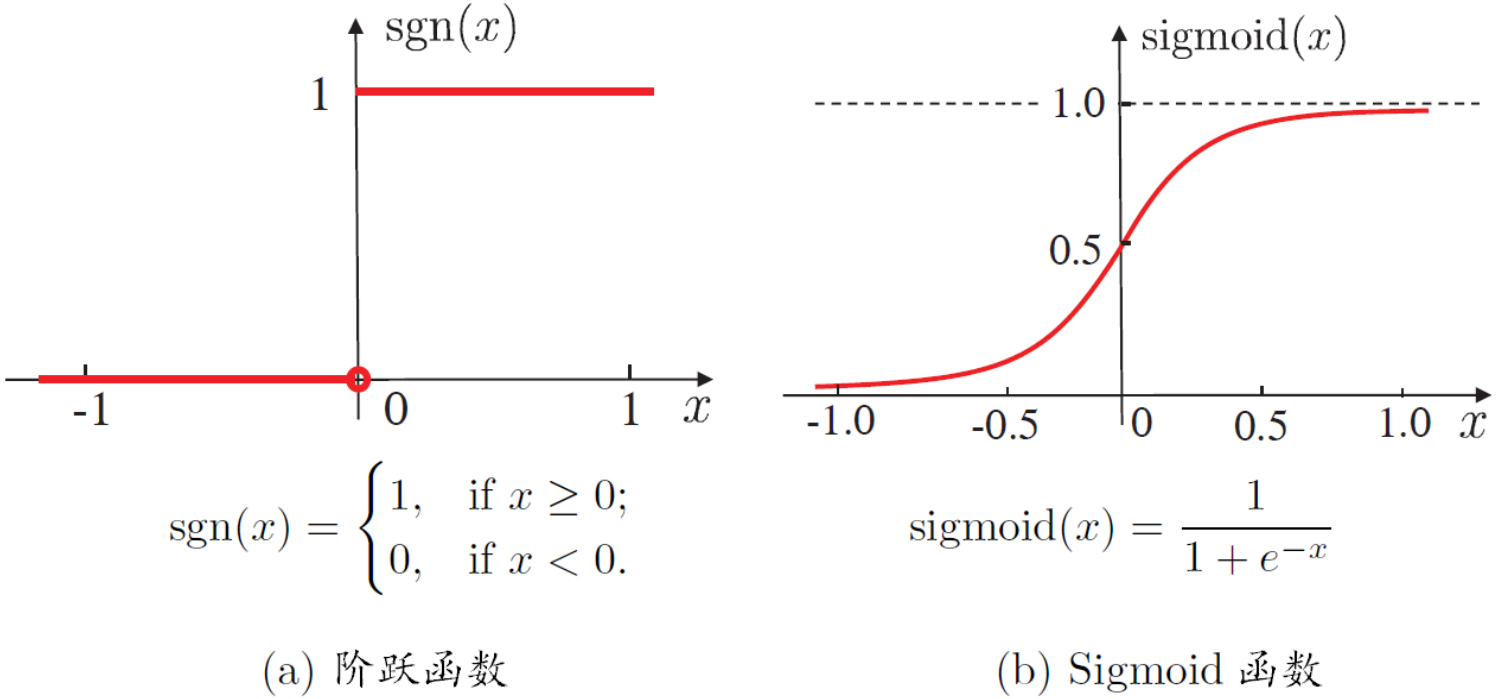


图 5.1 M-P 神经元模型

神经元模型-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



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

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

(a) 阶跃函数　　(b) Sigmoid 函数

图 5.2　典型的神经元激活函数

感知机与多层网络

- 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）

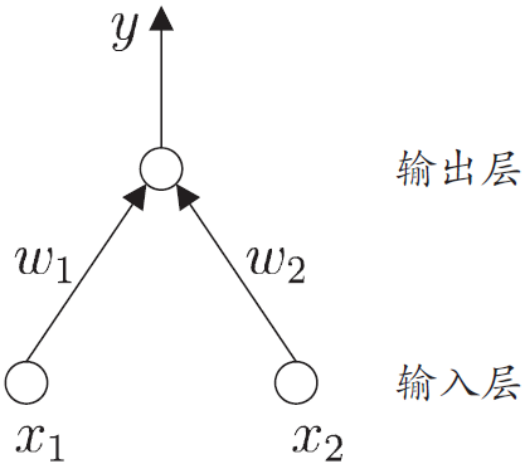


图 5.3　两个输入神经元的感知机网络结构示意图

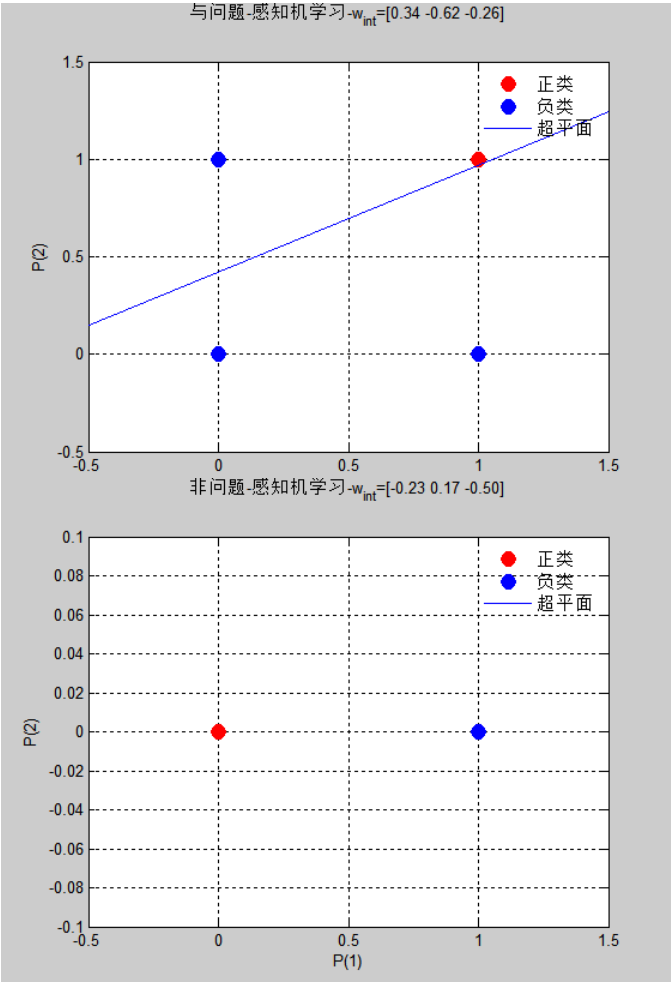
感知机与多层网络

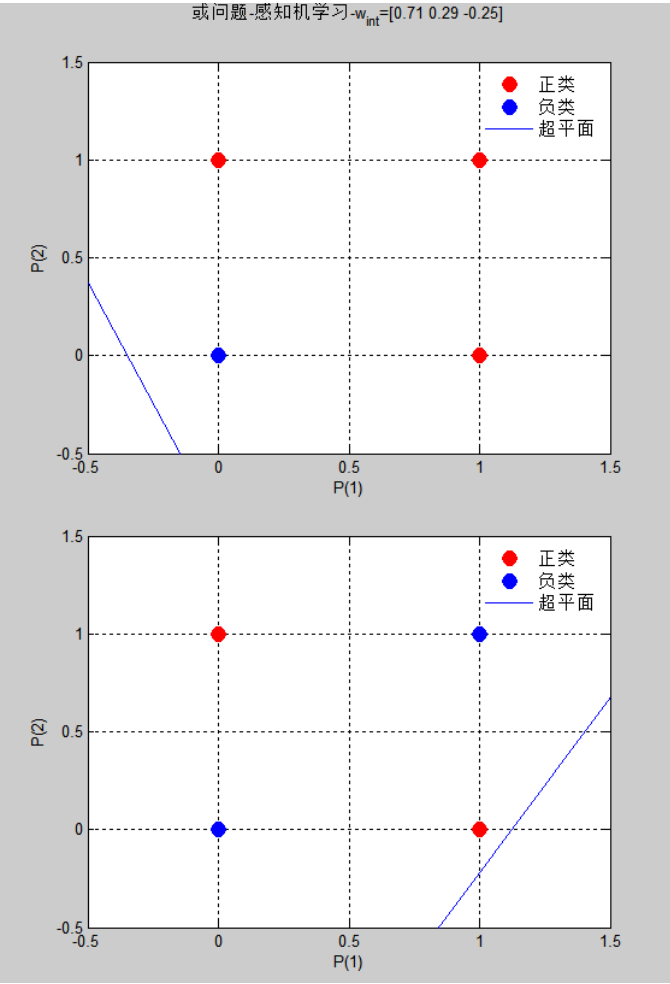
- 感知机能够容易实现逻辑与、或、非运算

  - “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, $y = 1$.

感知机与多层网络

感知机求解异、或、非问题的示例





感知机与多层网络

求解异、或、非问题的感知机模型

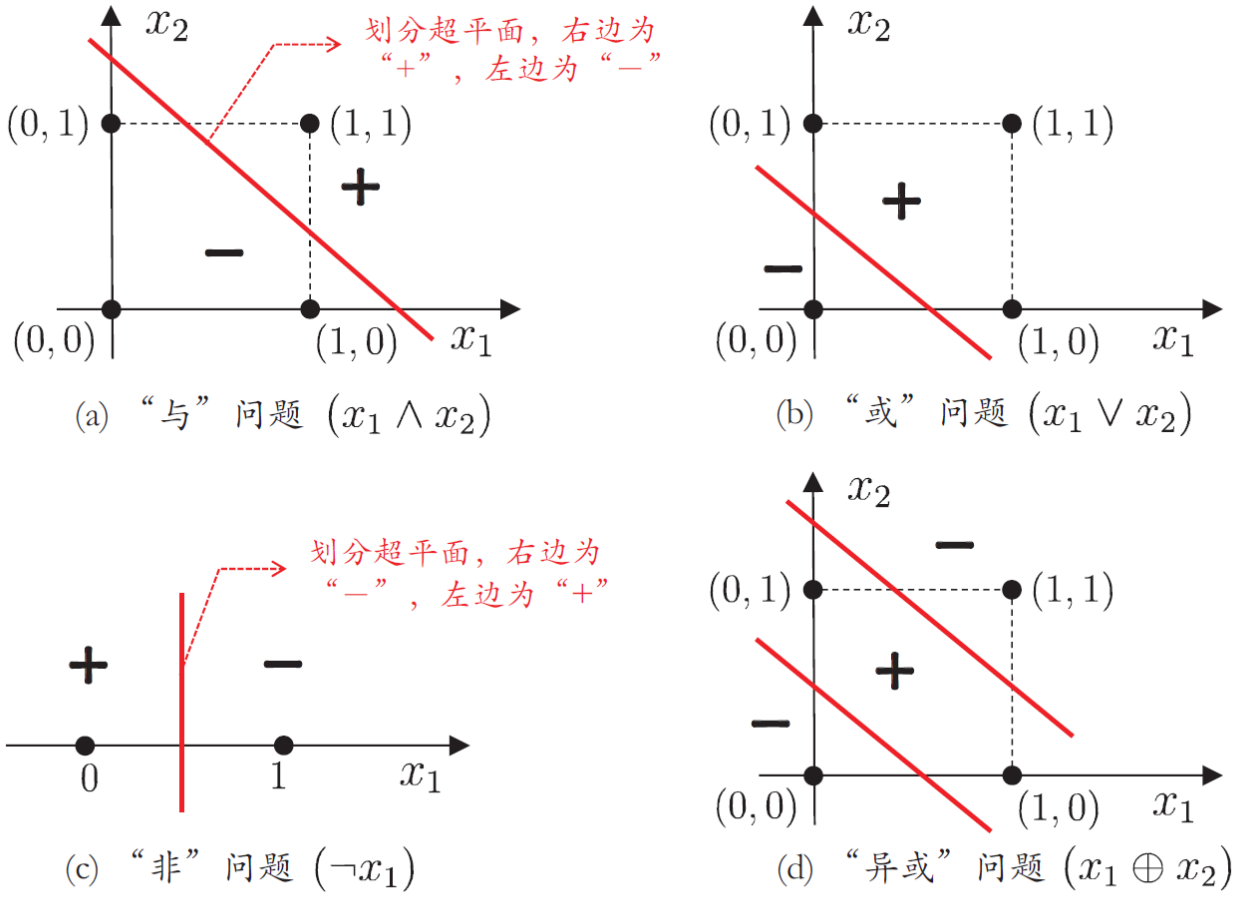


图 5.4 线性可分的“与”“或”“非”问题与非线性可分的“异或”问题

感知机与多层网络 - 感知机学习能力

- 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

感知机与多层网络 – 多层感知机

- 构建两层感知机，求解异或问题

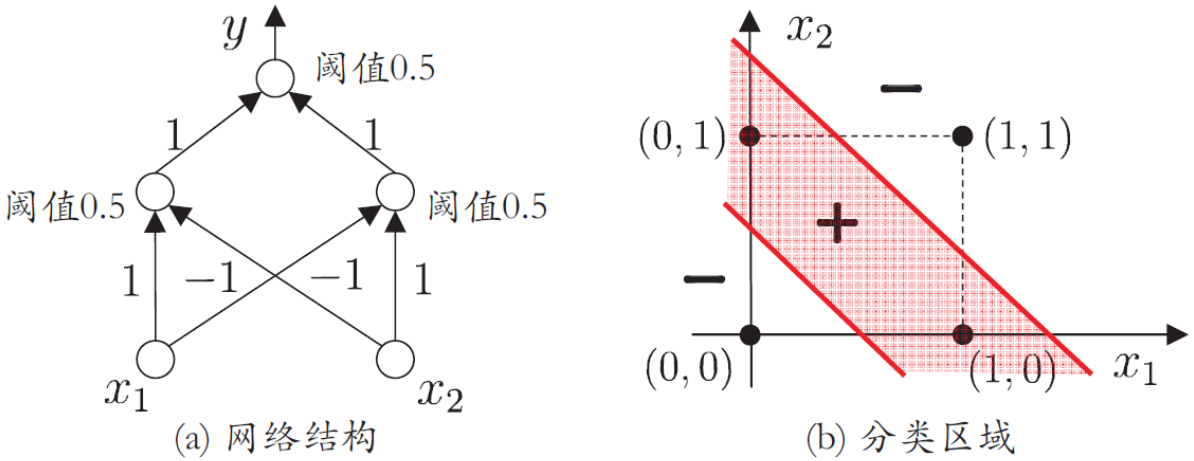
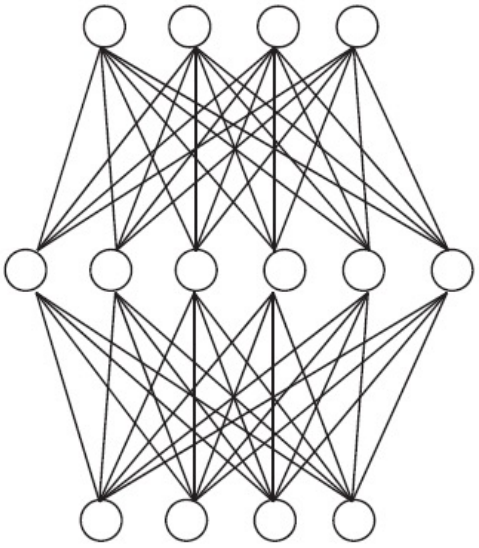


图 5.5　能解决异或问题的两层感知机

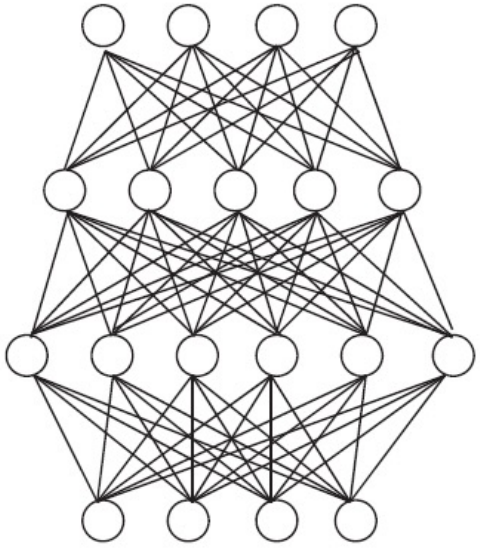
- 我们将输出层与输入层之间的一层神经元, 称为**隐层或隐含层**
- 隐含层和输出层神经元都是具有激活函数的功能神经元

感知机与多层网络 – 多层前馈神经网络

- 定义：每两层神经元全互联，不存在同层连接和跨层连接
- 前馈：接受外界输入信号，隐含层与输出层神经元对信号进行加工输出
- 学习：根据训练数据调整神经元的“连接权”以及功能神经元的“阈值”
- 多层网络：包含隐层的网络



(a) 单隐层前馈网络



(b) 双隐层前馈网络

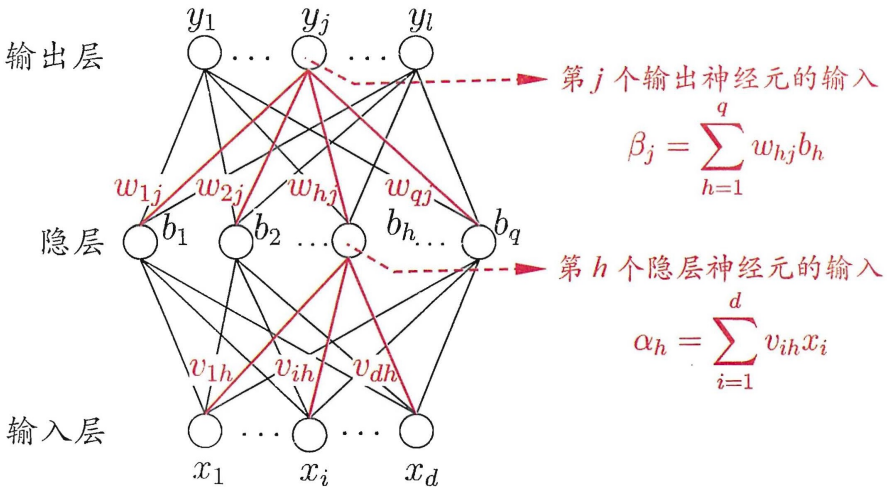
大纲

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

误差逆传播算法

误差逆传播算法（Error BackPropagation，简称BP）是最成功的训练多层前馈神经网络的学习算法.

- 给定训练集 $D=\{(\boldsymbol{x}_i,y_i)\},\boldsymbol{x}_i\in R^d,y_i\in R^l,(i=1,2,\cdots,m)$，即输入示例由$d$个属性描述，输出 l 维实值向量.
- 为方便讨论，给定一个拥有 d个输入神经元，l个输出神经元，q个隐层神经元的多层前向前馈网络结构.



- 记号：

θ_j：输出层第j个神经元阈值；

γ_h：隐含层第h个神经元阈值；

v_{ih}：输入层与隐层神经元之间的连接权重；

w_{hj}：隐层与输出层神经元之间的连接权重

误差逆传播算法

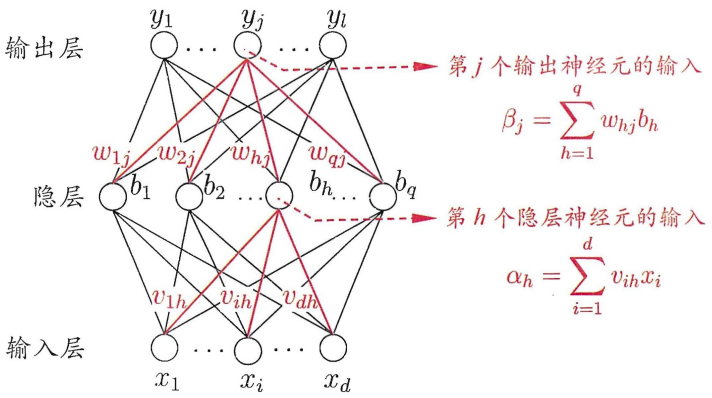
对于样例 $(\boldsymbol{x}_k, \boldsymbol{y}_k)$，假设网络的实际输出为 $\hat{\boldsymbol{y}}_k$

前向计算

step1: $b_h = f(\alpha_h - \gamma_h)$ ， $\alpha_h = \sum_{i=1}^{d} v_{ih} x_i$

step2: $\hat{y}_j^k = f(\beta_j - \theta_j)$ ， $\beta_j = \sum_{i=q}^{d} w_{hj} b_h$

step3: $E_k = \frac{1}{2} \sum_{j=1}^{l} (\hat{y}_j^k - y_j^k)^2$



参数数目

权重：v_{ih}，w_{hj}　阈值：θ_j，γ_h　$(i = 1, \cdots, d, h = 1, , \cdots, q, j = 1, \cdots, l)$

因此网络中需要 $(d + l + 1)q + l$ 个参数需要优化

参数优化

BP是一个迭代学习算法，在迭代的每一轮中采用广义的感知机学习规则对参数进行更新估计，任意的参数 v 的更新估计式为

$$v \leftarrow v + \triangle v.$$

误差逆传播算法

- BP算法基于梯度下降策略, 以误差率为目标，计算负梯度方向对参数进行调整

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

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

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

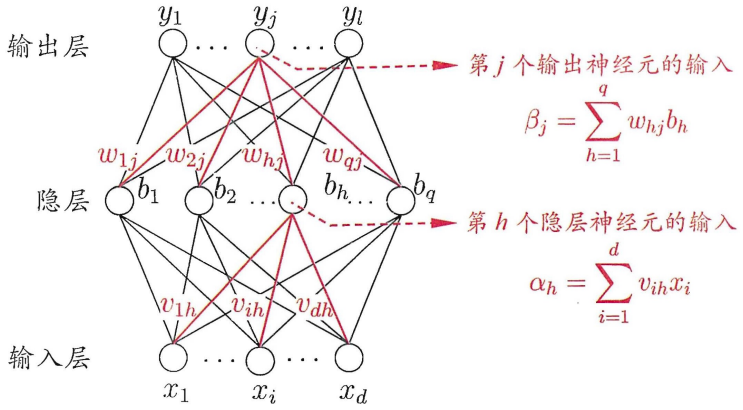


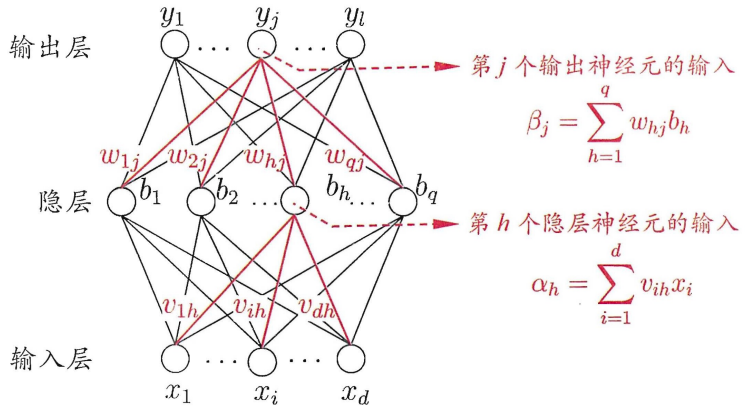
图 5.2 中的 Sigmoid 函数有一个很好的性质:

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

误差逆传播算法

- BP算法基于梯度下降策略, 以误差率为目标，计算负梯度方向对参数进行调整

$$\frac{\partial \beta_j}{\partial w_{hj}} = b_h \ .$$



$$\Delta w_{hj} = \eta g_j b_h \ .$$

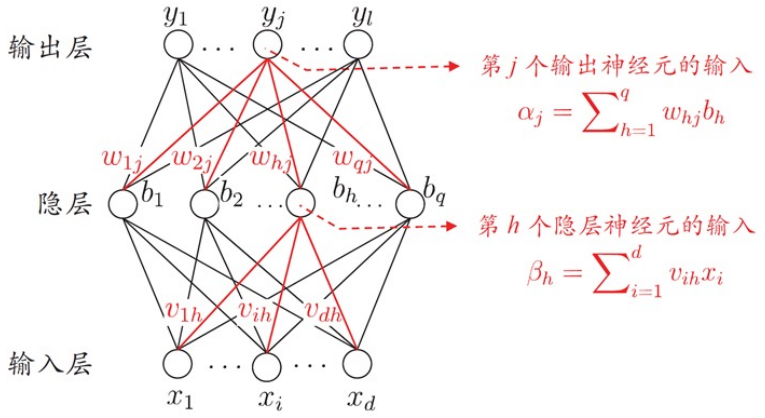
误差逆传播算法

类似的可以推导出其它结果，例如，

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

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

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



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

- 学习率 $\eta \in (0,1)$ 控制着算法每一轮迭代中的更新步长, 若太长则让容易震荡, 太小则收敛速度又会过慢.

误差逆传播算法

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

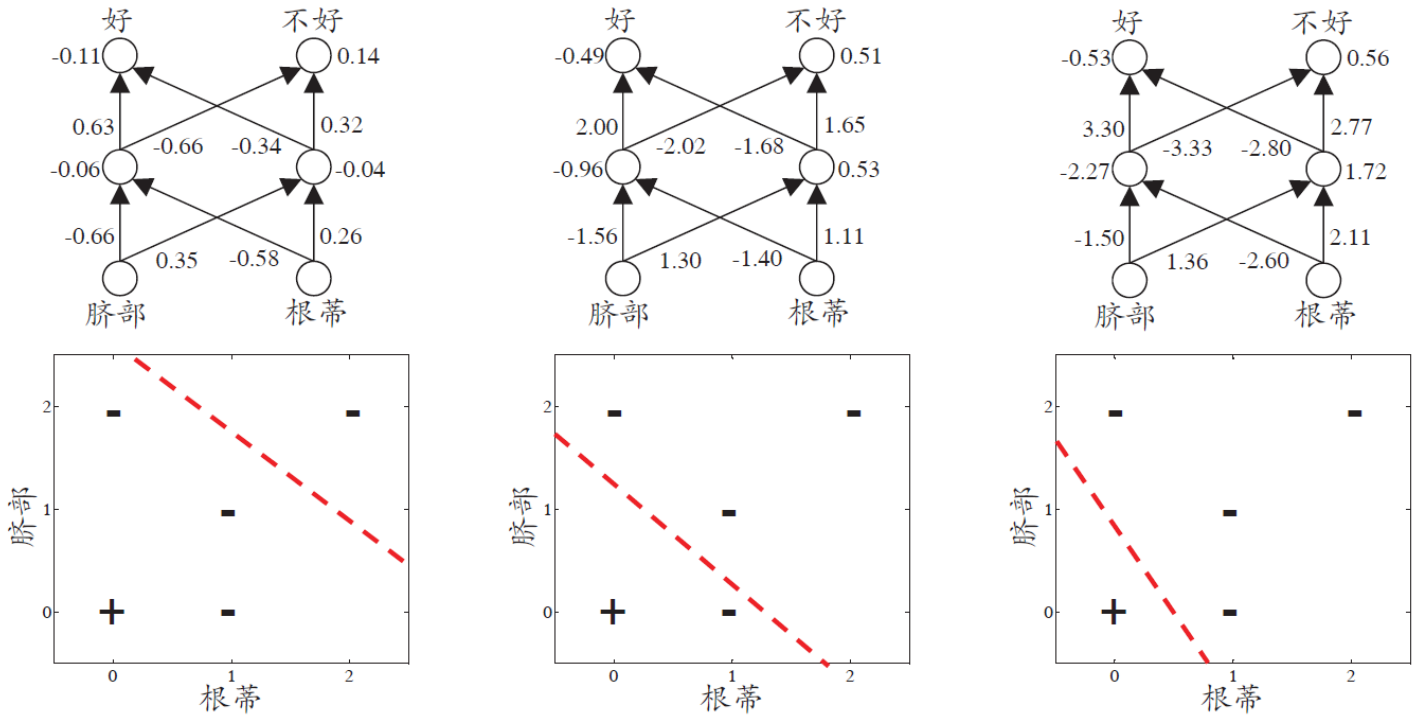
过程:

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

误差逆传播算法



(a) 第 25 轮　　(b) 第 50 轮　　(c) 第 100 轮

图 5.9　在 2 个属性、5 个样本的西瓜数据上, BP网络参数更新和分类边界的变化情况

误差逆传播算法 - 具体实现

- 标准 BP 算法
  - 每次对单个训练样例更新权值与阈值；单次计算开销小，但参数更新频繁, 不同样例对参数更新可能会相互冲抵，迭代次数较多
- 累计 BP 算法
  - 最小化整个训练集上的累计误差 $E = \frac{1}{m}\sum_{k=1}^{m} E_k$
  - 读取整个训练集后对更新参数，参数更新频率低，但单次计算开销大
- 实际应用
  - 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算法更为青睐, 尤其当训练数据规模巨大时效果更为明显.

误差逆传播算法 - 多层前馈网络优缺点

- 多层前馈网络具有强大的学习能力: **包含足够多神经元的隐层, 多层前馈神经网络能以任意精度逼近任意复杂度的连续函数** [Hornik et al., 1989]
- 多层前馈网络的局限
  - 多层前馈神经网络由于强大的表示能力, 经常遭遇过拟合。
  - 如何设置隐层神经元个数是难题，实际应用中常使用“试错法”
- 缓解过拟合的一些策略
  - **早停**：在训练过程中, 若训练误差降低, 但验证误差升高, 则停止训练
  - **正则化**：在误差目标函数中增加一项描述网络复杂程度的成分, 防止模型过于复杂，例如连接权值与阈值的平方和

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

大纲

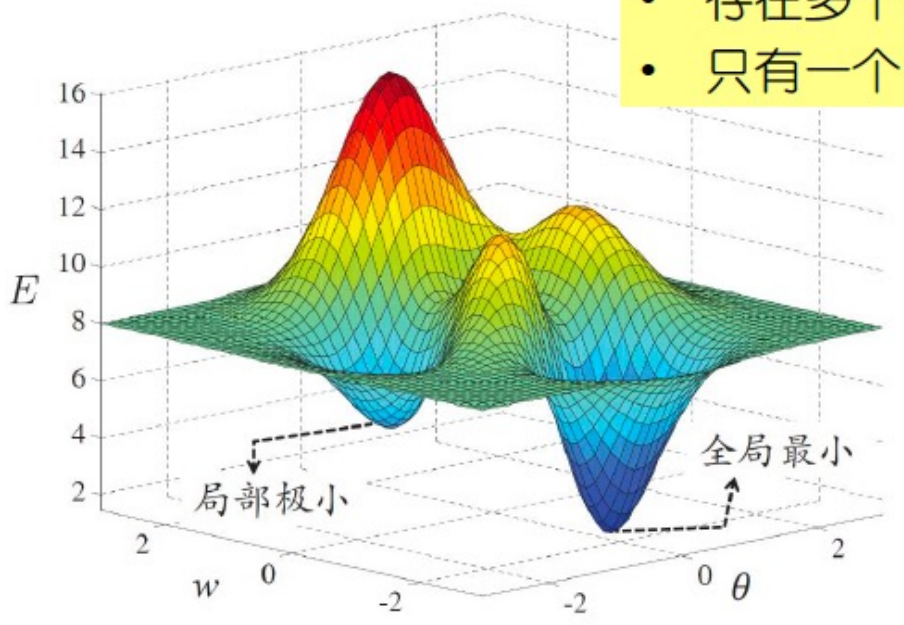
- 神经网络历史
- 神经元模型
- 感知机与多层网络
- 误差逆传播算法
- 全局最小与局部最小
- 深度学习
- 小结

全局最小与局部极小

学习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
- ✓ 遗传算法
- ✓ ……

大纲

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

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 – 训练方法

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 – 训练方法

- 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.

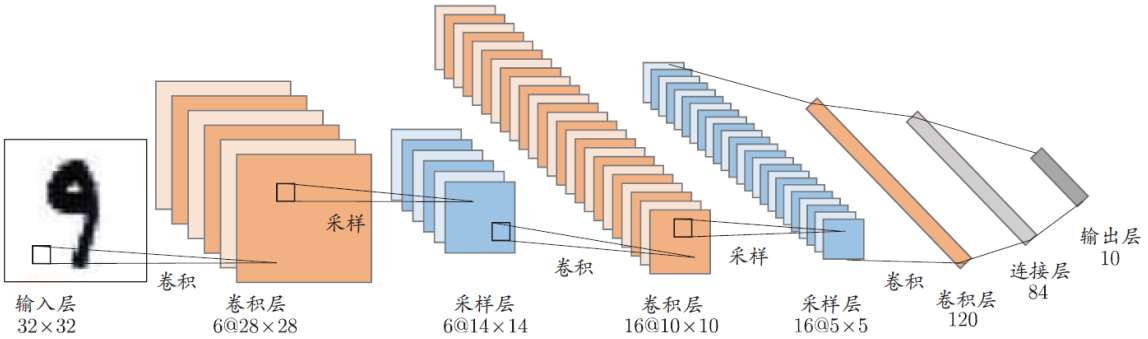


图 5.15　卷积神经网络用于手写数字识别 [LeCun et al., 1998]

深度学习模型 – 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
- 采样层：亦称“汇合层”, 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
- 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

- 卷积神经网络激活函数
  - 在CNN中采用修正的线性函数，增加特征稀疏性 $f(x) = \max(0, x)$
- 卷积神经网络训练
  - CNN 可用BP进行训练。但在训练中, 将卷积层和采样层的每一组神经元约定为相同的连接权, 从而大幅减少了参数数目

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

手工设计特征 → 分类识别

- 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.

学习生成特征 → 分类识别

大纲

- 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：了解
- 小结