

西安电子科技大学  
XIDIAN UNIVERSITY

# 人工智能概论

西安电子科技大学 人工智能学院

西安电子科技大学  
XIDIAN UNIVERSITY

## 第十章 神经网络

- 10.1 神经网络的生物学基础**
- 10.2 神经网络的基本概念及历史**
- 10.3 神经网络结构**
- 10.4 学习算法**

人工智能学院

西安电子科技大学  
XIDIAN UNIVERSITY

## 第十章 神经网络

- 10.1 神经网络的生物学基础**
- 10.2 神经网络的基本概念及历史**
- 10.3 神经网络结构**
- 10.4 学习算法**



人工智能学院

西安电子科技大学  
XIDIAN UNIVERSITY

## 神经网络的生物学基础

### ➤ 生物神经系统

- 最早人们并不知人的“智慧”是人的大脑的功能，以为“智慧”是从“肚子”中来，如说“宰相肚里能撑船”；又如以为智慧是发自“心”，如说“某人很有心计”。直到近代科学才明确，**智慧（思维）是人的大脑的功能的表现**。
- 大脑是由无数的脑细胞组成。既然“思维”是大脑的功能的表现，即**智慧是脑神经网络的功能**。那么人们希望利用**人工神经网络**来模拟人脑的神经网络，研究其性能，希望从中悟出人的思维的一些“奥秘”。
- **人工神经网络技术**为人工智能提供新的解决问题的方法，并广泛应用于各个领域。

人工智能学院

 西安交通大学  
XIDIAN UNIVERSITY

## 神经网络的生物学基础

➤ 生物神经系统



人工智能学院

 西安交通大学  
XIDIAN UNIVERSITY

## 神经网络的生物学基础

➤ 生物神经系统

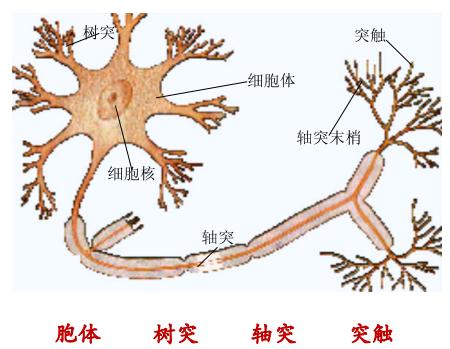
- 生物神经系统是一个有高度组织和相互作用的数量巨大的细胞组织群体。人类大脑的神经细胞大约在 $10^{11}$ - $10^{13}$ 个左右。
- 神经细胞也称**神经元**，是神经系统的基本单元，它们按不同的结合方式构成了复杂的**神经网络**。通过神经元及其联接的可塑性，使得大脑具有学习、记忆和认知等各种智能。

人工智能学院

 西安交通大学  
XIDIAN UNIVERSITY

## 神经网络的生物学基础

➤ 生物神经系统



➤ 生物神经元主要由以下几个部分组成：

- 胞体：是神经细胞的本体；
- 树突：用以接受来自其它神经元的信号；
- 轴突：用以控制信号的输出；
- 突触：是一个神经元与另一个神经元相联系的特殊部位，通过神经元轴突的端部靠化学接触和电接触将信号传递给下一个神经元的树突或胞体。

人工智能学院

 西安交通大学  
XIDIAN UNIVERSITY

## 神经网络的生物学基础

➤ 生物神经元的基本工作机制

- 一个神经元有两种状态——**兴奋**和**抑制**。
- 平时处于抑制状态的神经元，其**树突**和**胞体**接受其它神经元经由**突触**传来的兴奋电位，多个输入在神经元中以代数和的方式叠加；
- 如输入兴奋总量超过阈值，神经元被激发进入兴奋状态，发出输出脉冲，由**轴突**的突触传递给其它神经元。

人工智能学院

 西安交通大学  
XIDIAN UNIVERSITY

## 神经网络的生物学基础

- 生物神经特性

### 1. 并行分布处理的工作模式

实际上大脑中单个神经元的信息处理速度是很慢的，每次约1毫秒(ms)，比通常的电子门电路要慢几个数量级。每个神经元的处理功能也很有限，估计不会比计算机的一条指令更复杂。但人脑对某一复杂过程的处理和反应却很快，一般只需几百毫秒。因此只能把它看成是一个由众多神经元所组成的超高密度的并行处理系统。

大脑信息处理的并行速度已达到了极高的程度。

人工智能学院

 西安交通大学  
XIDIAN UNIVERSITY

## 神经网络的生物学基础

- 生物神经特性

### 2. 神经系统的可塑性和自组织性。

神经系统的可塑性和自组织性与人脑的生长发育过程有关。如人的幼年时期约在9岁左右，学习语言的能力十分强，说明在幼年时期，大脑的可塑性和柔软性特别良好。从生理学的角度看，它体现在突触的可塑性和联接状态的变化，同时还表现在神经系统的自组织特性上。

人工智能学院

 西安交通大学  
XIDIAN UNIVERSITY

## 神经网络的生物学基础

- 生物神经特性

### 2. 神经系统的可塑性和自组织性。

如在某一外界信息反复刺激下，接受该信息的神经细胞之间的突触结合强度会增强。这种可塑性反映出大脑功能既有先天的制约因素，也有可能通过后天的训练和学习而得到加强。

神经网络的学习机制就是基于可塑性，并通过修正突触的结合强度来实现的。

人工智能学院

 西安交通大学  
XIDIAN UNIVERSITY

## 神经网络的生物学基础

- 生物神经特性

### 3. 信息处理与信息存贮合二为一。

大脑中的信息处理与信息存贮是有机结合在一起的，而不像现行计算机那样，存贮地址和存贮内容是彼此分开的。由于大脑神经元兼有信息处理和存贮功能，所以在进行回忆时，不但不存在先找存贮地址而后再调出所存内容的问题，而且还可以由一部分内容恢复全部内容。

人工智能学院

 西安交通大学  
XIDIAN UNIVERSITY

## 神经网络的生物学基础

- 生物神经特性
- 4. 信息处理的系统性。
 

大脑是一个复杂的大规模信息处理系统，单个的元“神经元”不能体现全体宏观系统的功能。实际上，可以将大脑的各个部位看成是一个大系统中的许多子系统。各个子系统之间具有很强的相互联系，一些子系统可以调节另一些子系统的行为。

如视觉系统和运动系统就存在很强的系统联系，可以相互协调各种信息处理功能。

人工智能学院

 西安交通大学  
XIDIAN UNIVERSITY

## 神经网络的生物学基础

- 生物神经特性
- 5. 信息的分布存储提供容错功能。
 

由于信息被分布存放在几乎整个网络中，所以当其中的某一个点或者某几个点被破坏时，信息仍然可以被存取。系统在受到局部损伤时还可以正常工作。
- 6. 能接受和处理模糊的、模拟的、随机的信息。
- 7. 求满意解而不是精确解。

人类处理日常行为时，往往都不是一定要按最优或最精确的方式去求解，而是以能解决问题为原则，即求得满意解就行了。

人工智能学院

 西安交通大学  
XIDIAN UNIVERSITY

## 第十章 神经网络

- 10.1 神经网络的生物学基础
- 10.2 神经网络的基本概念及历史
- 10.3 神经网络结构
- 10.4 学习算法



人工智能学院

 西安交通大学  
XIDIAN UNIVERSITY

## 神经网络的基本概念与历史

- 人工神经网络
- (1) 生理结构的模拟:
 

用仿生学观点，**探索人脑的生理结构**，把对人脑的微观结构及其智能行为的研究结合起来即人工神经网络方法。
- (2) 宏观功能的模拟:
 

从人的思维活动和智能行为的**心理学特性出发**，利用计算机系统来对人脑智能进行宏观功能的模拟，即**符号处理方法**。

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 神经网络的基本概念与历史

### ➤ 人工神经网络

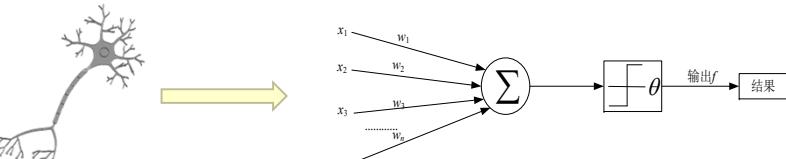
神经网络 (Neural Network, NN) 一般也称为人工神经网络 (Artificial Neural Network, ANN)，是科学家们在对生物的神经元、神经系统等生理学的研究取得了突破性进展以及对人脑的结构、组成和基本工作单元有了进一步认识的基础上，通过借助数学和物理的方法从信息处理的角度对人脑神经网络进行抽象后建立的简化模型。

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 神经网络的基本概念与历史

### ➤ 神经网络的抽象描述



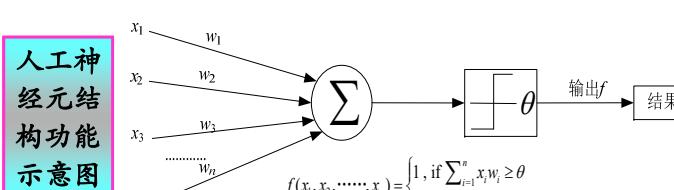
根据生物神经元可以抽象出一个神经元模型：该模型接受多个输入信号 $x_i$ ，并将其处理为 $h_\theta(x)$ 进行输出。该神经元模型可以看作一个简单的分类器，根据不同的输入给出对应的输出，类似于生物神经元的兴奋或抑制。

人工智能学院

 西安电子科技大学

## 神经网络的基本概念与历史

### ➤ 人工神经元



**人工神经元结构功能示意图**

$$f(x_1, x_2, \dots, x_n) = \begin{cases} 1, & \text{if } \sum_{i=1}^n x_i w_i \geq \theta \\ 0, & \text{otherwise} \end{cases}$$

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 神经网络的基本概念与历史

### ➤ 人工神经网络



和人脑的神经元类似，神经元之间的相互堆叠与连接会形成一种网络结构，这样的网络结构被称为人工神经网络。输入的信息在各神经元之间传递，最后在网络的末端输出处理后的结果。这个结果包含着整个神经网络对输入数据的学习与认知，因此人工神经网络是一种信息处理工具。

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 神经网络的基本概念与历史

### ➤ 人工神经网络的特点

- ◆ 大规模并行处理
 

人工神经网络的基本结构模仿人脑，具有并行处理的特征，可以大大提高工作速度。
- ◆ 信息的分布式存放
  - 信息的分布式存储提供容错功能。  
由于信息被分布存放在几乎整个网络中，所以，当其中的某一个点或者某几个点被破坏时，信息仍然可以被存取。
  - 系统在受到局部损伤时还可以正常工作。
- ◆ 学习能力
  - 人工神经网络可以根据所在的环境去改变它的行为
  - 不同的人工神经网络模型，有不同的学习/训练算法

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 神经网络的基本概念与历史

### ➤ 人工神经网络发展历史



The timeline shows three main periods:

- 启蒙萌芽时期 1940s至1960s**
- 低潮反思时期 1960s至1970s**
- 新的发展时期 1990s中后期之后**

**启蒙萌芽时期 1940s至1960s**

1943年McCullouch和Pitts提出了神经元的数学描述和网络的结构方法，这标志着神经网络计算时代的开始。1957年Rosenblatt定义一个称为感知器的神经网络结构，第一次把神经网络从纯理论的探讨推向了工程实现，掀起了神经网络研究的高潮。

**低潮反思时期 1960s至1970s**

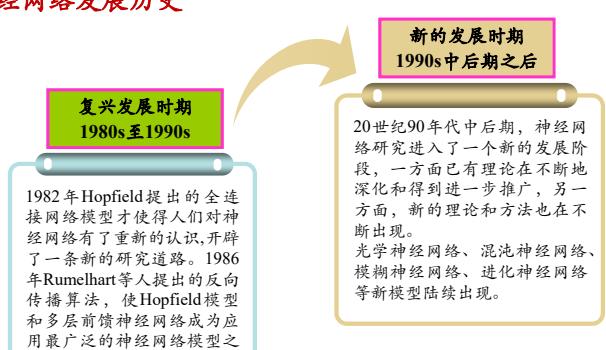
Minsky和Papert在1969年发表论著《Perceptrons》指出感知器仅能解决一阶谓词逻辑，只能完成线性划分，对于非线性或者其他分类会遇到很多困难，就连简单的XOR（异或）问题都解决不了。由此，神经网络的研究进入了反思期。

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 神经网络的基本概念与历史

### ➤ 人工神经网络发展历史



The timeline shows two main periods:

- 复兴发展时期 1980s至1990s**
- 新的发展时期 1990s中后期之后**

**复兴发展时期 1980s至1990s**

1982年Hopfield提出的全连接网络模型才使得人们对神经网络有了重新的认识，开辟了一条新的研究道路。1986年Rumelhart等人提出的反向传播算法，使Hopfield模型和多层前馈神经网络成为应用最广泛的神经网络模型之一。

**新的发展时期 1990s中后期之后**

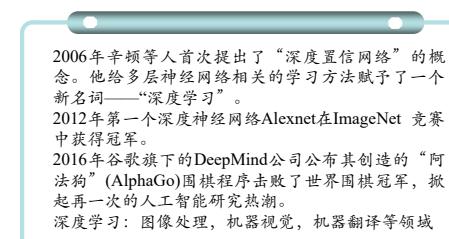
20世纪90年代中后期，神经网络研究进入了一个新的发展阶段，一方面已有理论在不断地深化和得到进一步推广，另一方面，新的理论和方法也在不断出现。光学神经网络、混沌神经网络、模糊神经网络、进化神经网络等新模型陆续出现。

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 神经网络的基本概念与历史

### ➤ 人工神经网络发展历史



The timeline shows one main period:

- 深度神经网络 2006年后**

**深度神经网络 2006年后**

2006年辛顿等人首次提出了“深度置信网络”的概念。他给多层次神经网络相关的学习方法赋予了一个新名词——“深度学习”。2012年第一个深度神经网络Alexnet在ImageNet竞赛中获得冠军。2016年谷歌旗下的DeepMind公司公布其创造的“阿法狗”(AlphaGo)围棋程序击败了世界围棋冠军，掀起再一次的人工智能研究热潮。深度学习：图像处理，机器视觉，机器翻译等领域

人工智能学院

西安交通大学  
XIDIAN UNIVERSITY

## 第十章 神经网络

- 10.1 神经网络的生物学基础
- 10.2 神经网络的基本概念及历史
- 10.3 神经网络结构**
- 10.4 学习算法



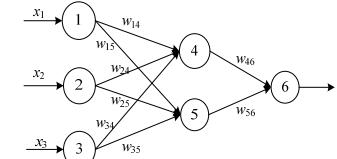
人工智能学院

西安交通大学  
XIDIAN UNIVERSITY

## 神经网络结构

➤ 人工神经网络是具有下列特性的有向图

- 对于每个节点*i*存在一个状态变量 $x_i$ ;
- 从节点*j*至节点*i*, 存在一个连接权系数 $w_{ij}$ ;
- 对于每个节点*i*, 存在一个阈值 $\theta_i$ ; □ □
- 对于每个节点*i*, 定义一个传输/激励函数 $f_i$ ;

$$y_j = f\left(\sum_{i=1}^n w_{ij} x_i - \theta_j\right)$$


人工智能学院

西安交通大学  
XIDIAN UNIVERSITY

## 神经网络结构

➤ 神经网络中常见模型

- 神经网络中常见模型: 前馈神经网络、反馈神经网络。
- 前馈神经网络: 各神经元接受前一层的输入并输出给下一层, **无反馈**, 常见的有BP神经网络和RBF径向基神经网络。
- 反馈神经网络: 其输入数据决定反馈系统的初始状态, 然后系统经过一系列的状态转移后逐渐收敛于平衡状态, 即为反馈神经网络经过计算后的输出结果。

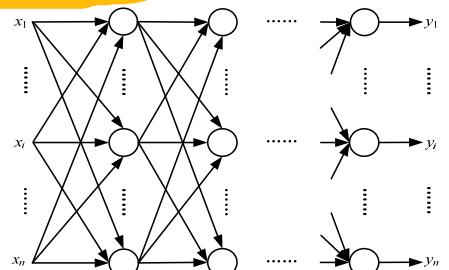
人工智能学院

西安交通大学  
XIDIAN UNIVERSITY

## 神经网络结构

➤ 前馈型网络

- 前馈型网络的信号由输入层到输出层单向传输;
- 每层的神经元仅与其前一层的神经元相连, 仅接受前一层传输来的信息;
- 是一种最为广泛使用的神经网络模型, 因为它本身的结构也不太复杂, 学习和调整方案也比较容易操作, 而且由于采用了多层的网络结构, 其求解问题的能力也得到明显的加强, 基本上可以满足使用要求。



人工智能学院

**神经网络结构**

西电  
XIDIAN UNIVERSITY

➤ 反馈型网络

- 输入层不仅接受外界的输入信号，同时接受网络自身的输出信号。输出反馈信号可以是原始输出信号，也可以是经过转化的输出信号；可以是本时刻的输出信号，也可以是经过一定延迟的输出信号；
- 此种网络经常用于系统控制、实时信号处理等需要根据系统当前状态进行调节的场合；

人工智能学院

**神经网络结构**

西电  
XIDIAN UNIVERSITY

➤ 单输入人工神经元

输入 通用神经元

$a = f(w \cdot p + b)$

w:  $1 \times 1$  (单个神经元)  
p:  $1 \times 1$  (1个输入)

神经元输出 激励函数 权值 输入 偏置(值)

人工智能学院

**神经网络结构**

西电  
XIDIAN UNIVERSITY

➤ 多输入人工神经元

$a = f(Wp + b)$

W:  $1 \times R$  (单个神经元)  
p:  $R \times 1$  (R个输入)

人工智能学院

**神经网络结构**

西电  
XIDIAN UNIVERSITY

➤ 激励函数总结

名称	输入/输出关系	图标	Matlab函数
硬极限函数	$\begin{cases} a=0 & n < 0 \\ a=1 & n \geq 0 \end{cases}$		Hardlim
对称极限函数	$\begin{cases} a=-1 & n < 0 \\ a=1 & n \geq 0 \end{cases}$		Hardlims
线性函数	$a=n$		Pureline
饱和线性函数	$\begin{cases} a=0 & n < 0 \\ a=n & 0 \leq n < 1 \\ a=1 & n \geq 1 \end{cases}$		Satlin

人工智能学院

**神经网络结构**

西安交通大学  
XIDIAN UNIVERSITY

➤ 激励函数总结

对数S型函数	$a = \frac{1}{1 + e^{-n}}$		Logsig
对称饱和线性函数	$\begin{cases} a = -1 & n < 0 \\ a = n & 0 \leq n \leq 1 \\ a = 1 & n \geq 1 \end{cases}$		Satlns
双曲正切S型函数	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$		Tanhsg
正线性函数	$\begin{cases} a = 0 & n < 0 \\ a = n & n \geq 0 \end{cases}$		Poslin

人工智能学院

**神经网络结构**

西安交通大学  
XIDIAN UNIVERSITY

➤ 激励函数作用的实例

$p = [1, 2]^T$

$W = [1, 1]$

$b = -1.5$

$n = Wp + b = 1.5$

$f(n)$

$a$

人工智能学院

**神经网络结构**

西安交通大学  
XIDIAN UNIVERSITY

➤ 激励函数作用的实例

$p_1 \cdot w_{1,1}$

$p_2 \cdot w_{1,2}$

$\vdots$

$p_R \cdot w_{1,R}$

$w_{1,1}$

$w_{1,2}$

$\vdots$

$w_{1,R}$

$b$

$n$

$a = \text{Hardlim}(1.5) \rightarrow 1$

$a = \text{Hardlims}(1.5) \rightarrow 1$

$a = \text{Pureline}(1.5) \rightarrow 1.5$

$a = \text{Satin}(1.5) \rightarrow 1$

$a = \text{Logsig}(1.5) \rightarrow 0.8176$

$f$

人工智能学院

**神经网络结构**

西安交通大学  
XIDIAN UNIVERSITY

➤ 网络结构——神经元的层

$W: S$ 个神经元

$p: R$ 个输入

输入

$p_1, p_2, \dots, p_R$

$w_{1,1}, w_{1,2}, \dots, w_{1,R}$

$b_1$

$\sum$

$n_1$

$f$

$a_1$

$S$ 个神经元的层

$\vdots$

$w_{S,1}, w_{S,2}, \dots, w_{S,R}$

$b_S$

$\sum$

$n_S$

$f$

$a_S$

人工智能学院

**神经网络结构**

➤ 神经元的层简化模型

$$a = f(\mathbf{W}\mathbf{p} + \mathbf{b})$$

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & \ddots & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$

$$\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_R \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_S \end{bmatrix}, \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_S \end{bmatrix}$$

**W: S × R (S个神经元)**    **p: R × 1 (R个输入)**

人工智能学院

**神经网络结构**

➤ 多层神经网络 (3层)

$$a^1 = f^1(W^1\mathbf{p} + b^1)$$

$$a^2 = f^2(W^2a^1 + b^2)$$

$$a^3 = f^3(W^3a^2 + b^3)$$

人工智能学院

**神经网络结构**

➤ 多层神经网络 (3层) 简化表示

$$a^3 = f^3(W^3f^2(W^2f^1(W^1\mathbf{p} + b^1) + b^2) + b^3)$$

**W<sup>i</sup>: S × R (多个神经元)**    **p: R × 1 (R个输入)**, 2个中间层 (隐层)

人工智能学院

**神经元模型**

➤ 第一个神经元运算模型

1943年, McCulloch和Pitts提出首个神经元模型, 该模型是一个简单的二元分类器, 接受d维的输入:  $(x_1, x_2, \dots, x_d), x_i \in \{0,1\}$ , 随后对这个d维的输入进行加权求和, 并将其与阈值进行比较, 若大于阈值输出1, 否则输出0, 其数学表达式如下:

$$y = f\left(\sum_{i=1}^d w_i x_i - \theta\right)$$

硬极限函数  $f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$

多输入, 单个神经元

人工智能学院

**神经元模型**

原始神经元模型的缺点

- 输入输出都是二元的
- 无法对连续的实数进行处理，也不能处理多分类问题
- 不能训练也没有学习能力
- 权重是人工设定的，无法根据输出自动调整

该神经元可以看作是一个定义好的逻辑门电路，主要的意义在于验证了神经元模型具有计算的能力。

第一个神经元的数学描述和网络的结构方法，这标志着神经网络计算时代的开始。

人工智能学院

**感知机模型**

神经元模型的改进——感知机

1957年，Rosenblatt定义了一个称为感知机（perceptron）的神经网络结构，该模型与神经元模型类似，主要有以下几点改进：

- 输入为一个实数向量；
- 有多种激活函数可以选择；
- 属于一个可学习模型；

$$y = f\left(\sum_{i=1}^d w_i x_i - \theta\right)$$

多输入，单个神经元

人工智能学院

**感知机模型**

感知机的意义

- 感知机中第一次引入了学习的概念，使**人脑所具备的学习功能在基于符号处理的数学模型中得到了一定程度的模拟**，所以引起了广泛的关注。
- 简单感知机：简单感知机模型实际上仍然是M-P模型的结构。它是一种单层感知机模型，一层为输入层（**只负责接收输入信号，无信息处理能力**），另一层具有计算单元，可以通过**采用监督学习来逐步增强模式划分的能力**，达到学习的目的。

人工智能学院

**感知机模型**

感知机模型的学习方法

感知机与神经元模型最大的区别在于感知机模型可以对训练样本进行学习，以符号函数为例，其训练过程如下：

- 将感知机模型用向量进行表示：  

$$h(\mathbf{x}) = \text{sgn}(\mathbf{w}^T \mathbf{x})$$

例：  
 $\mathbf{w}^T \mathbf{x}^{(i)} < 0$   
 $y_{pre} = \text{sgn}(\mathbf{w}^T \mathbf{x}^{(i)}) = -1$   
 $y^{(i)} = 1$   
 $\mathbf{w}' = \mathbf{w} + \mathbf{x}^{(i)} y^{(i)}$   
 $\mathbf{w}^T \mathbf{x}^{(i)} > 0$   
 $y_{pre} = \text{sgn}(\mathbf{w}'^T \mathbf{x}^{(i)}) = 1$   
 $y^{(i)} = 1$
- 对于给定的训练样本，**挑出分类错误的样本对**：  
 $\{\mathbf{x}^{(i)}, y^{(i)}\}$
- 对权重向量进行更新：  

$$\mathbf{w} = \mathbf{w} + \mathbf{x}^{(i)} y^{(i)}$$

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 感知机模型

感知机算法实质上是一种赏罚过程：

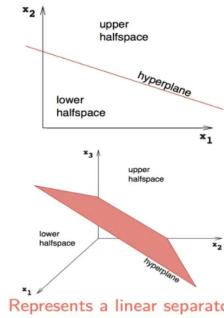
- 对正确分类的模式则“赏”，实际上是“不罚”，即权向量不变。
- 对错误分类的模式则“罚”，需要对权向量进行相应的更新。
- 当用全部模式样本训练过一轮以后，只要有一个模式是判别错误的，则需要进行下一轮迭代，即用全部模式样本再训练一次。
- 如此不断反复直到全部模式样本进行训练都能得到正确的分类结果为止。

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 感知机模型

➤ 感知机模型的数学原理



将输入 $(x_1, x_2, \dots, x_d)$ 看作d维空间中的一点，感知机模型实际上构建了一个超平面：

$$\sum_{i=1}^d w_i x_i - \theta = 0 \quad \sum_{i=1}^d w_i x_i + b = 0$$

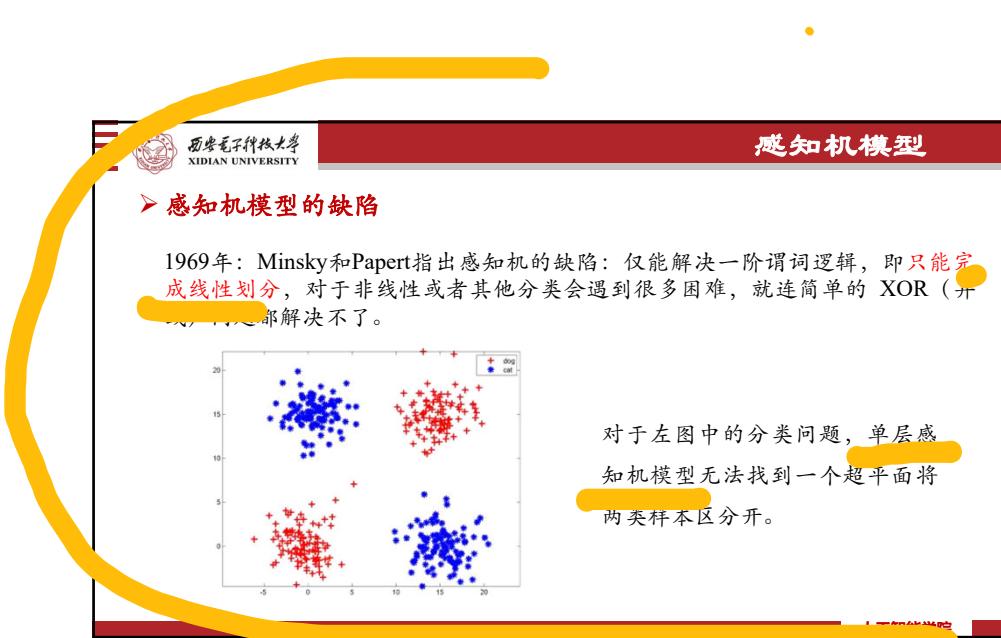
该平面将空间分成了两个部分，对应于平面上方和下方的点：

$$\sum_{i=1}^d w_i x_i + b > 0 \quad x \in \text{upper half space}$$

$$\sum_{i=1}^d w_i x_i + b < 0 \quad x \in \text{lower half space}$$

感知机实际上是通过构造超平面实现对不同点的分类。

人工智能学院

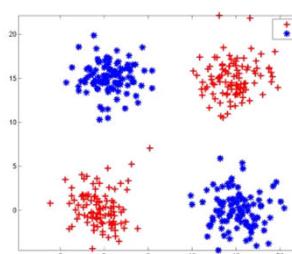


 西安电子科技大学  
XIDIAN UNIVERSITY

## 感知机模型

➤ 感知机模型的缺陷

1969年：Minsky和Papert指出感知机的缺陷：仅能解决一阶谓词逻辑，即只能完成线性划分，对于非线性或者其他分类会遇到很多困难，就连简单的 XOR（异或）问题都解决不了。



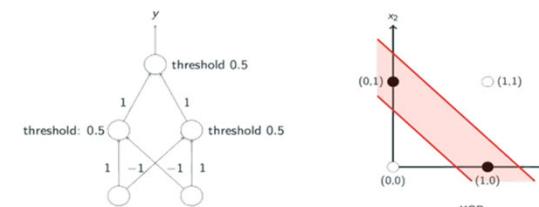
对于左图中的分类问题，单层感知机模型无法找到一个超平面将两类样本区分开。

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 多层感知机

异或问题无法用一个超平面将两类样本分隔开，于是人们考虑对多个感知机模型进行组合，即采用多个超平面去分割样本——多层感知机。



We can model XOR using a hidden layer.

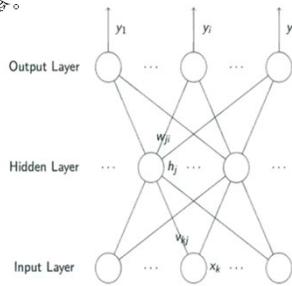
人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 多层感知机

### 多层感知机的难题

多层感知机具有多个神经元，之前神经元模型的学习方法不适用于多个神经元组成的网络。



多层感知机带来大量的权重，需要一种高效的学习方法来训练这样的网络。

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 第十章 神经网络

### 10.1 神经网络的生物学基础

### 10.2 神经网络的基本概念及历史

### 10.3 神经网络结构

### 10.4 学习算法



人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 梯度下降的思想

人们通常使用均方误差来衡量预测值与真实值之间的差距，此处考虑单输出的情况，即：

$$e = |y - y_{true}|^2$$

对于神经网络模型：

$$y = wx + b$$

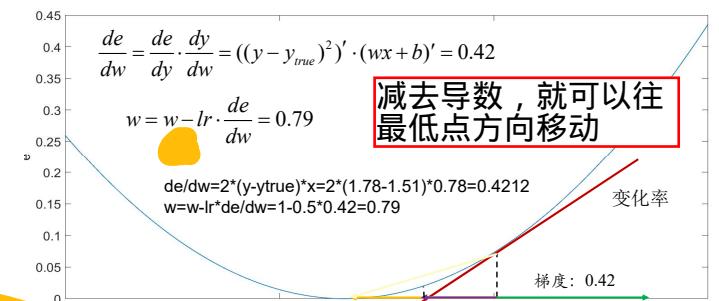
训练神经网络的目的实际上是通过确定合适的  $w, b$  使得误差  $e$  尽可能地变小。要想知道  $w, b$  是如何影响误差  $e$  的，可以用误差  $e$  对  $w, b$  进行求导；导数的方向为误差  $e$  上升最快的方向，只需将  $w, b$  向着导数的相反方向进行调整即可使误差  $e$  减小——梯度下降。

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 梯度下降的思想

例：若预测值： $y = wx + b = 1.78$ ，真实值  $y_{true} = 1.51$ ，其中  $w=1, b=1, x=0.78$ ，试用梯度下降法对  $w$  进行调整，假设学习率  $lr=0.5$ 。



变化率

梯度：0.42

减去导数，就可以往最低点方向移动

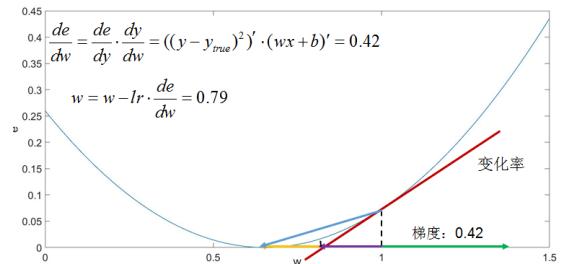
人工智能学院



## 梯度下降的思想

**注意：**为了使 $w$ 能较快地收敛于一个使函数 $e$ 极小的解， $lr$ 值的选择是很重要的。

- 若 $lr$ 值太小，则收敛太慢；
- 若 $lr$ 值太大，则搜索可能过头，引起发散。



人工智能学院



## 反向传播算法

- 1986年：Rumelhart等人提出的反向传播算法，为多层网络的训练提供了有力的工具。
- 为什么需要反向传播算法：梯度下降可以应对带有明确求导函数的情况，或者说可以应对那些可以求出误差的情况，我们可以把它看做没有隐层的网络；但对于多隐层的神经网络，输出层可以直接求出误差来更新参数，但其中隐层的误差是不存在的，因此不能对它直接应用梯度下降，而是先将误差反向传播至隐层，然后再应用梯度下降，其中将误差从末层往前传递的过程需要链式法则（Chain Rule）的帮助，因此反向传播算法可以说是梯度下降在链式法则中的应用。

人工智能学院



## 反向传播算法

- 多层网络的学习能力比单层感知机强得多。要训练多层网络，简单感知机学习规则显然不够，需要更强大的学习算法。
- 误差反向传播算法就是其中最杰出的代表，它是迄今最成功的神经网络学习算法。
- 现实任务中使用神经网络时，大多是在使用误差反向传播算法进行训练。该算法不仅可用于多层前馈神经网络，还可用于其他类型的神经网络，如训练递归神经网络。

人工智能学院



## 反向传播算法

- 在感知器算法中我们实际上是在利用理想输出与实际输出之间的误差作为增量来修正权值，然而在多层感知器中，我们只能计算出输出层的误差，中间隐层由于不直接与外界连接，其误差无法估计。
- 反向传播算法（BP算法）的思想：从后向前反向逐层传播输出层的误差，以间接计算隐层的误差。算法可以分为两个阶段：
  - 正向过程：从输入层经隐层逐层正向计算各单元的输出
  - 反向过程：由输出误差逐层反向计算隐层各单元的误差，并用此误差修正前层的权值。

人工智能学院


西安电子科技大学  
XIDIAN UNIVERSITY

# 反向传播算法

## ➤ 反向传播算法的简单描述

The diagram illustrates a feed-forward neural network with three layers: Input Layer, Hidden Layer, and Output Layer. The Input layer contains nodes labeled  $x_1, \dots, x_i, \dots, x_d$ . The Hidden layer contains nodes labeled  $h_1, \dots, h_k, \dots, h_n$ . The Output layer contains nodes labeled  $y_1, \dots, y_j, \dots, y_l$ . Red lines represent the weights  $w_{ij}$ ,  $v_{ik}$ , and  $v_{kj}$  connecting nodes between adjacent layers. Labels  $w_{ij}$ ,  $v_{ik}$ , and  $v_{kj}$  are placed near their respective red lines.

第  $j$  个输出神经元的输入:

$$\beta_j = \sum_{k=1}^p w_{kj} h_k$$

第  $k$  个隐层神经元的输入:  $h_k$  表示隐层第  $k$  个神经元的输出

$$\alpha_k = \sum_{i=1}^d v_{ik} x_i$$

$w_{kj}$  表示隐层第  $k$  个神经元与输出层第  $j$  个神经元相连的权重

$v_{ik}$  表示输入层第  $i$  个神经元与隐层第  $k$  个神经元相连的权重

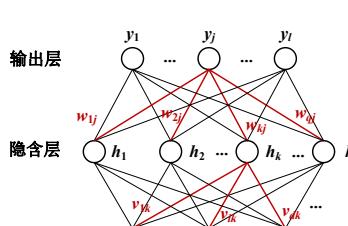
其中输出层第  $j$  个神经元的偏置用  $b_j$  表示, 隐层第  $k$  个神经元的偏置用  $\gamma_k$  表示。

人工智能学院


 西安电子科技大学  
 XIDIAN UNIVERSITY

反向传播算法

▶ 反向传播算法的简单描述



The diagram illustrates a feed-forward neural network with three layers: Input layer, Hidden layer, and Output layer. The Input layer consists of nodes  $x_1, \dots, x_i, \dots, x_d$ . The Hidden layer consists of nodes  $h_1, \dots, h_k, \dots$ . The Output layer consists of nodes  $y_1, \dots, y_j, \dots, y_t$ . Red lines represent the weights  $w_{ij}$ ,  $w_{ik}$ ,  $v_{jk}$ , and  $v_{ik}$  connecting neurons between adjacent layers.

假定网络的理想输出为：  $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_t)$

假定网络的实际输出为：  $\mathbf{y} = (y_1, y_2, \dots, y_t)$

则网络均方误差为：

$$E = \frac{1}{2} \sum_{j=1}^t (\hat{y}_j - y_j)^2$$

给定学习率  $\eta$ ，有：

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$$

由链式法则可得：

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{ij}}$$

人工智能学院



西安电子科技大学  
XIDIAN UNIVERSITY

# 反向传播算法

## ▶ 反向传播算法的简单描述

$$\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}} \quad w_{kj} = w_{kj} + \Delta w_{kj} \quad \frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{kj}}$$

$$E = \frac{1}{2} \sum_{j=1}^J (\hat{y}_j - y_j)^2 \quad y_j = f(\beta_j + b_j) \quad \beta_j = \sum_{k=1}^P w_{kj} h_k$$

$$\frac{\partial E}{\partial y_j} = -(\hat{y}_j - y_j) \quad \frac{\partial \beta_j}{\partial w_{kj}} = h_k$$

$$\frac{\partial y_j}{\partial \beta_j} = f(\beta_j + b_j) \cdot (1 - f(\beta_j + b_j)) = y_j(1 - y_j) \quad \text{Sigmoid 函数}$$

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{kj}} = -(\hat{y}_j - y_j) \cdot y_j \cdot (1 - y_j) \cdot h_k$$

$$g_j = (\hat{y}_j - y_j) \cdot y_j \cdot (1 - y_j)$$

$$\frac{\partial E}{\partial w_{kj}} = -g_j \cdot h_k$$

$$\boxed{\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}} = \eta \cdot g_j \cdot h_k}$$

人工智能学院


西安电子科技大学  
XIDIAN UNIVERSITY

# 反向传播算法

$$\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}} = \eta \cdot g_j \cdot h_k$$

$$g_j = (\hat{y}_j - y_j) \cdot y_j \cdot (1 - y_j)$$

类似可得：

$$\Delta b_j = -\eta \frac{\partial E}{\partial b_j} \quad b_j = b_j + \Delta b_j \quad \frac{\partial E}{\partial b_j} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial b_j}$$

$$\frac{\partial E}{\partial y_j} = -(\hat{y}_j - y_j) \quad \frac{\partial y_j}{\partial b_j} = f(\beta_i + b_j) \cdot (1 - f(\beta_i + b_j)) = y_j(1 - y_j)$$

$$\frac{\partial E}{\partial b_j} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial b_j} = -(\hat{y}_j - y_j) \cdot y_j \cdot (1 - y_j) = -g_j$$

$$\Delta b_j = -\eta \frac{\partial E}{\partial b_j} = \eta \cdot g_j$$

人工智能学院

**反向传播算法**

西电  
XIDIAN UNIVERSITY

➤ 反向传播算法的简单描述

$$\Delta v_{ik} = -\eta \frac{\partial E}{\partial v_{ik}} \quad v_{ik} = v_{ik} + \Delta v_{ik} \quad \frac{\partial E}{\partial v_{ik}} = \frac{\partial E}{\partial h_k} \cdot \frac{\partial h_k}{\partial \alpha_k} \cdot \frac{\partial \alpha_k}{\partial v_{ik}}$$

$$E = \frac{1}{2} \sum_{j=1}^l (\hat{y}_j - y_j)^2 \quad y_j = f(\beta_j + b_j) \quad \beta_j = \sum_{k=1}^p w_{kj} h_k$$

$$h_k = f(\alpha_k + \gamma_k) \quad \alpha_k = \sum_{i=1}^d v_{ik} x_i$$

$$\frac{\partial E}{\partial h_k} = \sum_{j=1}^l \frac{\partial E}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial h_k} = -\sum_{j=1}^l g_j w_{kj}$$

$$\frac{\partial h_k}{\partial \alpha_k} = f(\alpha_k + \gamma_k) - (1 - f(\alpha_k + \gamma_k)) = h_k(1 - h_k) \quad \text{Sigmoid 函数}$$

$$\frac{\partial \alpha_k}{\partial v_{ik}} = x_i$$

人工智能学院

**反向传播算法**

西电  
XIDIAN UNIVERSITY

➤ 反向传播算法的简单描述

$$\frac{\partial E}{\partial v_{ik}} = \frac{\partial E}{\partial h_k} \cdot \frac{\partial h_k}{\partial \alpha_k} \cdot \frac{\partial \alpha_k}{\partial v_{ik}} = -h_k(1 - h_k) \cdot x_i \cdot \sum_{j=1}^l g_j w_{kj}$$

$$e_k = h_k(1 - h_k) \cdot \sum_{j=1}^l g_j w_{kj}$$

$$\frac{\partial E}{\partial h_k} = \frac{\partial E}{\partial \alpha_k} \cdot \frac{\partial \alpha_k}{\partial h_k} = -e_k \cdot x_i$$

$$\Delta v_{ik} = -\eta \frac{\partial E}{\partial v_{ik}} = \eta \cdot e_k \cdot x_i$$

$$v_{ik} = v_{ik} + \Delta v_{ik} = v_{ik} + \eta \cdot e_k \cdot x_i$$

人工智能学院

**反向传播算法**

西电  
XIDIAN UNIVERSITY

➤ 反向传播算法的简单描述

$$\Delta \gamma_k = -\eta \frac{\partial E}{\partial \gamma_k} \quad \gamma_k = \gamma_k + \Delta \gamma_k \quad \frac{\partial E}{\partial \gamma_k} = \frac{\partial E}{\partial h_k} \cdot \frac{\partial h_k}{\partial \gamma_k}$$

$$E = \frac{1}{2} \sum_{j=1}^l (\hat{y}_j - y_j)^2 \quad y_j = f(\beta_j + b_j) \quad \beta_j = \sum_{k=1}^p w_{kj} h_k$$

$$h_k = f(\alpha_k + \gamma_k) \quad \alpha_k = \sum_{i=1}^d v_{ik} x_i$$

$$\frac{\partial E}{\partial h_k} = \sum_{j=1}^l \frac{\partial E}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial h_k} = -\sum_{j=1}^l g_j w_{kj}$$

$$\frac{\partial h_k}{\partial \gamma_k} = f(\alpha_k + \gamma_k) - (1 - f(\alpha_k + \gamma_k)) = h_k(1 - h_k) \quad \text{Sigmoid 函数}$$

$$\frac{\partial E}{\partial \gamma_k} = \frac{\partial E}{\partial h_k} \cdot \frac{\partial h_k}{\partial \gamma_k} = -h_k(1 - h_k) \cdot \sum_{j=1}^l g_j w_{kj} = -e_k$$

$$\Delta \gamma_k = -\eta \frac{\partial E}{\partial \gamma_k} = \eta \cdot e_k$$

人工智能学院

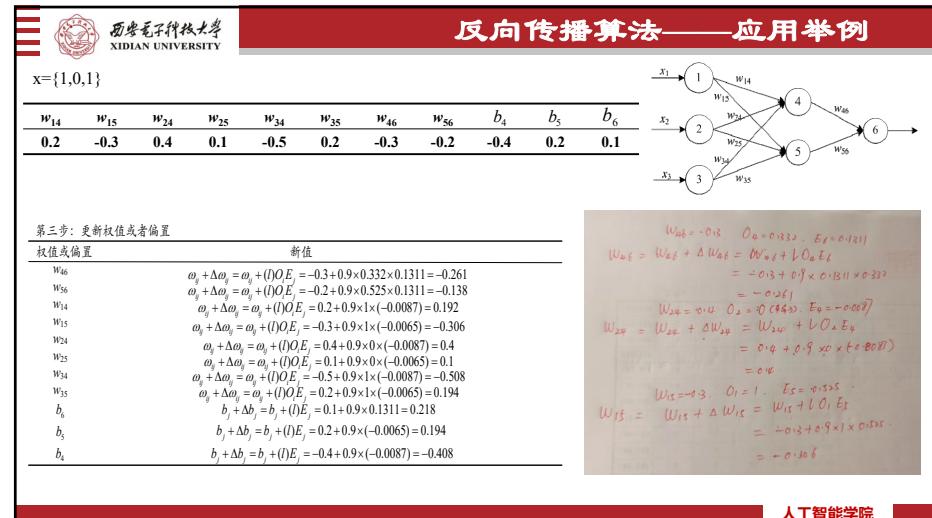
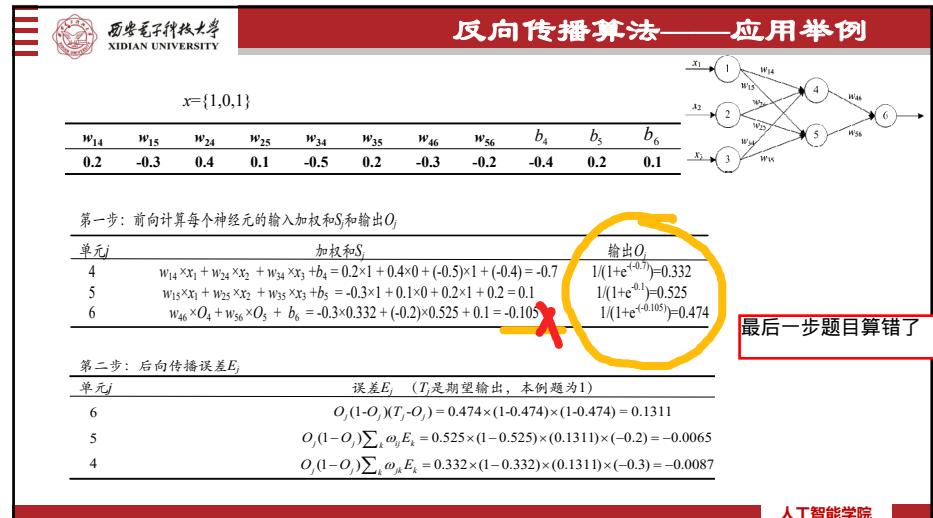
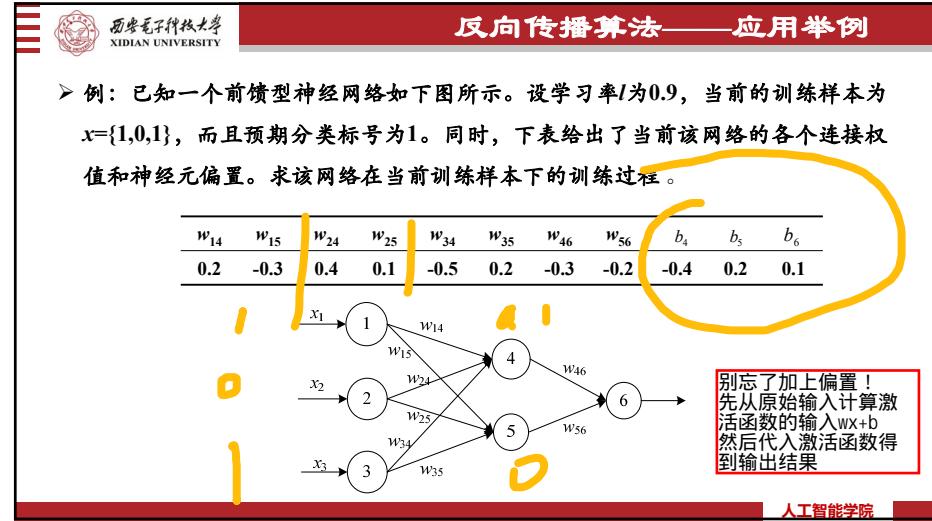
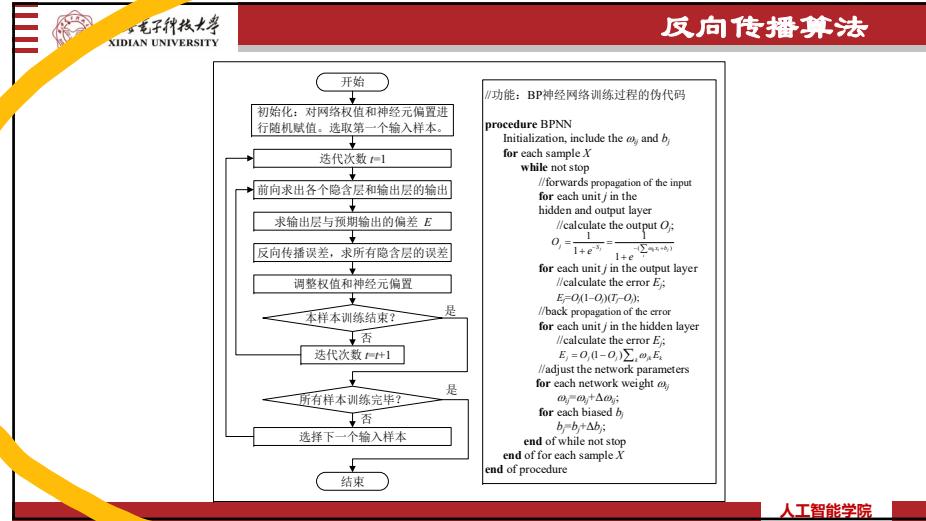
**反向传播算法**

西电  
XIDIAN UNIVERSITY

➤ 反向传播算法的学习过程

- ① 选择一组训练样本，每一个样本由输入信息和期望的输出结果两部分组成。
- ② 从训练样本集中取一样本，把输入信息输入到网络中。
- ③ 分别计算经神经元处理后的各层结点的输出。
- ④ 计算网络的实际输出和期望输出的误差。
- ⑤ 从输出层反向计算到第一个隐层，并按照某种能使误差向减小方向发展的原则，调整网络中各种神经元的连接权值。
- ⑥ 对训练样本集中的每一个样本重复③-⑤的步骤，直到对整个训练样本集的误差达到要求时为止。

人工智能学院



$$\begin{aligned}
 w_{46} &= -0.18 & O_6 &= 0.1332 & E_6 &= 0.1311 \\
 W_{46} &= W_{46} + \Delta W_{46} = W_{46} + l O_6 E_6 \\
 &= -0.3 + 0.9 \times 0.332 \times 0.1311 \\
 &= -0.261 \\
 W_{56} &= -0.12 & O_5 &= 0.525 & E_5 &= -0.0065 \\
 W_{56} &= W_{56} + \Delta W_{56} = W_{56} + l O_5 E_5 \\
 &= 0.4 + 0.9 \times 0.525 \times 0.1311 \\
 &= 0.408 \\
 W_{14} &= -0.3 & O_4 &= 0.332 & E_4 &= 0.1311 \\
 W_{14} &= W_{14} + \Delta W_{14} = W_{14} + l O_4 E_4 \\
 &= -0.3 + 0.9 \times 1 \times (-0.0087) \\
 &= -0.306 \\
 W_{15} &= -0.2 & O_5 &= 0.1 & E_5 &= -0.0065 \\
 W_{15} &= W_{15} + \Delta W_{15} = W_{15} + l O_5 E_5 \\
 &= -0.2 + 0.9 \times 0.1 \times (-0.0065) \\
 &= -0.138
 \end{aligned}$$

 西安电子科技大学  
XIDIAN UNIVERSITY

## 反向传播算法

### ➤ 反向传播算法优缺点

- 优点:
  - 理论基础牢固，推导过程严谨，物理概念清晰，通用性好等。所以，它是目前用来训练前馈多层网络较好的算法。
- 缺点:
  - BP算法的收敛速度一般来说比较慢；
  - BP算法只能收敛于局部最优解，不能保证收敛于全局最优解；
  - 当隐层元的数量足够多时，网络对训练样本的识别率很高，但对测试样本的识别率有可能很差，即网络的泛化能力有可能较差。

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 神经网络的学习算法

神经网络学习算法

学习方法	有监督学习
	无监督学习
	再励学习

学习规则	Hebb学习规则
	Delta( $\delta$ )学习规则
	梯度下降学习规则

学习规则	Kohonen学习规则
	后向传播学习规则
	概率式学习规则
	竞争式学习规则

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 神经网络的学习算法



有监督学习



无监督学习



再励学习

在有监督的学习方式中，网络的输出和期望的输出（即教师信号）进行比较，然后根据两者之间的差异调整网络的权重，最终使差异变小。

在无监督的学习方式中，输入模式进入网络后，网络按照一预先设定的规则（如竞争规则）自动调整权重，使网络最终具有模式分类等功能。

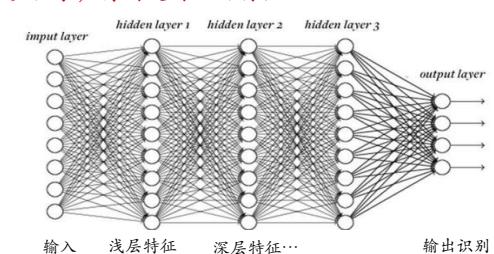
强化学习是介于上述两者之间的一种学习方法，它不需要老师给出目标输出。强化学习算法采用一个“评论员”来评价与给定输入相对应的神经网络输出的优度（质量因数）。

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 神经网络的隐层到底在学什么？

有了反向传播算法后，神经网络变得越来越深，并且随着神经网络隐藏层的增加，神经网络的表示能力也越来越强。一种通俗的理解是：隐藏层的神经元会提取输入中的特征，并且随着神经网络层数的增加，深层网络会将浅层网络中获得的特征进一步抽象，得到更高级的特征。



input layer      hidden layer 1      hidden layer 2      hidden layer 3      output layer

输入      浅层特征      深层特征...      输出识别

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 神经网络的隐层可视化

2012年多伦多大学的Krizhevsky等构建了一个超大型卷积神经网络，该网络有9层，65万个神经元，6千万个参数。网络的输入是图片，输出是1000个类，对应图片中物体的类别，如小虫、美洲豹。救生船等等。随后纽约大学的Zeiler和Fergus对这个训练好的网络进行了可视化，他们将网络中的某些神经元挑选出来，将其与输入图像进行对比，发现中间层的神经元响应了某些十分抽象的特征。

	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
										

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 神经网络的隐层可视化



	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
										

第一层神经元主要负责识别颜色和简单纹理。

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 神经网络的隐层可视化



第二层的一些神经元可以识别更加细化的纹理，比如布纹、刻度、叶纹。

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 神经网络的隐层可视化



第三层的一些神经元负责感受黑夜里的黄色烛光、鸡蛋黄、高光。

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 神经网络的隐层可视化



第四层的一些神经元负责识别萌狗的脸，七星瓢虫和一堆圆形物体的存在。

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 神经网络的隐层可视化

可以发现，输入在网络中的传递，实际上是一个特征提取的过程。浅层的网络提取出颜色、纹理等初级特征，深层的网络对这些简单特征进一步抽象，获得更高级的特征，最后输出层将这些高级特征映射到我们想要的输出，如所属类别、预测结果、生成样本等。

人工智能学院

 西安电子科技大学  
XIDIAN UNIVERSITY

## 神经网络的应用

- 民用应用领域
  - 语音识别、图像识别与理解、计算机视觉、智能机器人故障检测、实时语言翻译、企业管理、市场分析、决策优化、物资调运、自适应控制、专家系统、智能接口、神经生理学、心理学和认知科学研究等等；
- 军用应用领域
  - 语音、图像信息的录取与处理、雷达、声纳的多目标识别与跟踪、战场管理和决策支持系统、军用机器人控制、各种情况、信息的快速录取、分类与查询、导弹的智能引导、保密通讯、航天器的姿态控制等等；



人工智能学院