机器人与人工智能

--学习

刘之涛

浙江大学智能系统与控制研究所

学习

对于未知环境来说学习是必须的，因为设计者不是无所不知的

学习是一种很有用的系统构造方法，让智能体自己去揭露现实胜于事先告诉它

学习对智能体的决策机制进行改进来改善性能

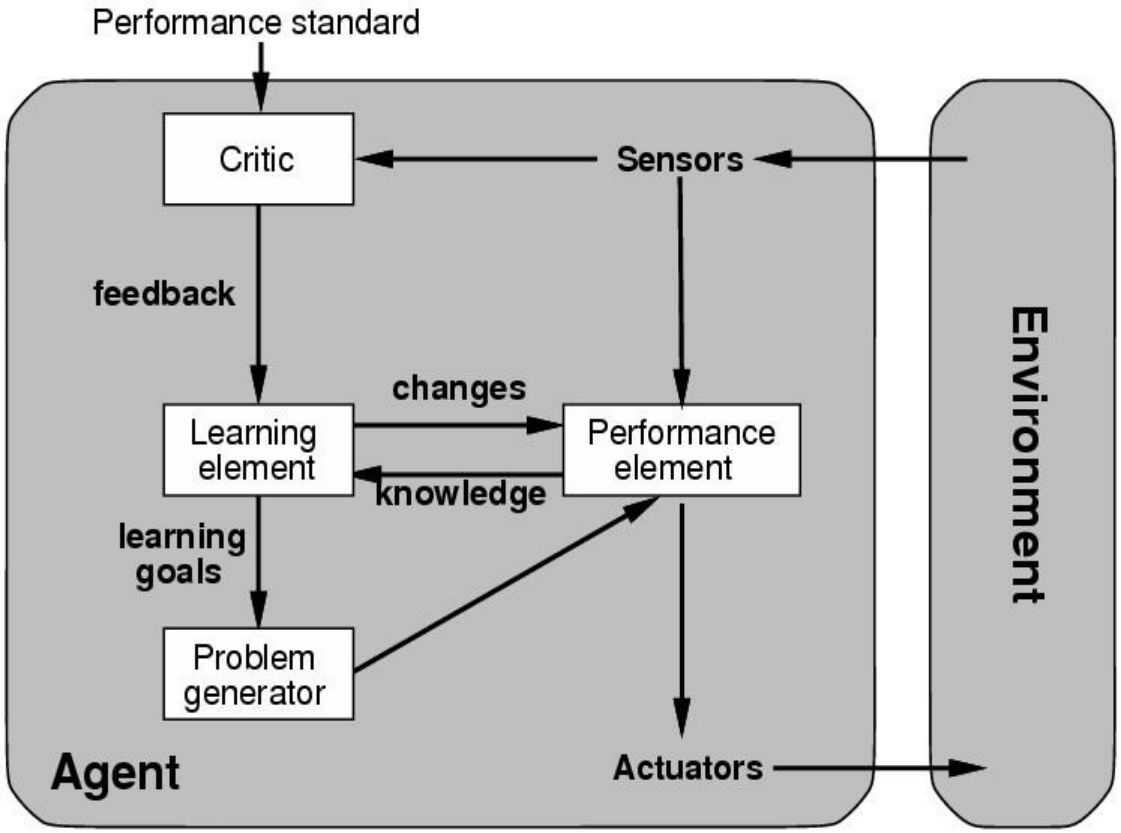
学习智能体

Performance element：根据感知选择行为

Learning element： 用来改进执行组件

Critic： 用来对主体进行评判的一种固定标准（单单依赖感知学习组件无法知道主体的好坏；如果不固定，无法进行学习。）

Problem Generator： 为了得到更多的经验或者数据，而对执行组件提出尝试的要求（否则执行组件总是按照自认的最优方案动作）



学习组件

学习组件设计与以下各项有关

- **学习执行组件中的哪个函数?**
- **这个函数如何表达的?**
- **有怎样的先验知识**
- **可以得到什么反馈?**

举例:

执行组件	函数	表达	反馈
Alpha-beta搜索	评估函数	加权线性函数	赢/输
简单反射主体	感知—行为函数	神经网络	正确行为

有监督学习：对每个实例返回正确答案

强化学习：对行为的评估反馈

无监督学习：没有明确的输出情况反馈

学习概述

机器学习的研究大致可以分为三个阶段：

- 五六十年代的探索阶段：

  主要受神经生理学、生理学和生物学的影响，研究主要侧重于非符号的神经元模型的研究，主要研制通用学习系统，即神经网络或自组织系统。

  主要成果有：

  感知机（Perceptron）

  Friedberg等模拟随机突变和自然选择过程的程序，

  Hunt等的决策树归纳程序CLS。

学习概述

- 七十年代的发展阶段：

由于当时专家系统的蓬勃发展，知识获取成为当务之急，这给机器学习带来了契机，主要侧重于符号学习的研究。机器学习的研究脱离了基于统计的以优化理论为基础的研究方法，提出了基于符号运算为基础的机器学习方法，并产生了许多相关的学习系统， 主要系统和算法包括：

Winston的积木世界学习系统；

Michalski基于逻辑的归纳学习系统AQVAL；

Michalski和Chilausky的AQ11；

Quinlan的ID3程序

Mitchell的版本空间方法。

学习概述

- 八九十年代至今的鼎盛阶段。

理论研究和应用研究也有了新的突破，机器学习的研究进入了全面的、系统化的时期。 主要成果有：

一方面传统的符号学习的各种方法已日臻完善。Michalski等将AQ11扩充为一个多功能学习系统AQ15，ID3算法中使用了熵，从而使决策树归纳得到了很大的改进。

科学发现系统BACON开辟了无导师学习的两个重要研究领域。

神经网络学习在消沉了一段时期后又重新蓬勃发展起来了，同时计算机硬件技术的高速发展也为开展大规模和高性能的人工神经网络提供了保障，使得基于神经网络的连接学习从低谷走出，发展迅猛。其中Rumelhart等人提出的BP模型，提供了一个训练多层网络的实际可行的方法，克服了Perceptron的大部分局限性。

学习概述

- 另一方面，机器学习的基础理论的研究越来越引起人们的重视。

1984年美国学者Valiant提出了基于概率近似正确性的学习理论（PAC学习），对布尔函数的一些特殊子类的可学习性进行了探讨，将可学习性与计算复杂性联系在一起，并由此派生出了“计算学习理论”（COLT）

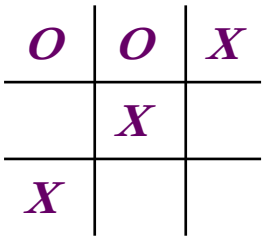
我国学者洪家荣教授证明了两类布尔表达式：析取范式和合取范式都是PAC不可学习的，揭示了PAC方法的局限性

1995年，Vapnik出版了“统计学习理论”一书。

对PAC的研究是一种理论性，存在性的；Vapnik的研究却是构造性的，他将这类研究模型称为支持向量机SVM（Support Vector Machine）。

有监督学习

根据样本学习一个未知的**目标函数** f，**样本**是数据对$(x, f(x))$，

例如 $(+1,$  $)$

问题：给定一个包含N个样本的**训练集**，寻找一个**假设** h 满足 $h \approx f$

这是真实学习的一种高度简化模型：

- 忽略先验知识
- 处于一个确定的，可观测的“环境”中
- 样本给定
- 智能体希望学习 f

分类：$f(x)$是有限种取值

回归：$f(x)$是无限的数值

无监督学习

- 监督算法和无监督算法之间的区别没有规范严格的定义，因为没有客观的判断来区分监督者提供的值是特征还是目标。
- 无监督学习：是指从不需要认为注释的样本的分布中抽取信息。通常与密度估计有关，学习从分布中采样、学习从分布中去噪、寻找数据分布的流形或是将数据中相关的样本聚类。

监督学习与无监督学习对比

- 在监督学习中，给定一组数据，我们知道正确的输出结果应该是什么样子，并且知道在输入和输出之间有着一个特定的关系。

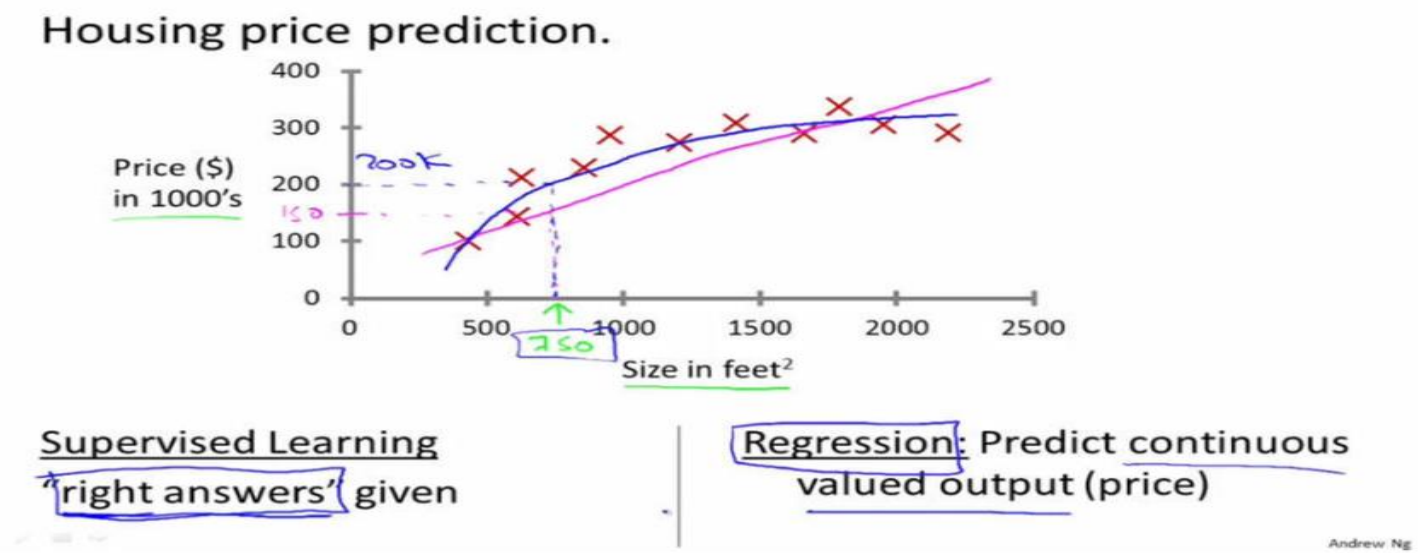
$$\text{监督学习}：\begin{cases}\text{回归}(Regression)\\ \text{分类}(Classification)\end{cases}$$

- 在回归问题中，我们会预测一个连续值。也就是说我们试图将输入变量和输出用一个连续函数对应起来；而在分类问题中，我们会预测一个离散值，我们试图将输入变量与离散的类别对应起来

监督学习举例

- 回归

通过房地产市场的数据，预测一个给定面积的房屋的价格就是一个回归问题。这里我们可以把价格看成是面积的函数，它是一个连续的输出值。 但是，当把上面的问题改为“预测一个给定面积的房屋的价格是否比一个特定的价格高或者低”的时候，这就变成了一个分类问题, 因为此时的输出是‘高’或者‘低’两个离散的值。

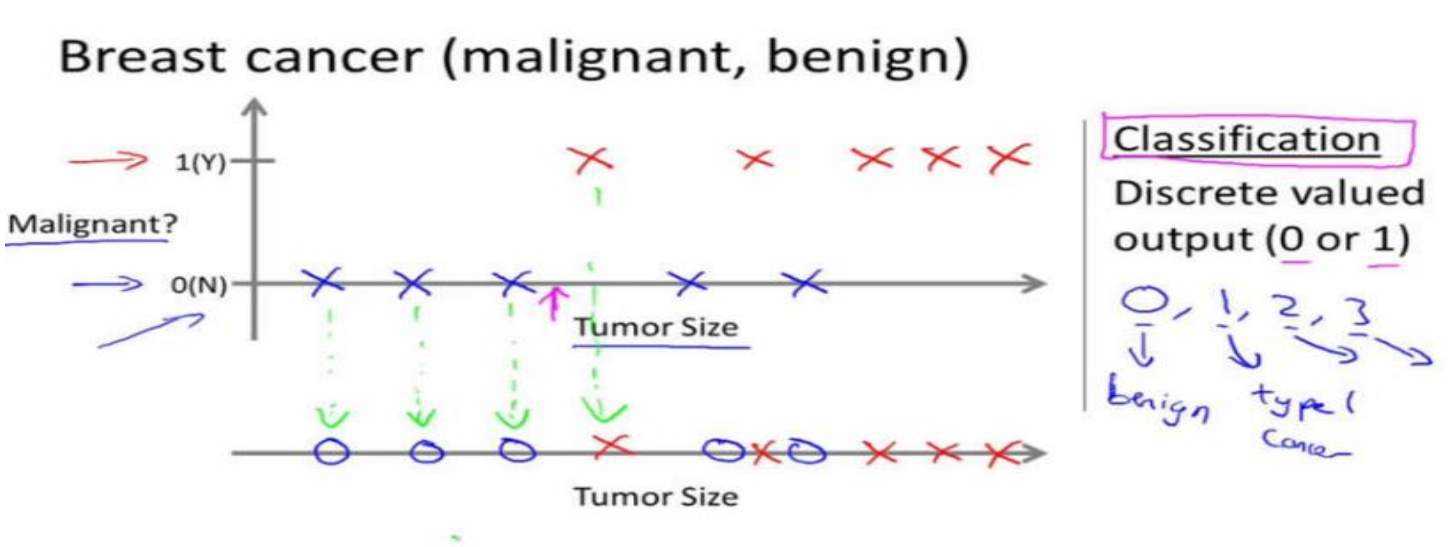


监督学习举例

- **分类：** 给定医学数据，通过肿瘤的大小来预测该肿瘤是恶性瘤还是良性瘤(课程中给的是乳腺癌的例子)，这就是一个分类问题，它的输出是0或者1两个离散的值。(0代表良性，1代表恶性)。

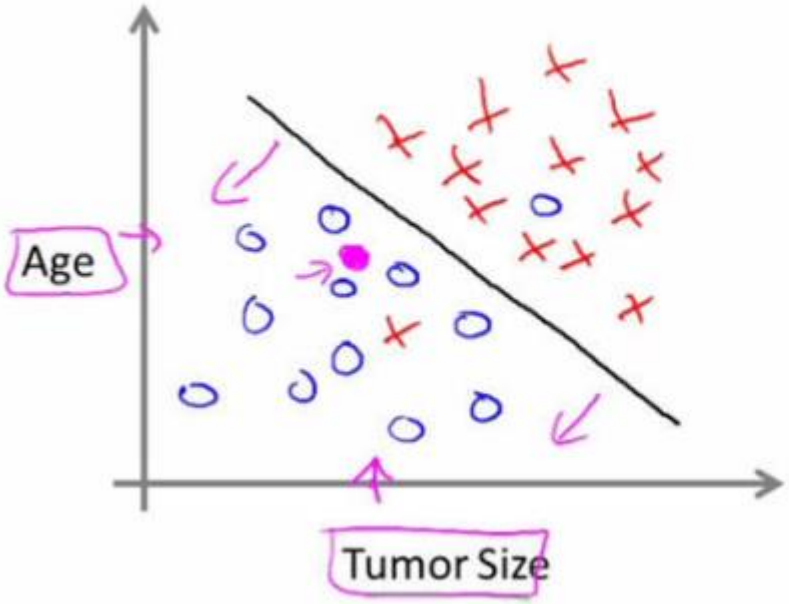
分类问题的输出可以多于两个，比如在该例子中可以有{0,1,2,3}四种输出，分别对应{良性, 第一类肿瘤, 第二类肿瘤, 第三类肿瘤}。

下图中上下两个图只是两种画法。第一个是有两个轴，Y轴表示是否是恶性瘤，X轴表示瘤的大小; 第二个是只用一个轴，但是用了不同的标记，用O表示良性瘤，X表示恶性瘤。



监督学习举例

- 有时候也有两个或者多个特征。例如下图， 有“年龄”和“肿瘤大小”两个特征。(还可以有其他许多特征，如下图右侧所示)

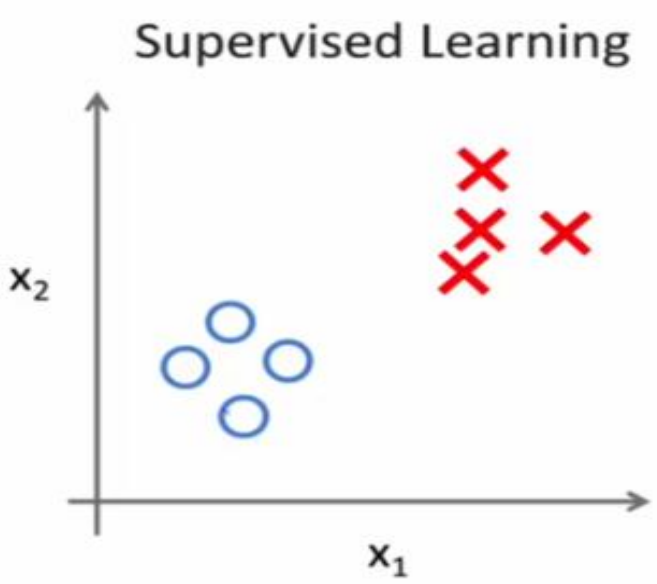


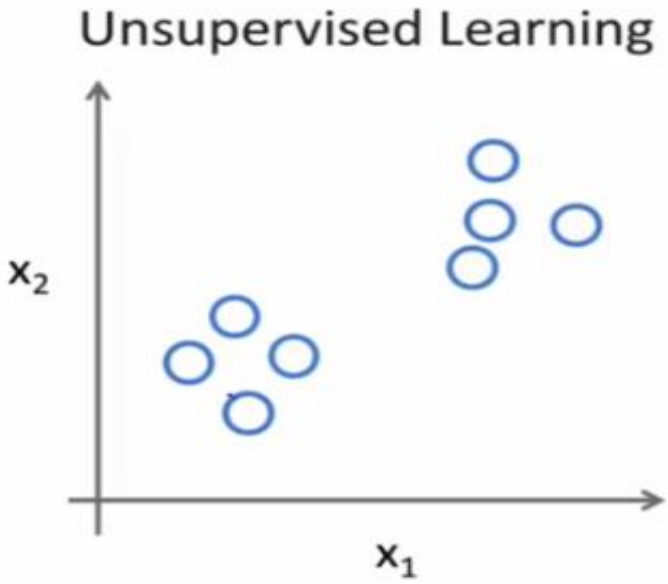
- Clump Thickness
- Uniformity of Cell Size
- Uniformity of Cell Shape

...

无监督学习

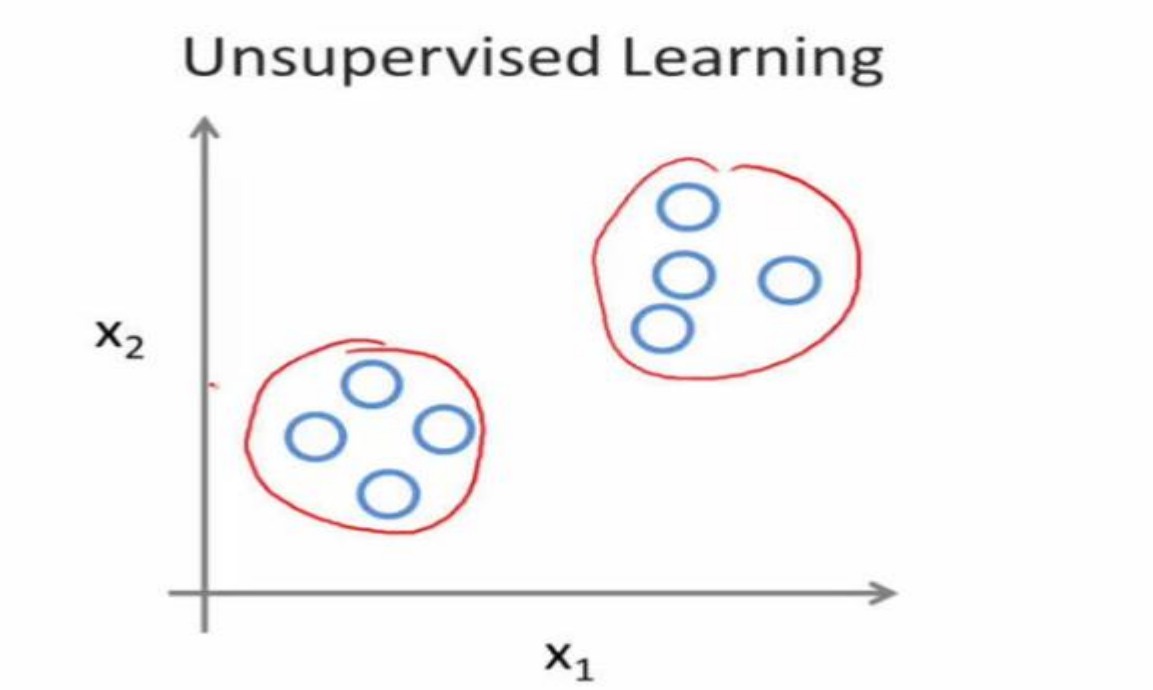
- 在无监督学习中，我们基本上不知道结果会是什么样子，但我们可以通过聚类的方式从数据中提取一个特殊的结构。在无监督学习中给定的数据是和监督学习中给定的数据是不一样的。在无监督学习中给定的数据没有任何标签或者说只有同一种标签。如下图所示





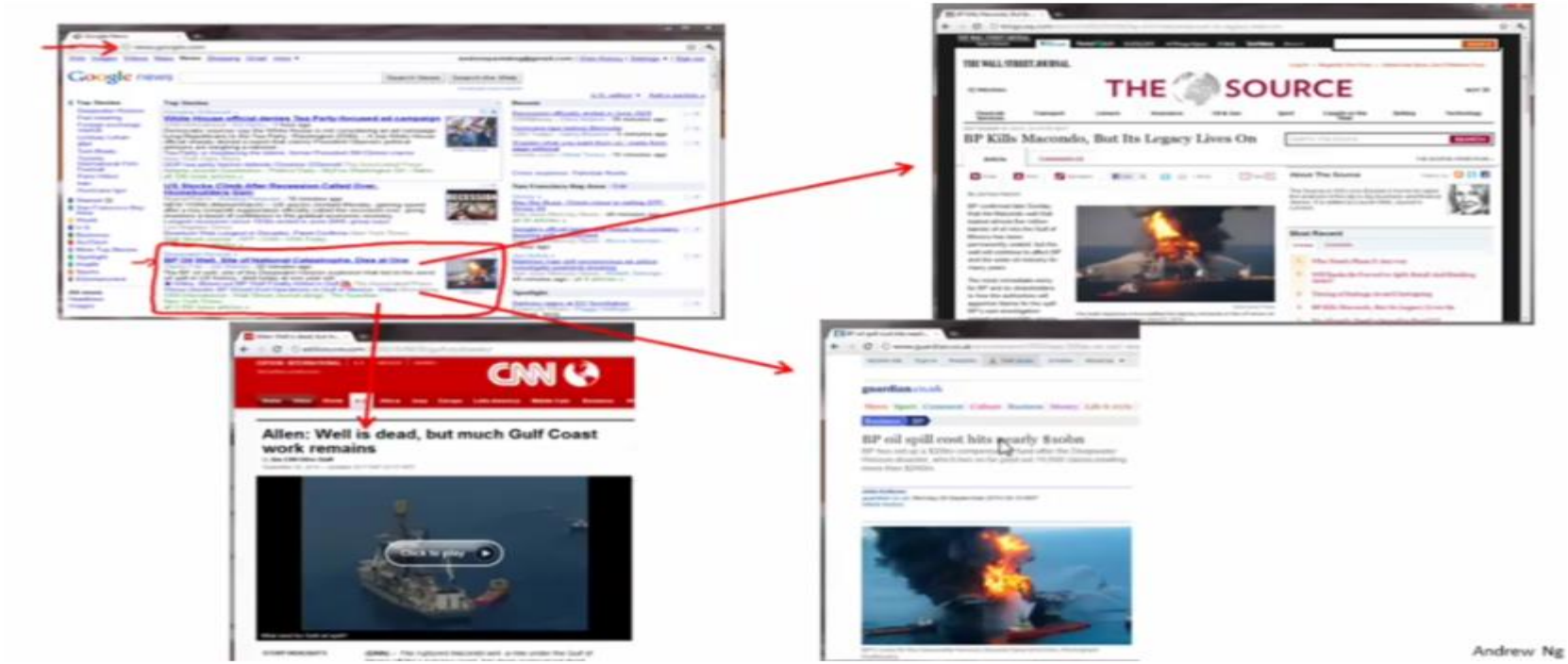
无监督学习

- 如下图所示，在无监督学习中，我们只是给定了一组数据，我们的目标是发现这组数据中的特殊结构。例如我们使用无监督学习算法会将这组数据分成两个不同的簇,，这样的算法就叫聚类算法。



无监督学习举例

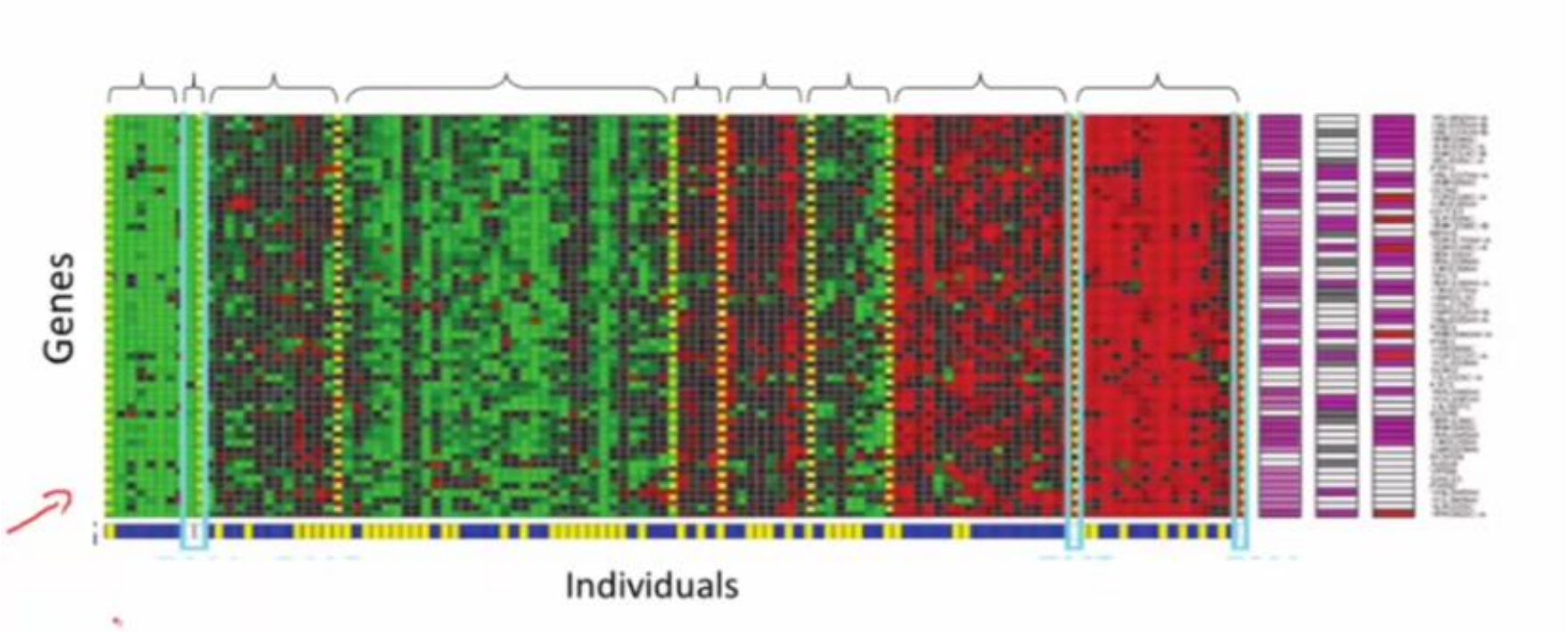
- 新闻分类： 第一个例子举的是Google News的例子。Google News搜集网上的新闻，并且根据新闻的主题将新闻分成许多簇，然后将在同一个簇的新闻放在一起。如图中红圈部分都是关于BP Oil Well各种新闻的链接，当打开各个新闻链接的时候，展现的都是关于BP Oil Well的新闻。



内容

- 根据给定基因将人群分类

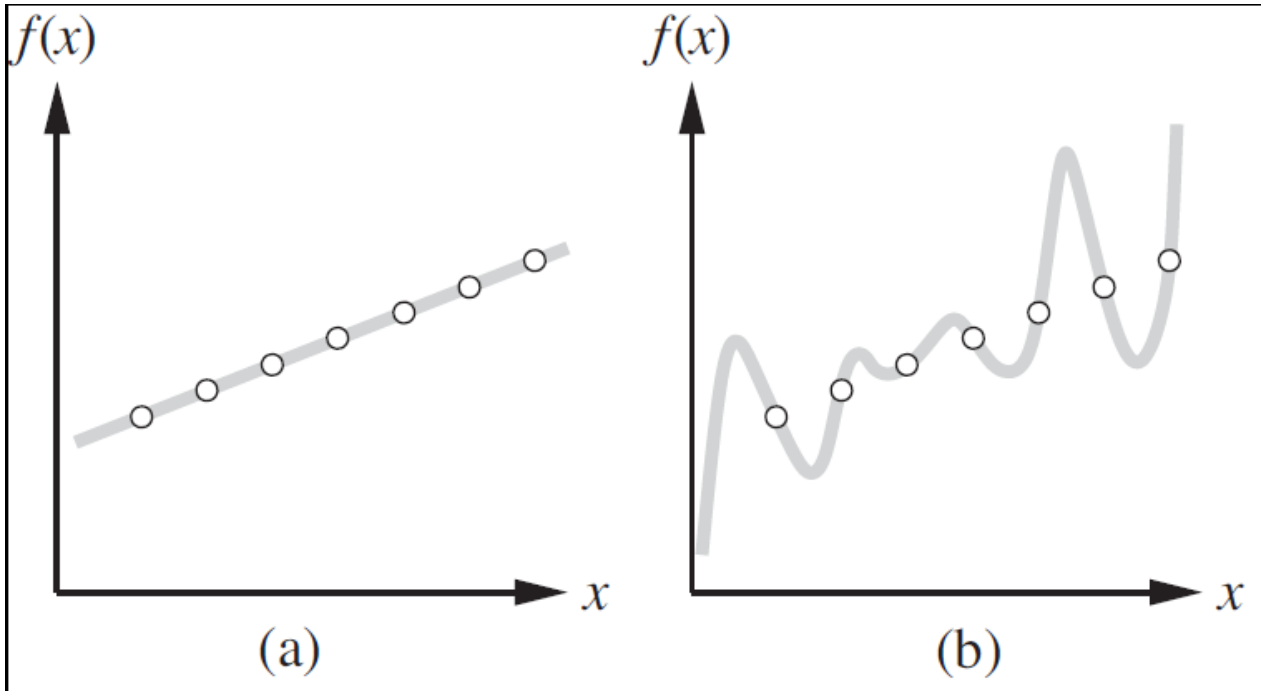
如图是DNA数据，对于一组不同的人我们测量他们DNA中对于一个特定基因的表达程度。然后根据测量结果可以用聚类算法将他们分成不同的类型。这就是一种无监督学习，因为我们只是给定了一些数据，而并不知道哪些是第一种类型的人，哪些是第二种类型的人等等。



归纳学习方法

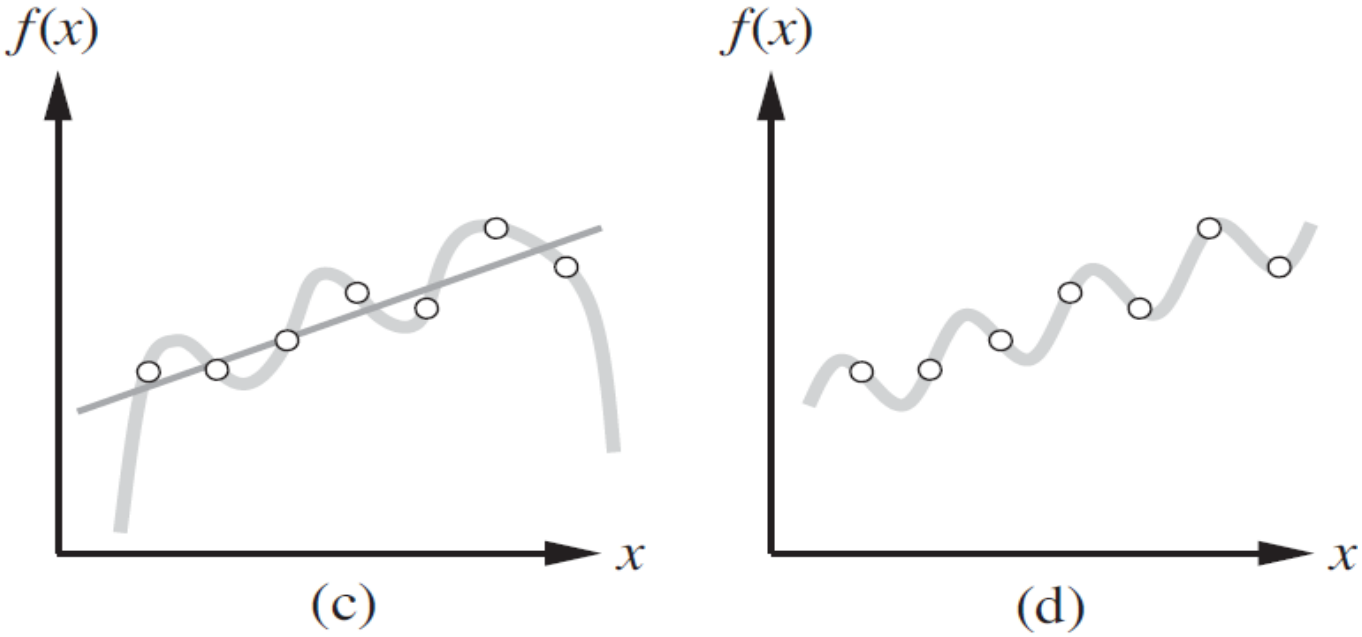
在假设空间H中选择假设h，使之在训练集上与f 相吻合

(如果h在所有的样本点上都与f 相吻合，则h是一致的)



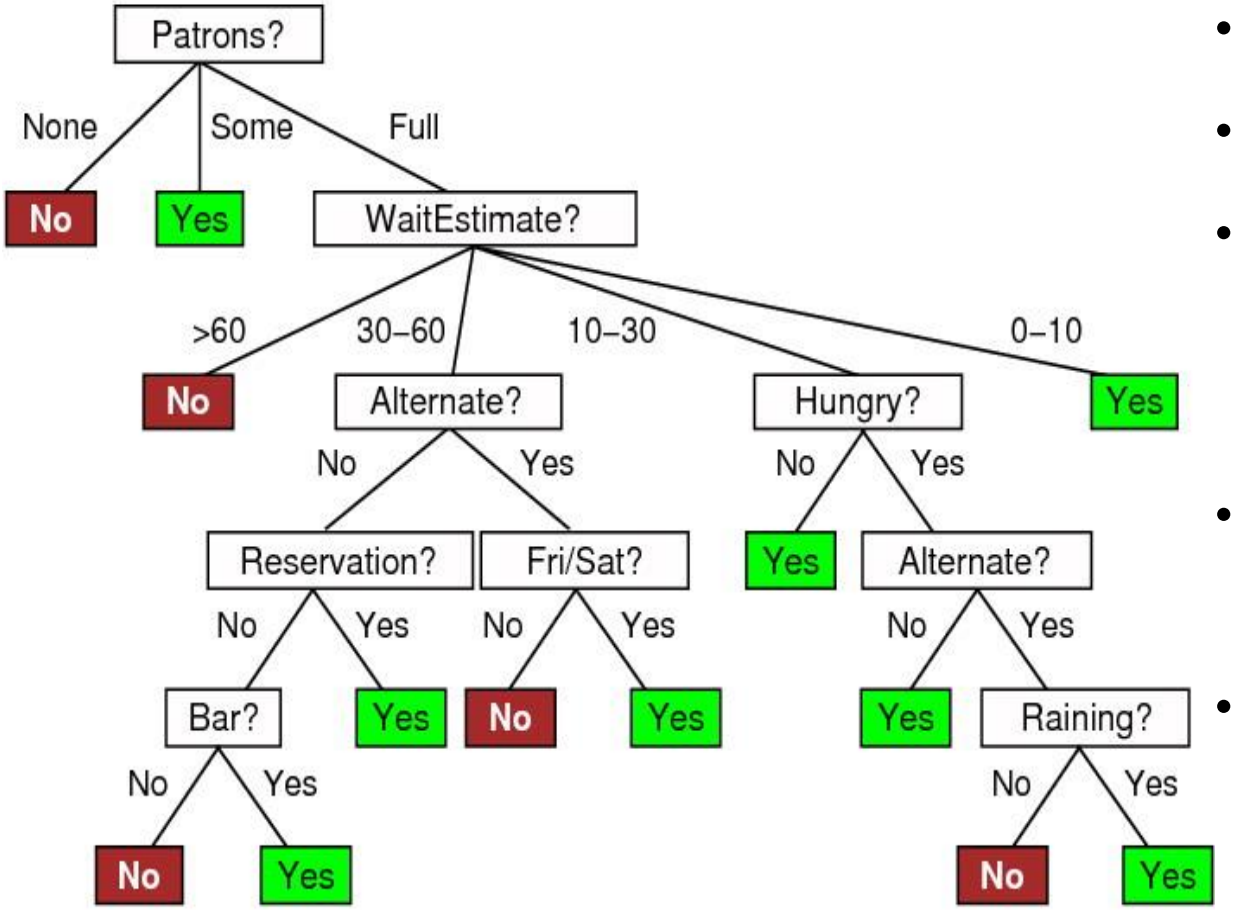
Ockham’s Razor 原则：优先选择与数据一致的最简单的假设！

归纳学习方法



- (c)中没有一致的直线，需要6阶多项式才能取得一致，但泛化能力很差
- 如果扩展假设空间包含多项式和sin(x)函数，则可得到(d)中的假设 ax+b+csin(x)
- 假设空间中包含目标函数则学习问题是可实现的。
- 需要在假设空间的表达能力和寻找合适假设的复杂性之间权衡

学习决策树



决定是否应该等待就餐的真实的决策树

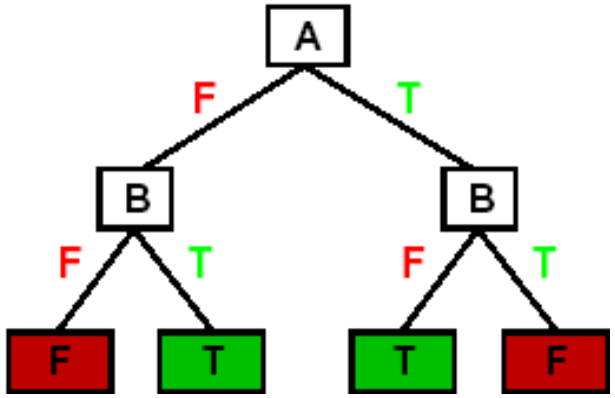
- 以一组属性作为输入
- 输出相应的决策
- 由节点构成，每个内部节点对应一个属性检验
- 节点出来的分支用属性的检验结果标注
- 叶节点对应指定的输出值,如果到达该叶节点，则返回它所对应的值

表达

决策树可以表达输入属性的任何函数。

如对于布尔函数，真值表中的行对应于决策树中从根节点到叶节点的路径

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



一般的，对任何训练集都存在一个一致的决策树，其中每个样本都有一条到叶节点的路径（除非f 在x处不确定），但这颗树可能不能正确分类新样本

假设空间

*n*个布尔属性可以构成多少种不同的决策树?

= 布尔函数的数量

= 具有2^n行的不同真值表数量 $= 2^{2^n}$

例如对于10个布尔属性存在

2^{1024} (约10^{308}) 种决策树

多少个只包含合取的假设（例如*Hungry*∧¬ *Rain*）?

每个属性可以为（T），为（F），或者没有

⇒3^n个不同的合取假设

随着假设空间表达能力的增强

- 包含目标函数的可能性增加
- 与训练集一致的假设的数量增加

=> 可能得到更差的预估器

举例：是否在餐厅等待

- 用属性值来描述样本（布尔量，离散量，连续量等）
  - Alt：附近是否有其它合适的餐厅
  - Bar： 餐厅是否有舒适的等待区
  - Fri： 是否是周五
  - Hun： 饿吗?
  - Pat： 餐厅里有多少人用餐（None，Some，Full)
  - Price： 餐厅的价格区间（$，$$，$$$)
  - Raining： 下雨吗?
  - Res： 有预定吗?
  - Type： 餐厅的类型（French，Italian， Thai， burger)
  - Est： 需要等待多久（0-10min，10-30,30-60，>60)

样本点

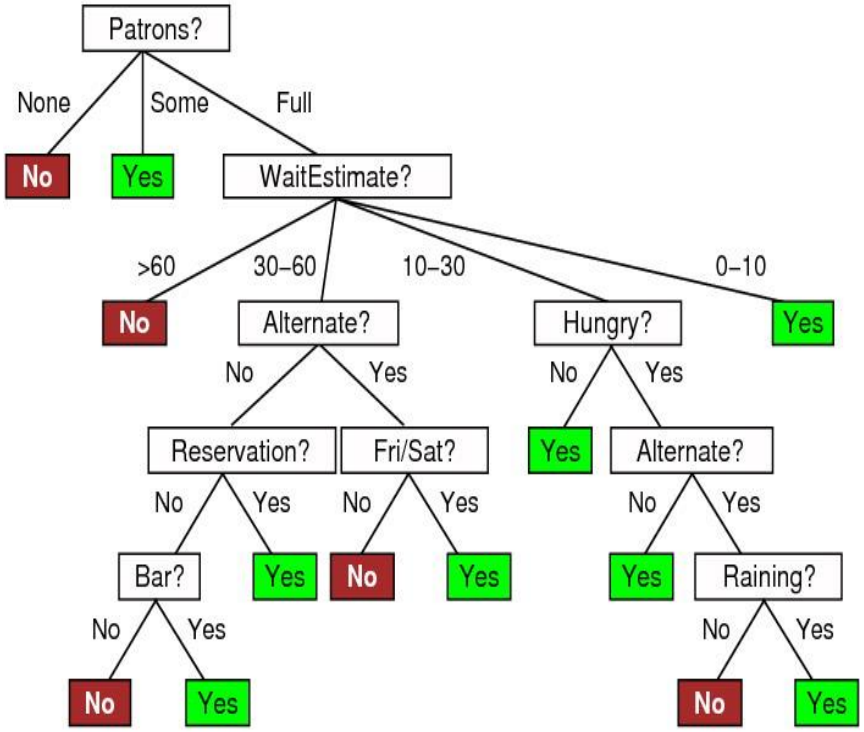
样本	属性										目标
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*是否等待*
X_1	*T*	*F*	*F*	*T*	*Some*	*$$$*	*F*	*T*	*French*	*0-10*	*T*
X_2	*T*	*F*	*F*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*30-60*	*F*
X_3	*F*	*T*	*F*	*F*	*Some*	*$*	*F*	*F*	*Burger*	*0-10*	*T*
X_4	*T*	*F*	*T*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*10-30*	*T*
X_5	*T*	*F*	*T*	*F*	*Full*	*$$$*	*F*	*T*	*French*	*>60*	*F*
X_6	*F*	*T*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Italian*	*0-10*	*T*
X_7	*F*	*T*	*F*	*F*	*None*	*$*	*T*	*F*	*Burger*	*0-10*	*F*
X_8	*F*	*F*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Thai*	*0-10*	*T*
X_9	*F*	*T*	*T*	*F*	*Full*	*$*	*T*	*F*	*Burger*	*>60*	*F*
X_{10}	*T*	*T*	*T*	*T*	*Full*	*$$$*	*F*	*T*	*Italian*	*10-30*	*F*
X_{11}	*F*	*F*	*F*	*F*	*None*	*$*	*F*	*F*	*Thai*	*0-10*	*F*
X_{12}	*T*	*T*	*T*	*T*	*Full*	*$*	*F*	*F*	*Burger*	*30-60*	*T*

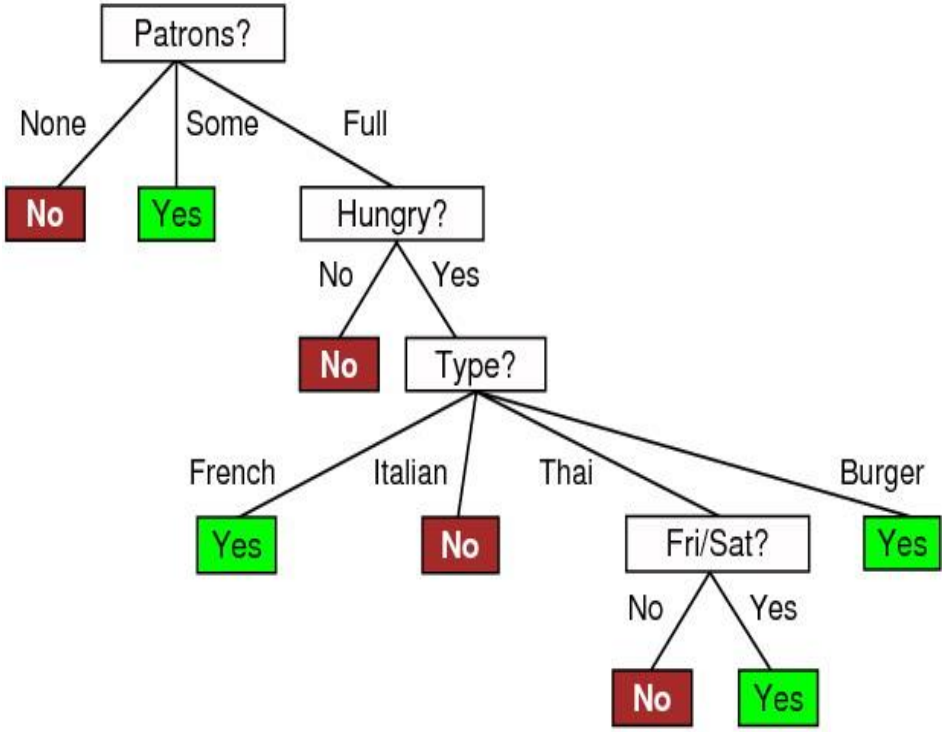
决策树学习

目标：寻找与训练样本一致的最小决策树

思想：（递归）选择“最重要的”属性作为（子）树的根节点

用12个样本学习得到的决策树：





比“真实的”树简单很多，数据量太少将不能正确识别一个复杂的函数

决策树学习算法

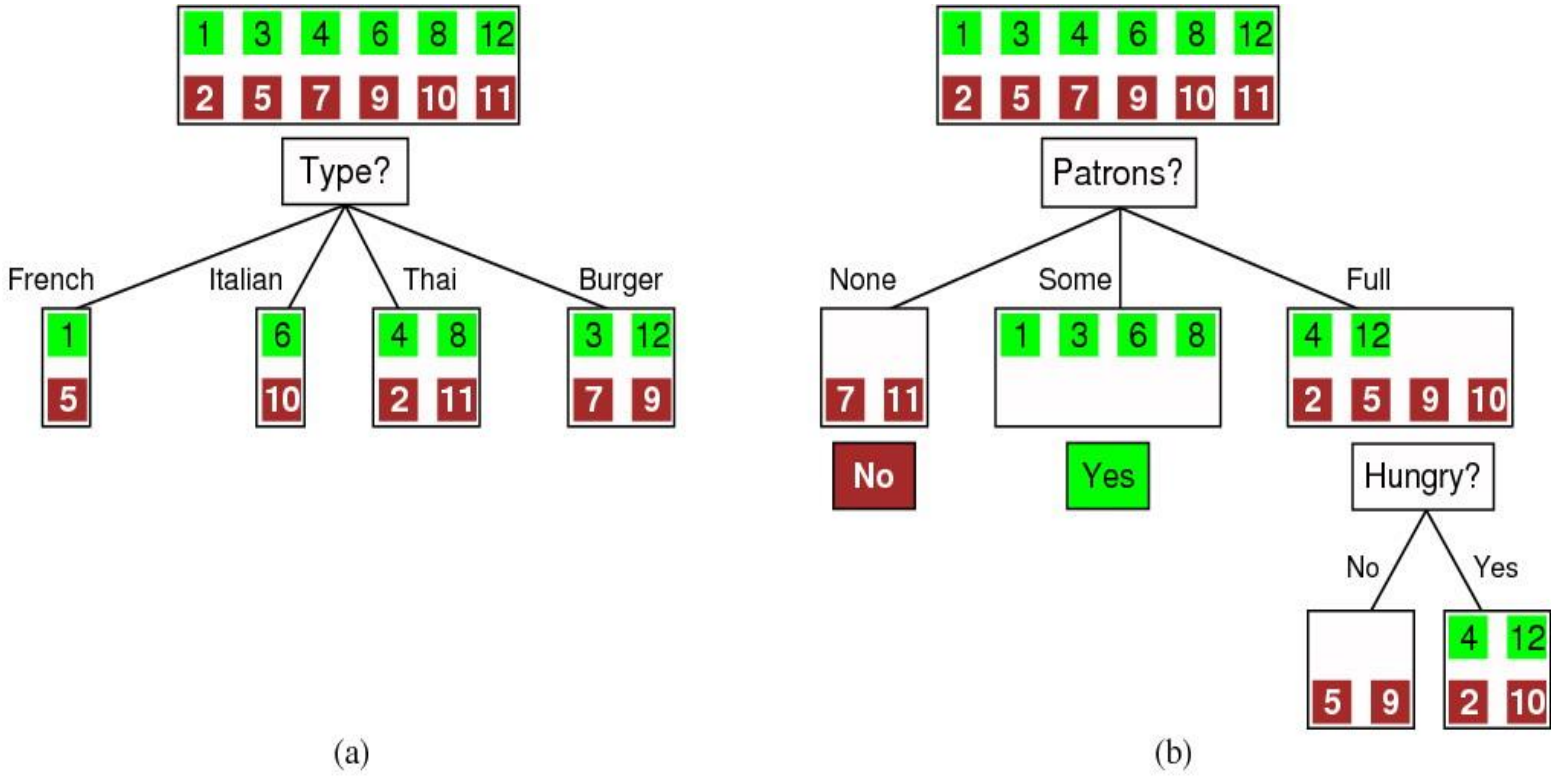
function DECISION-TREE-LEARNING($examples$, $attributes$, $parent\_examples$) **returns** tree

if $examples$ is empty **then return** PLURALITY-VALUE($parent\_examples$)
else if all $examples$ have the same classification **then return** the classification
else if $attributes$ is empty **then return** PLURALITY-VALUE($examples$)
else
  $A \leftarrow \text{argmax}_{a \in attributes}$ IMPORTANCE(a, $examples$)
  $tree \leftarrow$ a new decision tree with root test A
  for each value v_k of A **do**
    $exs \leftarrow \{e : e \in examples \text{ and } e.A = v_k\}$
    $subtree \leftarrow$ DECISION-TREE-LEARNING(exs, $attributes - A$, $examples$)
    add a branch to $tree$ with label $(A = v_k)$ and subtree $subtree$
  return $tree$

选择一个属性

思想： 一个好的属性是将样本分为“全正”或者“全负”子集

（理想状况）



Patrons? **是较好的选择 - 包含的关于分类的信息多**

信息量

信息提供答案

对答案知道的越少，对信息的需求就越大

记：1比特 = 回答先验概率为<0.5,0.5> 的布尔问题所需要的信息

当先验概率为<P_1,...,Pn>时答案所需要的信息为

$$I\left(\left\langle P_1,\ldots,P_n\right\rangle\right)=\sum_{i=1}^{n}-P_i\log_2 P_i$$

（也叫做先验概率的熵）

信息量

假设在根节点处有p个正样本，n 个负样本

=> 需要I(<p/(p+n), n/(p+n)>)比特信息来识别一个新样本

例如：对于12个餐厅样本，p=n=6 时需要1个比特的信息

一个属性将样本集E分为多个子集E_i，（我们希望）每个子集需要较少的信息就能完成分类

令E_i有p_i个正的和n_i个负的样本

=> 需要 $I\left(\left\langle p_i/(p_i+n_i), n_i/(p_i+n_i)\right\rangle\right)$ 比特信息来识别新样本

=>对于所有的分支，需要的信息量的期望是

$$\sum_i \frac{p_i+n_i}{p+n} I\left(\left\langle p_i/(p_i+n_i), n_i/(p_i+n_i)\right\rangle\right)$$

信息量

- 根据属性 A 将训练集 E 划分为子集E_1, … , E_v 其中 A 有 v 个不同的值.则测试完属性A后，还需要remainder(A)的信息量对实例进行分类。

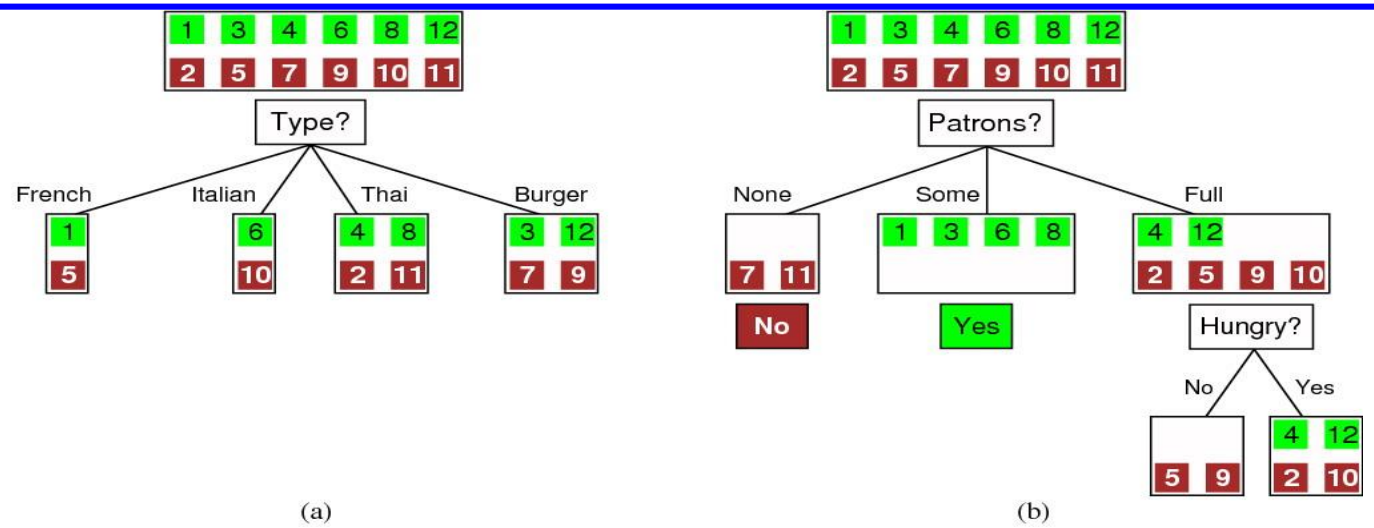
$$remainder(A)=\sum_{i=1}^{v}\frac{p_i+n_i}{p+n}I(\frac{p_i}{p_i+n_i},\frac{n_i}{p_i+n_i})$$

- 信息增益(IG) 是原始的信息需求和新的信息需求之间的差异:

$$IG(A)=I(\frac{p}{p+n},\frac{n}{p+n})-remainder(A)$$

- 选择增益最大的属性

信息量



对于Patrons?还需要0.459比特信息量：

$$\frac{2}{12}I\langle 0,1\rangle+\frac{4}{12}I\langle 1,0\rangle+\frac{6}{12}I\left\langle\frac{2}{6},\frac{4}{6}\right\rangle=0.459\text{bit}$$

Type仍然需要1比特信息量：

$$\frac{2}{12}I\left\langle\frac{1}{2},\frac{1}{2}\right\rangle+\frac{2}{12}I\left\langle\frac{1}{2},\frac{1}{2}\right\rangle+\frac{4}{12}I\left\langle\frac{1}{2},\frac{1}{2}\right\rangle+\frac{4}{12}I\left\langle\frac{2}{4},\frac{2}{4}\right\rangle=1\text{bit}$$

=> 选择对剩余信息量需求最小的属性

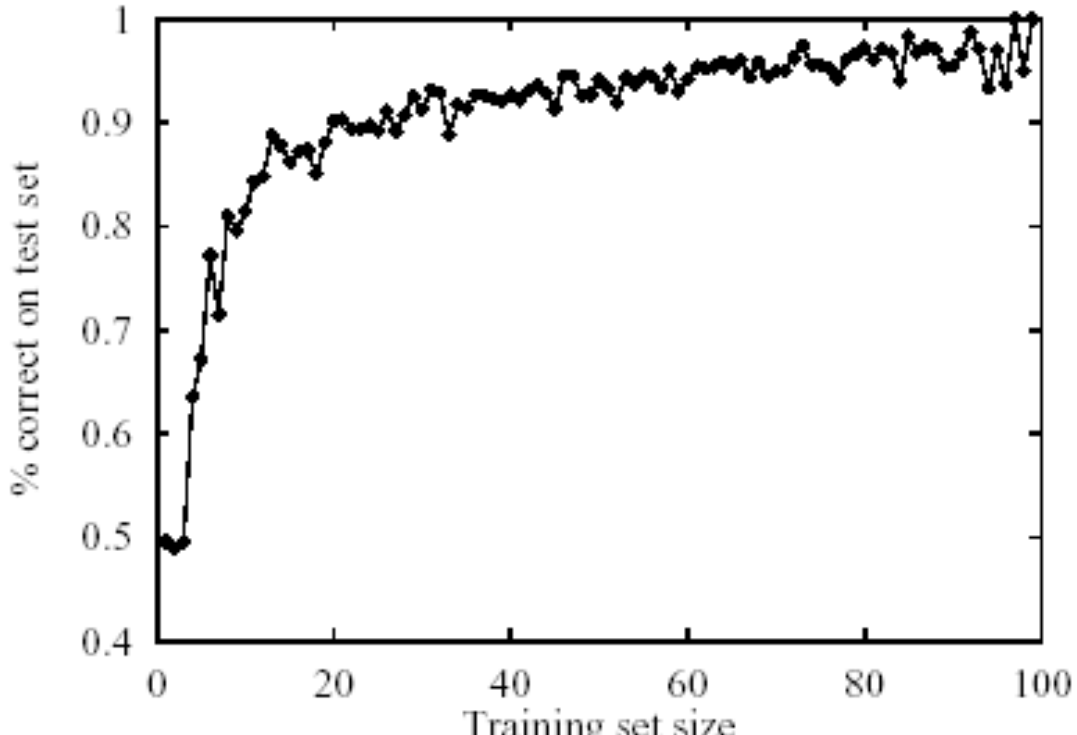
性能衡量

怎样知道h≈f？

1) 使用计算学习/统计学习理论中的定理

2) 用新的样本测试集对h进行测试（与训练样本的分布相同）

学习曲线 = 测试集上辨识的正确率和训练集大小之间的函数关系



过拟合与交叉检验

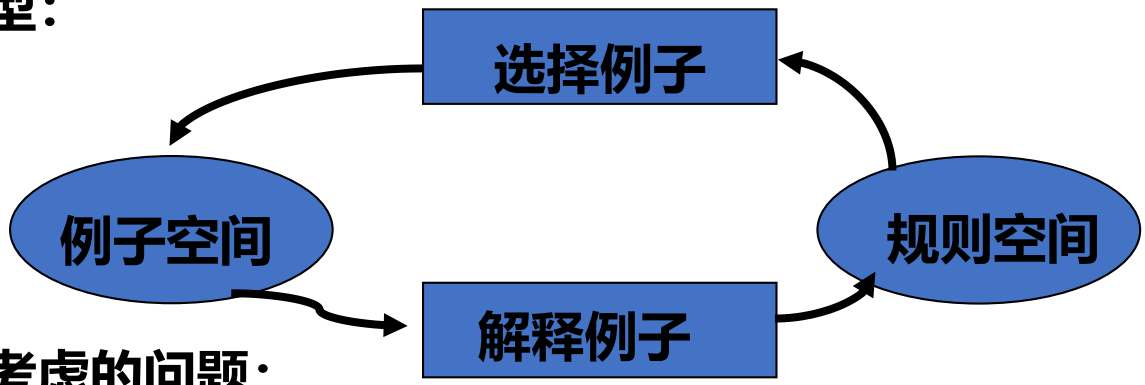
- 对于大规模的假设集或者存在噪声的情况，往往会发生过拟合的现象
- 可采用交叉检验的方法来减少过拟合，基本思想是估计每个假设对未知数据进行预测的准确程度。
- k-次交叉检验是进行k次实验，每次预留不同的1/k的数据用于检测，并对结果进行平均。

实例学习

- 基本思想：
  - 环境提供给系统一些特殊的实例，这些例子事先由施教者划分为正例和反例。 实例学习由此进行归纳推理，产生适用于更大范围的一般性知识，得到一般的规则 ，它将覆盖所有的正例并排除所有的反例。
  - 环境提供给学习环境的例子是低水平的信息，这是在特殊情况下执行环节的行为。学习环节归纳出的规则是高水平的信息，可以在一般情况下用这些规则指导执行环节的工作

实例学习

- 两个空间模型：



- 例子空间要考虑的问题：
  - 示教例子的质量
  - 例子空间的组织和搜索方法
- 规则空间要考虑的问题
  - 形成知识的归纳推理方法
  - 搜索规则空间的方法
  - 对规则空间的要求

实例学习

- 按规则空间搜索方法分类：
  - 数据驱动方法：
    - 变形空间方法：采用统一的形式表示规则和例子。
    - 改进假设方法：例子和规则的表示不统一。程序根据例子选择一种操作，用该操作修改H中的规则
  - 模型驱动方法：
    - 产生和测试方法：针对示教例子反复产生和测试假设的规则。利用基于模型的知识产生假设的规则，便于只产生可能合理的假设
    - 方案示例方法：使用规则方案的集合来限制可能合理的规则形式，最符合示教例子的规则被认为是最合理的规则

实例学习

- 按任务的复杂性划分为：
  - 学习单个概念：由系统提供的某个概念的正例和反例，只要求系统归纳出一个概念的描述规则
  - 学习多个概念：要求归纳出多个相互独立的概念
  - 学习执行多步任务：执行环节使用一个操作序列去完成任务，即执行环节进行任务规划。因此，归纳出的规则应该是进行任务规划的规则

变形空间法

- 基本思想：以整个规则空间为初始的假设规则集合H，根据示教例子中的信息，对集合H进行一般化或特殊化处理，逐步缩小集合H，最后使H收敛为只含要求的规则。
- 规则空间中的偏序关系：它是按一般性和特殊性来建立的一种概念之间的关系
- 排序后的变形空间：
  - 最上面：是最一般的 规则(概念)，是没有描述的点，所有的例子都符合这一概念
  - 最下面一行的各点：是示教正例对应的概念，每个点的概念只符合一个正例

变形空间法

- 假设规则的集合H：
  - H是规则空间的子集
  - H中最一般的元素组成的子集称为G集合
  - H中最特殊的元素组成的子集称为S集合
  - 在规则空间中，H是G和S中间的一段。
  - 可以用G和S来表示H
- 变形空间方法：
  - 初始：G是最上面一个点，S是最下面的直线（示教正例），H为整个规则空间
  - 搜索过程：G下移，S上移，H逐步缩小。
  - 结果：H收敛为只含一个要求的概念

变形空间法

- 消除侯选元素算法

(1)正规的初始H集是整个规则空间，这时S包含所有可能的示教正例（最特殊的概念）。

(2)接收一个新的示教例子。

如果是正例：去掉G中不覆盖新正例的概念，然后修改S为由新正例和S原有的元素共同归纳出的最特殊的结果

如果是反例：从S中去掉覆盖该反例的概念；然后修改G为由新反例和G原有元素共同特殊化为最一般的结果

(3)若G=S，且是单元集合，则转(4)，否则转(2)

(4)输出H中的概念（即G和S）

变形空间法

- 变形空间法的缺点

(1)抗干扰能力差

- 变形空间法是数据驱动的方法，所有数据驱动的方法都难以处理有干扰的训练例子
- 算法得到的概念应满足每个示教例子的要求，所以一个错误的例子会造成很大的影响

(2)无法发现析取概念

基于范例的学习CBR

- 人们为了解决一个新问题，先是进行回忆，从记忆中找到一个与新问题相似的范例，然后把该范例中的有关信息和知识复用到新问题的求解之中。
- 在基于范例推理 (Case-Based Reasoning, 简称CBR)中，把当前所面临的问题或情况称为目标范例(target case)，而把记忆的问题或情况称为源范例(base case)。粗略地说，基于范例推理就是由目标范例的提示而获得记忆中的源范例，并由源范例来指导目标范例求解的一种策略。

基于范例的学习CBR

- 范例（case）：“范例是一段带有上下文信息的知识，该知识表达了推理机在达到其目标的过程中能起关键作用的经验”。
- 具体来说，一个范例应具有如下特性：
  - 范例表示了与某个上下文有关的具体知识，这种知识具有可操作性。
  - 范例可以是各式各样的，可有不同的形状和粒度，可涵盖或大或小的时间片，可带有问题的解答或动作执行后的效应。
  - 范例记录了有用的经验，这种经验能帮助推理机在未来更容易地达到目标，或提醒推理机失败发生的可能性有多大等等。

基于范例的学习CBR

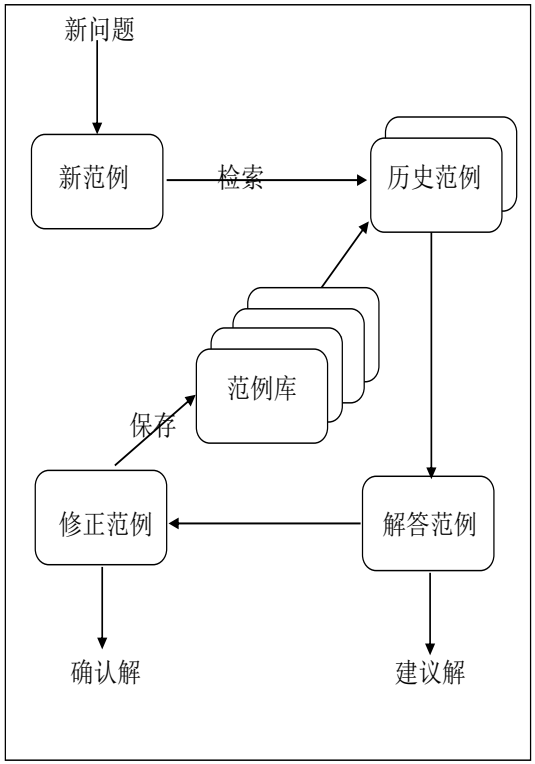
- 传统的推理观点是把推理理解为通过前因后果链(如规则链)演绎出结论的一个过程。许多专家系统使用的就是这种规则链式的推理方法。对于知识易于表示成启发式规则形式的问题来说，基于规则的方法比较适合，如分类问题和诊断问题等。但是人们在遇到一个新的问题的时候，一般先是回忆，从记忆中找到一个与新的问题相似的案例，然后把该案例中的有关信息和知识复用到新问题的求解之中。
- 基于范例推理中知识表示是以范例为基础，范例的获取比规则获取要容易,大大简化知识获取。对过去的求解结果进行复用，而不是再次从头推导，可以提高对新问题的求解效率。过去求解成功或失败的经历可以指导当前求解时该怎样走向成功或避开失败，这样可以改善求解的质量。对于那些目前没有或根本不存在可以通过计算推导来解决的问题。如在法律中的判例，基于范例推理能很好发挥作用。

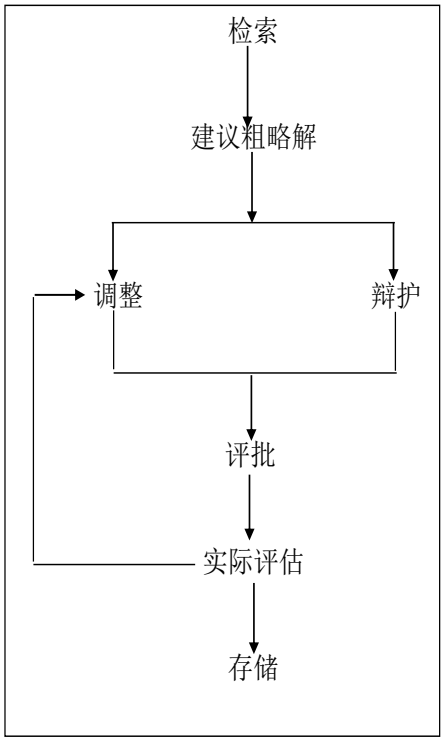
基于范例的学习CBR

- CBR采用的是和基于规则链推理完全不同的观点，在CBR中，使用的主要知识不是规则而是范例(case)，这些范例记录了过去发生的种种相关情况。
- 对CBR来讲，求解一个问题的结论不是通过链式推理产生的，而是从记忆里或范例库中找到与当前问题最相关的范例，然后对该范例作必要的改动以适合当前问题。
- CBR的基本思想是：人们的推理过程是基于特殊的经验而不是一组总的指导原则。
- 和其他基于AI的推理方法比较，CBR是通过联想（或类比），从过去的案例出发，把过去的案例和当前面临的问题相比较做出决策的过程。问题的解答来自于过去的经验而不是规则，这些经验以案例的方式存贮

CBR的过程模型

基于范例的推理是一个“回忆和调整”或“回忆和比较”的过程。在范例推理中，范例用于辅助理解和分析情景并用于辅助解决问题。我们的日常推理中，情景的理解、分析和问题的解决过程一般是相辅相成的。当我们还没有理解一个问题所处的情景时，是不可能解决该问题的；另一方面，我们又需要通过解决某个问题才能充分理解与它有关的情况。我们常常通过采用一些评估方法去检验结果来评价求解结论的好坏；同时我们在评估的过程中又需要解决新的问题。





解释学习

- 实例中进行学习历来是机器学习领域研究的焦点，早期的工作基本上局限于归纳学习的范畴，相应的研究成果很少或基本没有考虑背景知识对学习过程的影响。因此，这些方法从根本上来说是以数据为第一位的，没有反映出人工智能领域基于知识的研究和发展倾向。
- 从20世纪80年代中期开始，机器学习研究重点开始由以归纳方法为主的数据密集型学习方法的研究向多样化方法发展，开始研究分析方法、遗传算法、连接学习等。其中分析学习利用丰富的领域知识为背景，只需要通过分析很少的几个例子（通常是一个例子），就能将例子泛化为对目标概念的解释(通过泛化实例的解释，而不是泛化实例自身).
- 分析方法主要依赖于演绎推理，以产生更有效的问题求解知识，如搜索控制知识。因此，分析学习的主要目的是提高问题的求解效率，而不是获取新的概念描述。

解释学习

- 基于解释的学习是分析学习的主要方式，基于解释的学习（简称EBL）是将大量的成果汇集在一个统一、简单的框架内，通过分析为什么实例是某个目标概念的一个具体例子，EBL对分析过程（一个解释）加以推广，剔去与具体例子有关的成分，从而产生目标概念的一个描述。
- EBL的初始状态：
  - DT（Domain Theory）包含一组事实和规则，用于证明（解释）训练实例如何满足目标概念。
  - TC（Target Concept）是待学概念的一个非操作性描述。
  - E为目标概念的一个例子。
  - C（Operationality Criterion）是定义在概念描述上的一个二阶谓词，用以表示学习得到的目标概念可用哪些基本的、可操作的概念表示，以使这些知识能用于问题求解活动。

解释学习

- 基于解释的学习过程:可划分为下面两个步骤：

（1） 分析阶段：使用领域理论建立一个证明训练例子满足目标概念定义（初始描述）的解释结构，该结构可表示为一棵证明树，又称为解释树，它用于解释为什么实例是目标概念的一个实例，其每个分枝的叶结点上的表达式都必须满足可操作性准则。

（2） 基于解释的泛化（Explanation-Based Generalization，简称EBG）阶段：通过将实例证明树中的常量用变量进行替换，从而完成解释的泛化，并使其满足可操作准则，形成一棵基于解释的泛化树（简称EBG树），得到目标概念的一个充分条件。

神经网络

- 人工神经网络发展概况
- 神经网络基本概念
  - 生物神经元与人工神经元
  - 神经网络学习规则
- 前馈神经网络
  - 感知器
  - BP网络
  - 自组织神经网络（SOM）
- 反馈神经网络（Hopfield网络）
- 神经网络在控制中的应用

神经网络发展历史

四个发展阶段：

第一阶段：启蒙期，始于1943年。

形式神经元的数学模型提出。

第二阶段：低潮期，始于1969年。

《感知器》(Perceptions)一书出版，指出局限性 。

第三阶段：复兴期，从1982年到1986年。

Hopfield的两篇论文提出新的神经网络模型；

《并行分布处理》出版，提出反向传播算法。

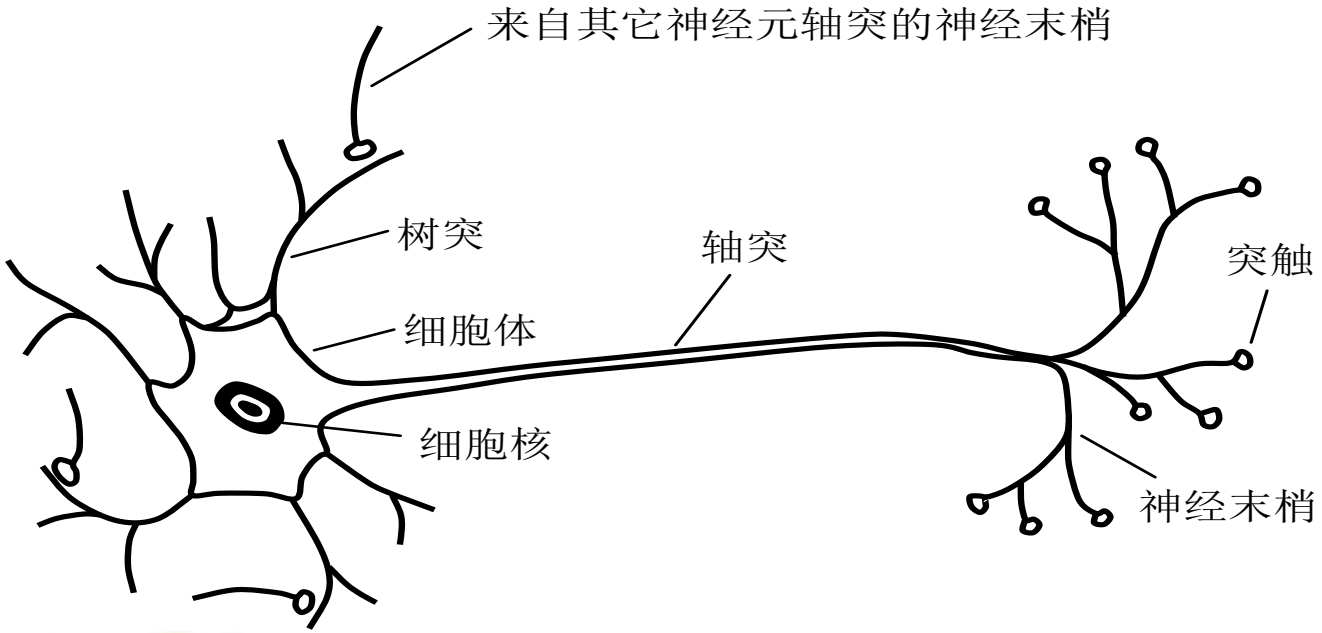
第四个阶段：1987年至今，趋于平稳。

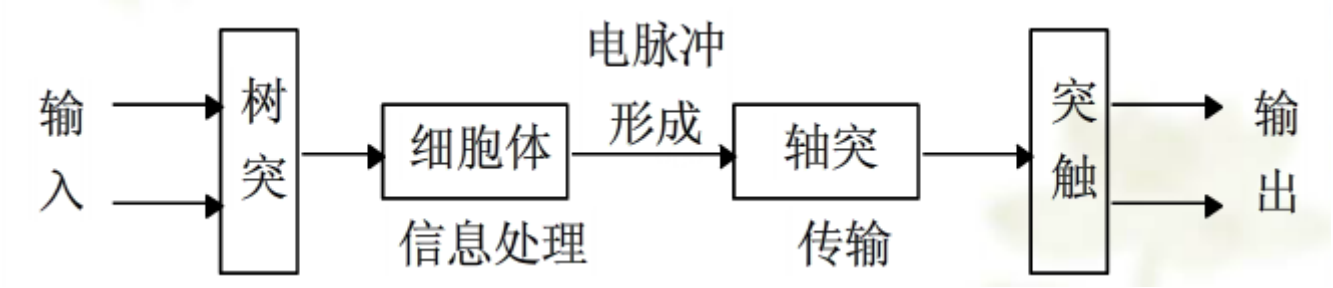
J．D．Cowan与D．H．Sharp回顾性综述文章
“Neural nets and artificial intelligence” 。

神经网络的基本概念

1．生物神经元的结构

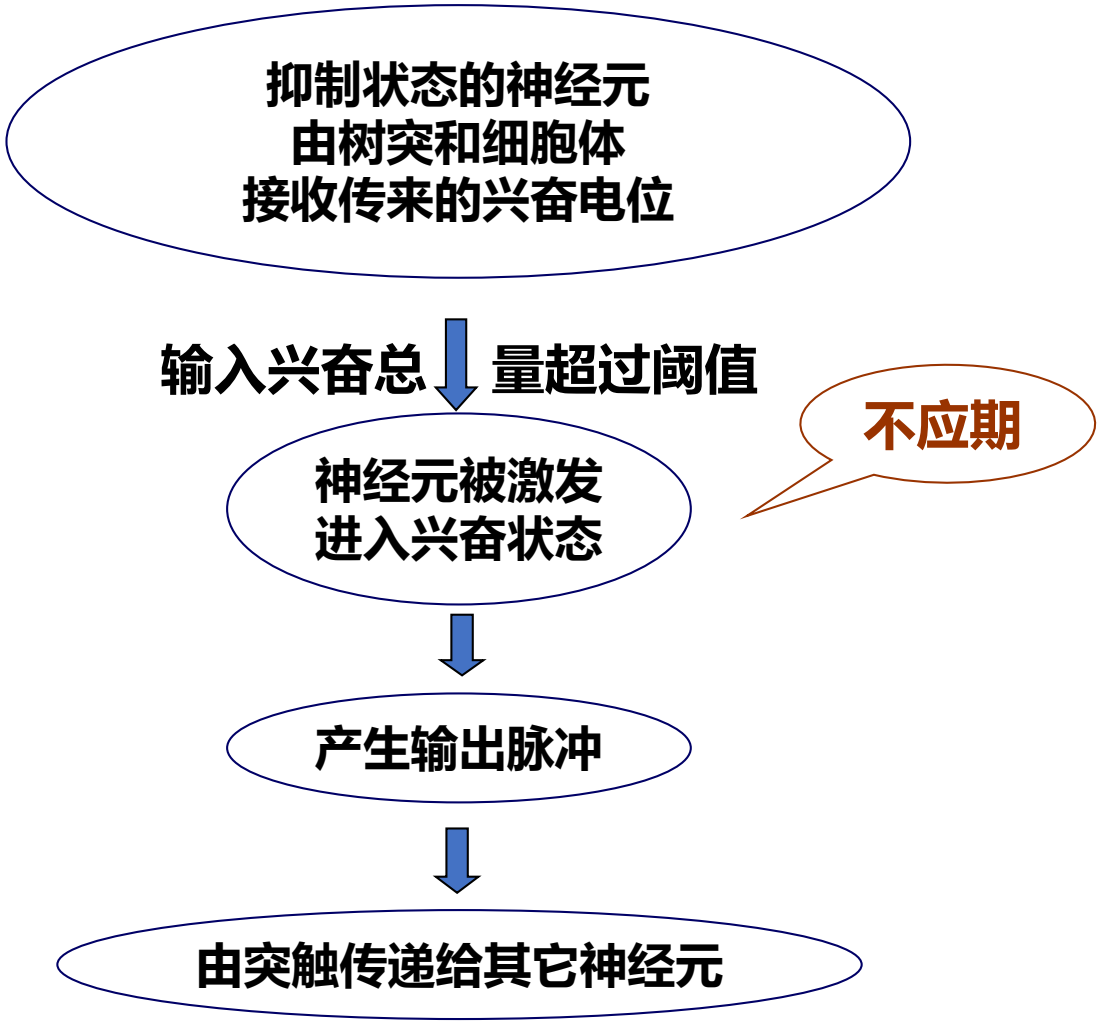
神经细胞是构成神经系统的基本单元，称之为生物神经元，简称神经元。
神经元主要构成：细胞体、轴突、树突和突触。





生物神经元的工作机制

兴奋和抑制两种状态。

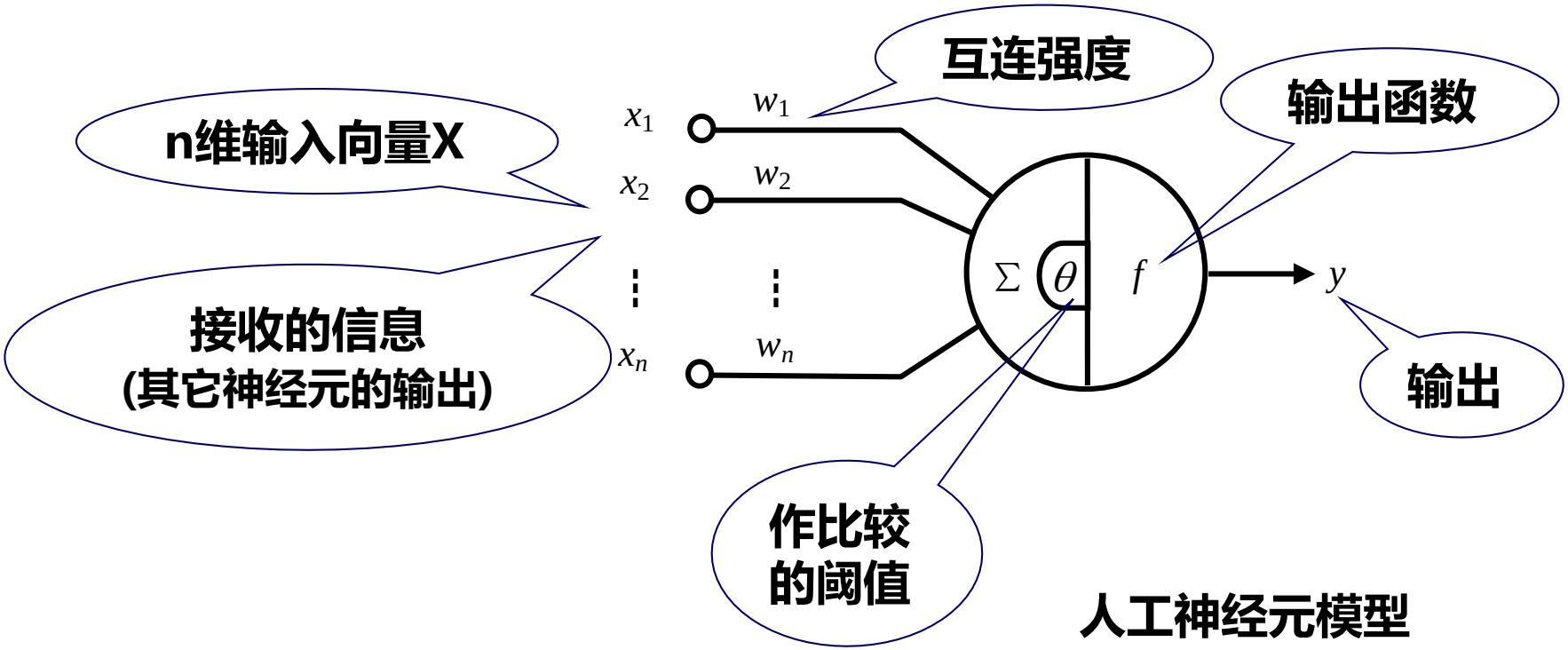


六个基本特征

- 1）神经元及其联接；
- 2）神经元之间的联接强度决定信号传递的强弱；
- 3）神经元之间的联接强度是可以随训练改变的；
- 4）信号可以是起刺激作用的，也可以是起抑制作用的；
- 5）一个神经元接受的信号的累积效果决定该神经元的状态；
- 6）每个神经元可以有一个“阈值”。

人工神经元及神经网络

人工神经元：生物神经元的简化模拟。



人工神经元模型

人工神经元间的互连：信息传递路径轴突-突触-树突的简化；

连接的权值：两个互连的神经元之间相互作用的强弱。

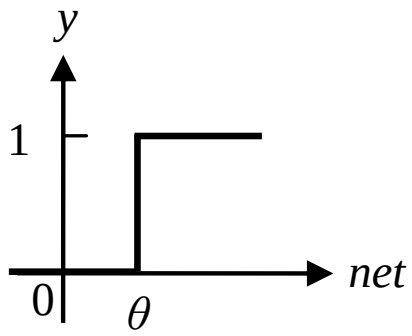
神经网络

神经元的动作：

$$net = \sum_{i=1}^{n} w_i x_i \qquad (x_i, w_i \in R)$$

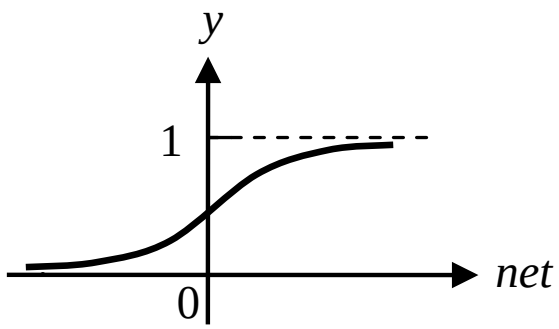
$$y = f(net - \theta)$$

输出函数 f：也称作用函数、活化函数、激活函数，非线性。



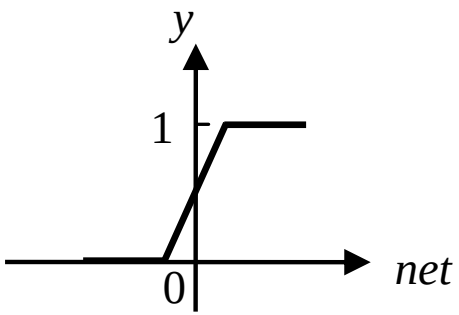
（a）

阈值型



（b）

S型



（c）

伪线性型

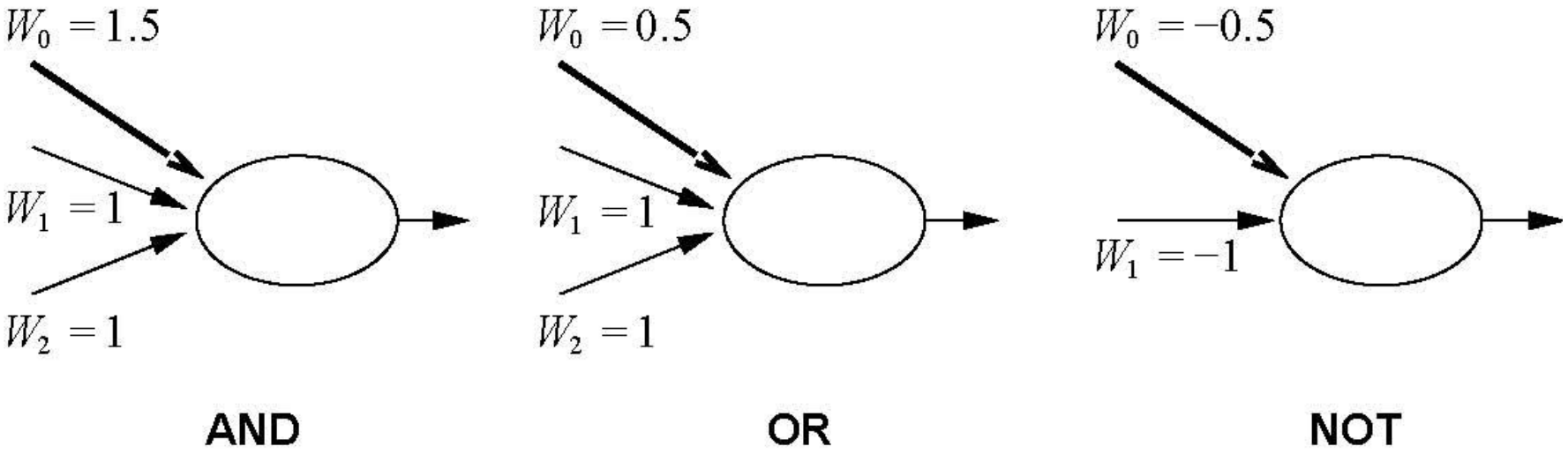
f 为阈值型函数时：

$$y = \mathrm{sgn}\left(\sum_{i=1}^{n} w_i x_i - \theta\right)$$

设 $w_{n+1} = \theta$ **，点积形式：** $y = \mathrm{sgn}(\boldsymbol{W}^{\mathrm{T}} \boldsymbol{X})$

式中， $\boldsymbol{W} = [w_1, \cdots, w_n, w_{n+1}]^{\mathrm{T}}$ $\boldsymbol{X} = [x_1, \cdots, x_n, -1]^{\mathrm{T}}$

实现逻辑函数



McCulloch和Pitts（1943）：可实现任何布尔函数

神经网络的学习

学习：

神经网络的最重要特征之一。

实质：

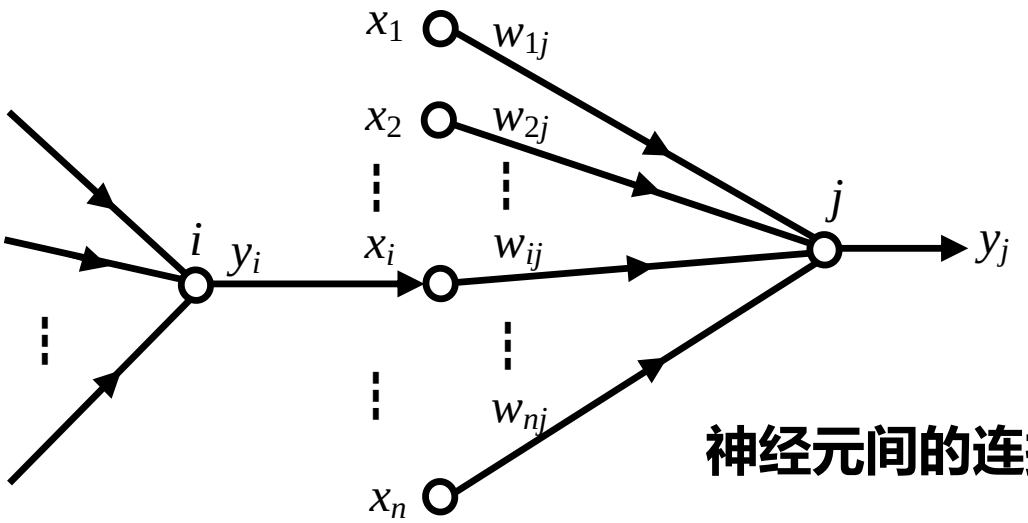
同一个训练集的样本输入输出模式反复作用于网络，网络按照一定的训练规则自动调节神经元之间的连接强度或拓扑结构，使实际输出满足期望的要求或者趋于稳定。

典型的权值修正方法： Hebb学习规则、误差修正学习

a．Hebb学习规则

如果神经网络中某一神经元与另一直接与其相连的神经元同时处于兴奋状态，那么这两个神经元之间的连接强度应该加强。

神经网络的学习



神经元间的连接

$$w_{ij}(t+1) = w_{ij}(t) + \eta[y_j(t)y_i(t)]$$

w_{ij}(t+1)：修正一次后的某一权值；

η：学习因子，表示学习速率的比例常数；

y_j(t)，y_i(t)：分别表示t时刻第j个和第i个神经元的状态（输出）。

由 $y_i(t) = x_i(t)$ **有：**

$$w_{ij}(t+1) = w_{ij}(t) + \eta[y_j(t)x_i(t)]$$

神经网络的学习

b. δ学习规则

（1）选择一组初始权值$w_{ij}(1)$；

（2）计算某一输入模式对应的实际输出与期望输出的误差；

（3）更新权值，阈值可视为输入恒为（1或-1）的一个权值；

$$w_{ij}(t+1) = w_{ij}(t) + \eta[d_j - y_j(t)]x_i(t)$$

式中，η：学习因子；

d_j，$y_j(t)$：第j个神经元的期望输出与实际输出；

$x_i(t)$：第j个神经元的第i个输入。

（4）返回（2），直到对所有训练模式网络输出均能满足要求。

神经网络的结构分类

分层结构

有明显层次，信息流向由输入层到输出层。

—— 前馈网络

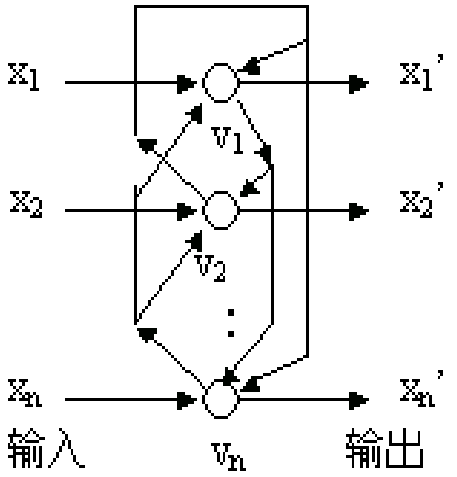
相互连接结构

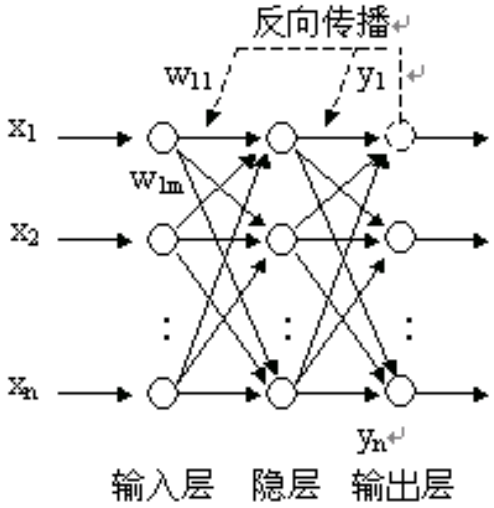
没有明显层次，任意两个神经元之间可达，具有输出单元到隐层单元或输入单元的反馈连接 。

—— 反馈网络

神经网络的结构

神经网络的结构





(1)递归网络

有些神经元的输出被反馈至同层或前层神经元，信号能够从正向或反向流通。又叫反馈网络。

典型例子：Hopfield网络、Elmman网络和Jordan网络

如图：vi表示接点的状态，xi为节点的输入值，xi’为收敛后的输出值，i=1,2,…,n

(2)前馈网络

具有递阶分层结构，由一些同层神经元间不存在互连的层次组成。从输入到输出的信号通过单向连接流通；神经元从一层连接至下一层，不存在同层神经元间的连接。如图：实线指明实际信号流通，虚线表示反向传播。

典型例子：多层感知器MLP

神经网络的结构

- 分层形前向网络具有任意精度的模式映射能力，因而可以用作模式分类、匹配等，
- 而反馈型神经网络则是一个非线性动力学系统，它具有如下两个重要特征：

1．系统具有多个稳定状态，从某一初始状态开始运动，系统最终可以到达某一个稳定状态；

2．不同的初始连接权值对应的稳定状态也不相同。

如果用系统的稳定状态作为记忆，那么由某一初始状态出发向稳态的演化过程，实际上就是一个联想过程，所以反馈型神经网络具有联想记忆的功能。

神经网络的结构

决定人工神经网络整体性能

- 节点本身的信息处理能力(数学模型)
- 节点与节点之间连接(拓扑结构)
- 相互连接的强度(通过学习来调整)

神经网络的结构

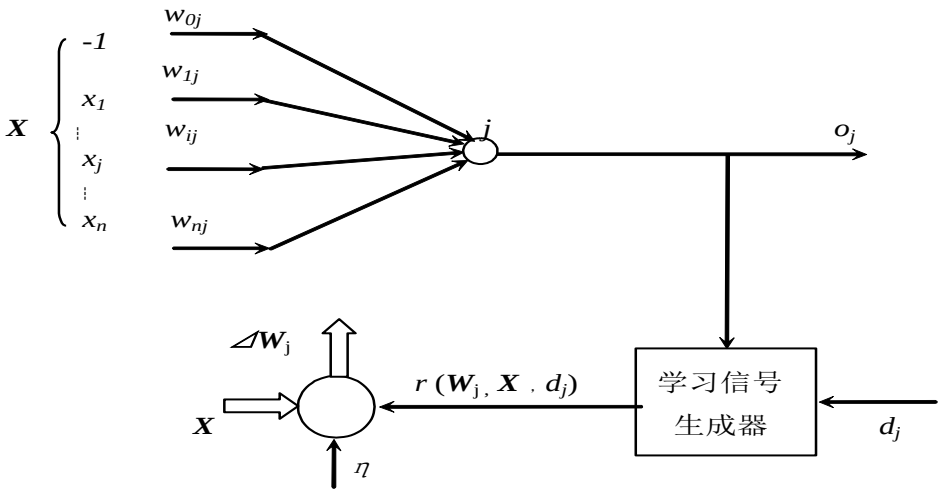
神经网络的主要学习算法

加拿大心理学家Donald Hebb出版了《行为的组织》一书，指出学习导致突触的联系强度和传递效能的提高，即为“赫布律”。

在此基础上，人们提出了各种学习规则和算法，以适应不同网络模型的需要。有效的学习算法，使得神经网络能够通过连接权值的调整，构造客观世界的内在表示，形成具有特色的信息处理方法，信息存储和处理体现在网络的连接中。

神经网络能够通过对样本的学习训练，不断改变网络的连接权值以及拓扑结构，以使网络的输出不断地接近期望的输出。这一过程称为神经网络的学习或训练，其本质是可变权值的动态调整。

神经网络的结构



$$\Delta W_j = \eta r[W_j(t), X(t), d_j(t)]X(t)$$

$$W_j(t+1) = W_j(t) + \eta r[W_j(t), X(t), d_j(t)]X(t)$$

3.神经网络的主要学习算法

(1)有师学习

能够根据期望的和实际的网络输出(对应于给定输入)之间的差来调整神经元间连接的强度或权。因此，有师学习需要有老师或导师来提供期望或目标输出信号。

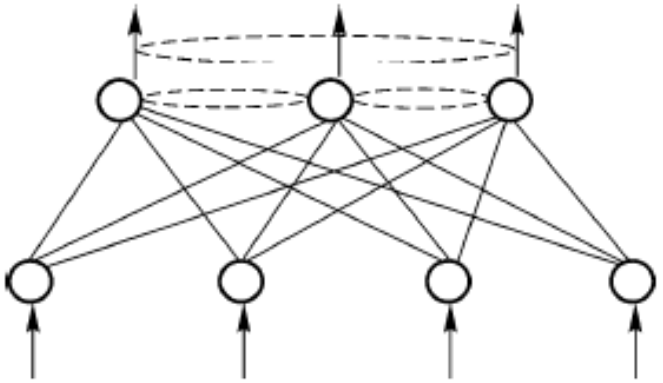
典型例子：Δ规则、广义Δ规则或反向传播算法

神经网络的结构

(2)无师学习

不需要知道期望输出。在训练过程中，只要向神经网络提供输入模式，神经网络就能够自动适应连接权，以便按相似特征把输入模式分组聚集。

典型例子：Kohonen算法、Carpenter-Grossberg自适应谐振理论。



获胜神经元才有权调整其权向量 W_{j^*} ,调整后权向量为

$$\begin{cases} W_{j^*}(t+1)=\hat{W}_{j^*}(t)+\Delta W_{j^*}=\hat{W}_{j^*}(t)+\alpha(\hat{X}-\hat{W}_{j^*}) \\ W_j(t+1)=\hat{W}_j(t) \qquad j\neq j^* \end{cases}$$

(3)强化学习

是有师学习的特例。它不许要老师给出目标输出，而采用一个评论员来评价与给顶输入相对应的神经网络输出的优度。

典型例子：遗传算法

神经网络的典型模型

感知器神经网络：

1.感知器(Perceptron)是最“古老”的网络(Rosenblatt，于1975年提出)，是一组可训练的线性分类器，目前已很少使用。

有师学习　误差修正　正向　线形分类、预测

2. MadaLine是AdaLine的发展，是一组具有最小均方差线性网络的组合，学习能力较强，但I/O间需满足线性关系。

有师学习　误差修正　正向　分类，噪声抑制

3.反向传递(BP)网是一种反向传递并修正误差的多层映射网，在参数适当时，能收敛到较小的均方误差，是当前应用最广的一种网络。缺点是训练时间长，易陷入局部极小。

有师学习　误差修正　反向　分类

神经网络的典型模型

自组织竞争学习神经网络模型：

4.自组织映射网(SOM)由Kohonen于1972年提出。能形成簇与簇之间的连续映射，起向量量化器的作用

无师学习　　竞争律　　正向　　自组织映射

5.CPN(Counter Propagation Network)由R Hecht和Nielsen于1987年提出，亦称对流网，将Kohonen 特征映射网络与Grossberg 基本竞争型网络相结合，充分发挥了它们各自的特长：无导师训练解决网络隐含层的理想输出未知问题，有导师训练解决输出层按系统要求给出指定输出结果的问题。经过反复学习，PN 可以将任意输入模式映射为输出模式.

无师学习和有师学习　Hebb律　　正向　　模式映射

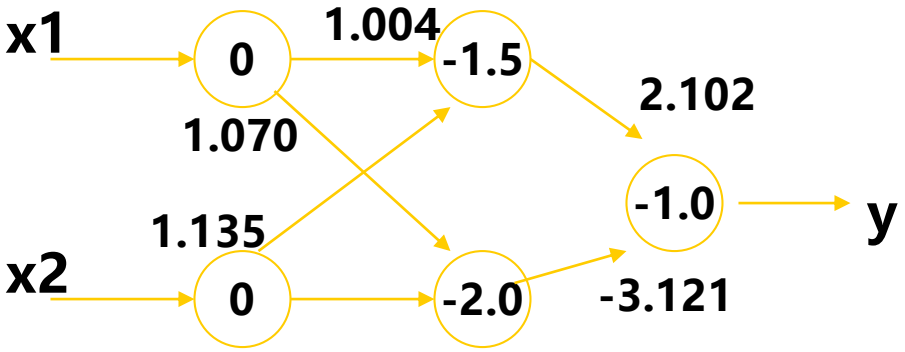
6. 自适应共振(ART)由Grossberg提出，是根据可选参数对输入数据进行粗分类的网络，ART Ⅰ用于二值输入，ARTⅡ用于连续值输入。缺点是太敏感，输入有小的变化，输出变化很大。

无师学习　　Hebb律　　反向　　模式分类

- **基于神经网络的知识表示**

这里采用的是一种隐式的表示方法。某一问题的若干知识在同一网络中表示。

例如：在有些神经网络系统中，知识用神经网络中所对应的有向权图的邻接矩阵及阈值向量表示的。



邻接矩阵为：

$$\begin{pmatrix} 0 & 0 & 1.004 & 1.070 & 0 \\ 0 & 0 & 1.135 & 1.100 & 0 \\ 0 & 0 & 0 & 0 & 2.102 \\ 0 & 0 & 0 & 0 & -3.121 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

该网络代表下列4条规则：IF x1=0 AND x2=0 THEN y=0

IF x1=0 AND x2=1 THEN y=1

IF x1=1 AND x2=0 THEN y=1

IF x1=1 AND x2=1 THEN y=0

医疗诊断实例：

假设系统的诊断模型只有六种症状、两种疾病、三种治疗方案。

首先选择一批合适的病人并从病历中采集如下信息：

(1)症状：对每一症状只采集有、无及没有记录这三种信息。

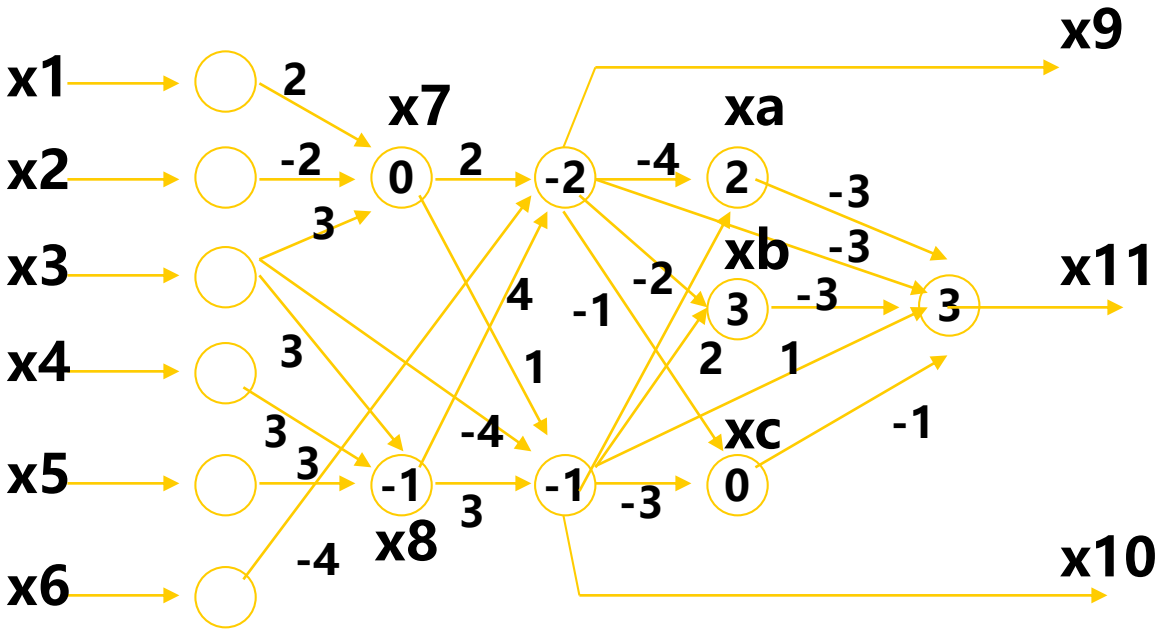
(2)疾病：对每一疾病也只采集有、无及没有记录这三种信息。

(3)治疗方案：对每一治疗方案只采集是否采用这两种信息。

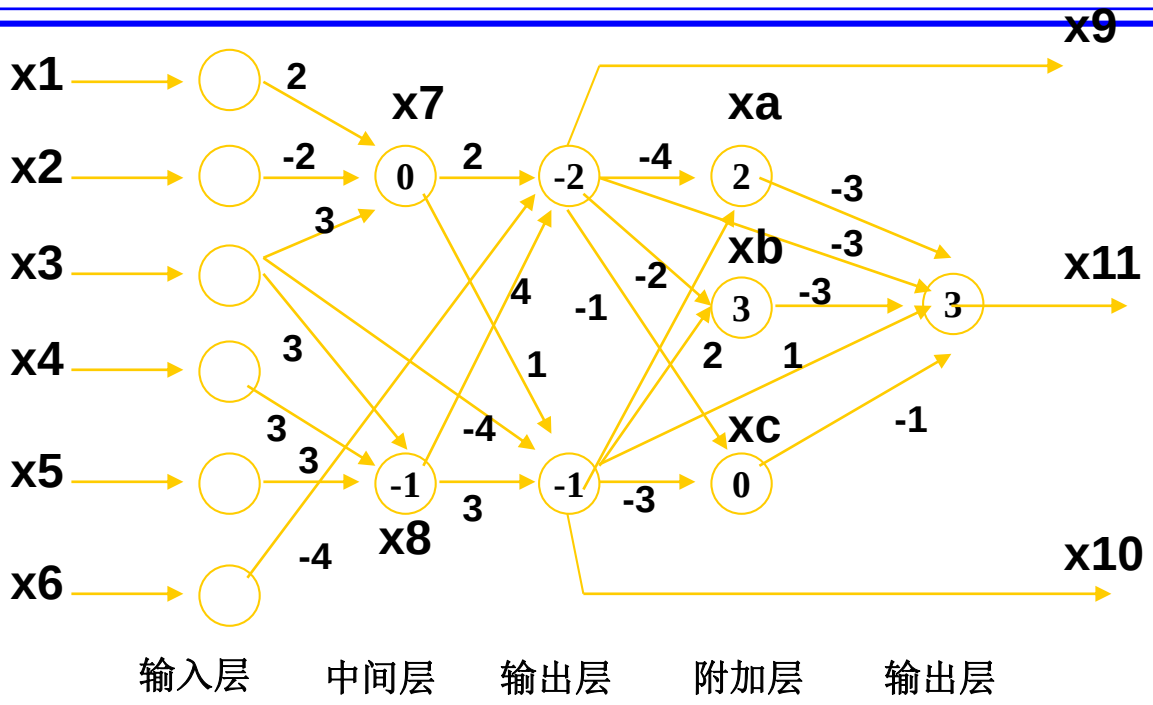
其中，对“有”、“无”、“没有记录”分别用+1、-1、0表示。这样每一个病人构成一个训练样本。

神经网络的知识表示与推理

根据症状、疾病及治疗方案间的因果关系，并通过训练样本对网络的训练得到如下神经网络 。其中，x_1, x_2 , … x_6 ,为症状； x_7, x_8 为疾病名； x_9, x_{10} , x_{11} 为治疗方案； x_a, x_b , x_c ,是附加层，这是由于学习算法的需要而增加的



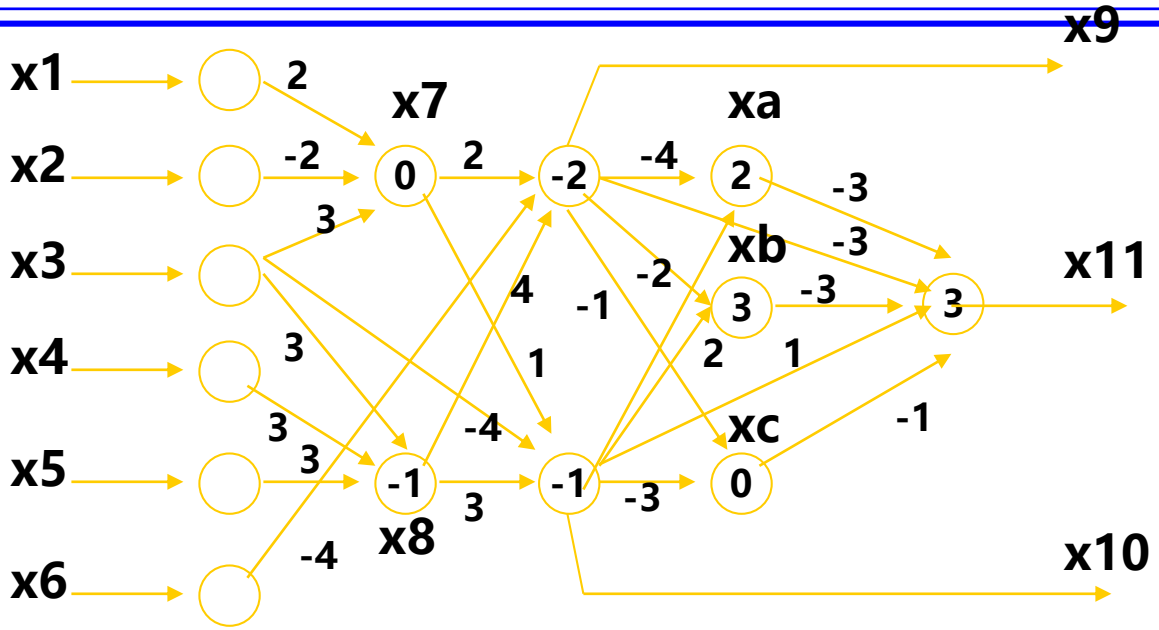
神经网络的知识表示与推理



说明：

(1)这是一个带正负权值w_{ij}的前向网络，有w_{ij}可构成相应的学习矩阵。当i>j时w_{ij} =0；当i<j且接点I和接点j之间不存在连接弧时， w_{ij}也为0；其余， w_{ij}为图中连接弧上所标的数据。

神经网络的知识表示与推理



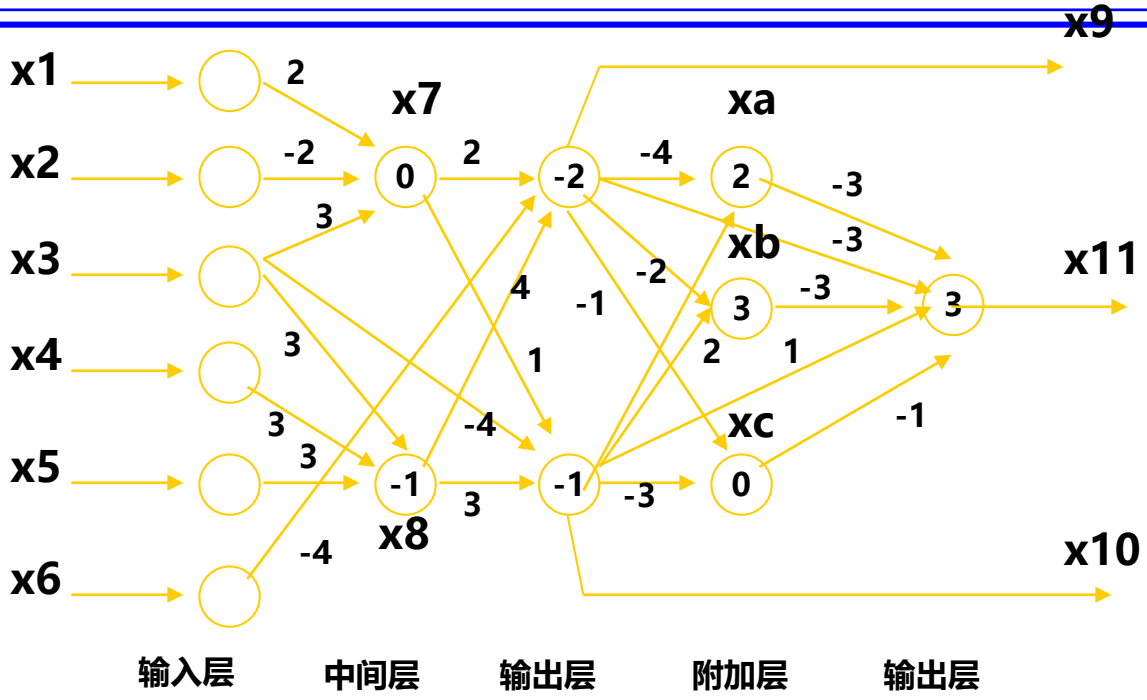
说明：

(2)神经元取值为+1,0,-1，特性函数为一离散型的阈值函数，其计算公式为

$$X_j=\sum_{i=1}^{n} w_{ij}x_i \qquad x_j' = \begin{cases} +1, & X_j>0 \\ 0, & X_j=0 \\ -1, & X_j<0 \end{cases}$$

为计算方便，增加了$w_{0j}x_0$项。X_0的值为常数，w_{0j}的值标在节点的圆圈中，它实际上是$-\theta_j$，θ_j是节点j的阈值。

神经网络的知识表示与推理



说明：

(3)图中连接弧上标出的w_{ij}值是根据一组训练样本，通过某种学习算法对网络进行训练得到的。这就是神经网络系统所进行的知识获取。

(4)由全体的值及各种症状、疾病、治疗方案名所构成的集合就形成了该疾病诊断系统的知识库。

浙江大学 智能系统和控制研究所
Institute of Cyber-Systems and Control,Zhejiang University

2.基于神经网络的推理

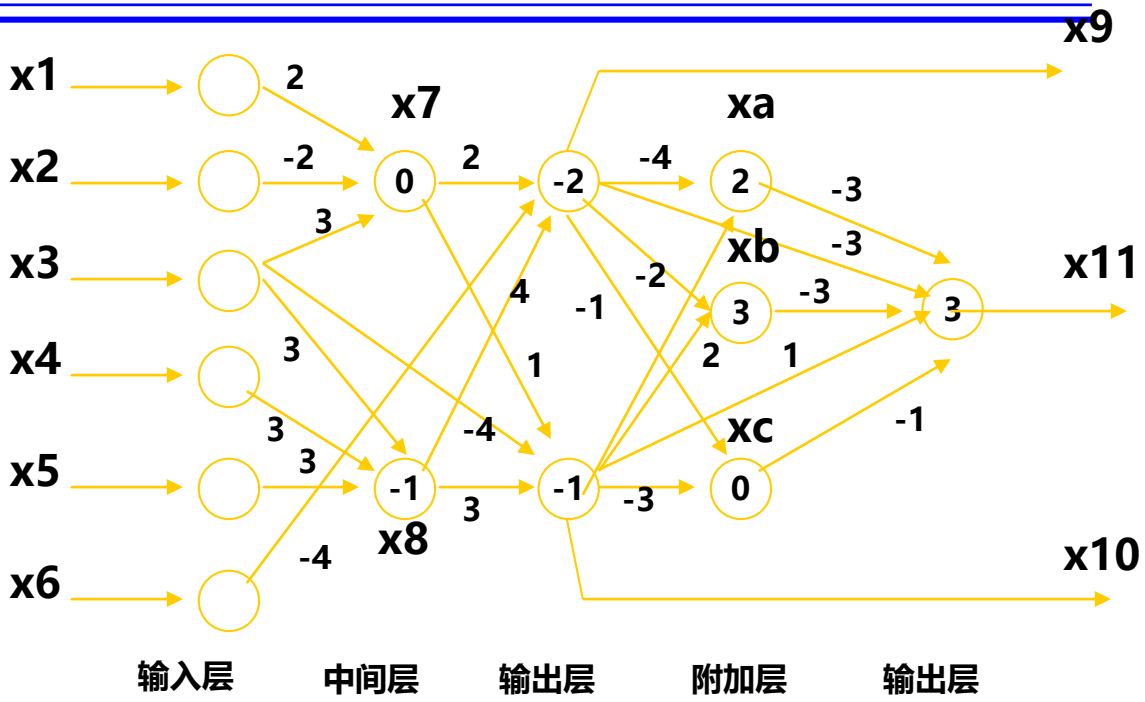
通过网络计算实现的，把用户提供的初始证据用作网络的输入，通过网络计算最终得到输出结果。

例如：

证据是$x_1=1, x_2=x_3=-1$

由 $0+2\times1+(-2)\times(-1)+3\times(-1)=1>0$

得$x_7=1$



当证据是x1=x3=1， x2=？ 由 $0+2\times1+3\times1=5+(-2)\times(?)>0$ 得x7=1

由此可见，在用神经网络进行推理时，即使已知的信息不完全，照样可以进行推理。

2.基于神经网络的推理

正向网络推理步骤：

(1)把已知数据输入网络输入层的各个节点。

(2)利用特性函数分别计算网络中各层的输出。计算中，前一层的输出作为后一层有关节点的输入，逐层进行计算，直到计算出输出层的输出值为止。

(3)用阈值函数对输出层的输出进行判定，从而得到输出结果。

推理具有如下特征：

(1)同一层的处理单元(神经元)是完全并行的，但层间的信息传递是串行的。由于层中处理单元的数目比网络的层次多得多，因此它是一种并行推理。

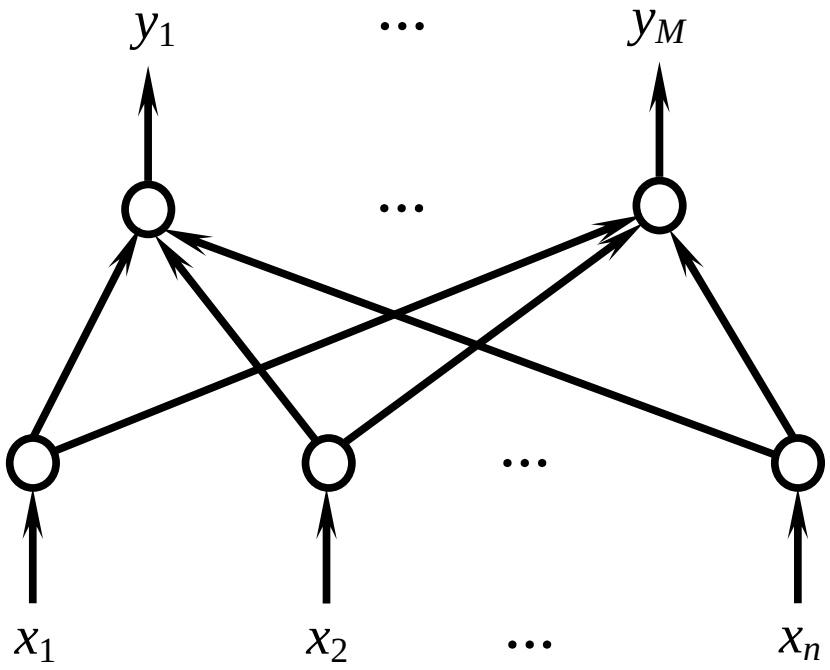
(2)在网络推理中不会出现传统人工智能系统中推理的冲突问题。

(3)网络推理只与输入及网络自身的参数有关，而这些参数又是通过多网络进行训练得到的，因此它是一种自适应推理。

前馈神经网络

1．感知器

感知器（Perceptron）：F．Rosenblatt于1957年提出。



感知器结构示意图

结构特点：

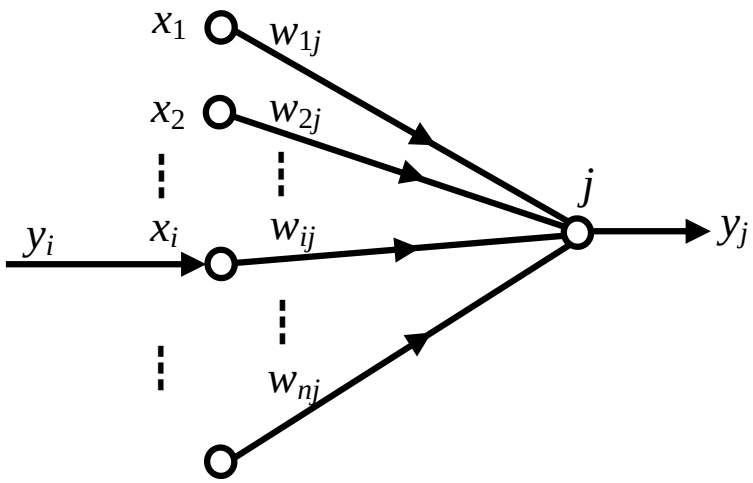
* 双层（输入层、输出层）；
* 两层单元之间为全互连；
* 连接权值可调。
* 输出层神经元个数等于类别数（两类问题时输出层为一个神经元）。

前馈神经网络

设输入模式向量，$\boldsymbol{X}=[x_1, x_2, \cdots, x_n]^{\mathrm{T}}$ 共*M*类。

输出层第*j*个神经元对应第*j*个模式类，

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



θ_j：第*j*个神经元的阈值；

w_{ij}：输入模式第*i*个分量与输出层第*j*个神经元间的连接权。

输出单元对所有输入数值加权求和，经阈值型输出函数产生一组输出模式。

令 $\theta_j = -w_{(n+1)j}$。取

$$\boldsymbol{W}_j = [w_{1j}, w_{2j}, \cdots, w_{(n+1)j}]^{\mathrm{T}} \qquad \boldsymbol{X} = [x_1, x_2, \cdots, x_n, 1]^{\mathrm{T}}$$

有

$$y_j = f(\sum_{i=1}^{n+1} w_{ij} x_i) = f(\boldsymbol{W}_j^{\mathrm{T}} \boldsymbol{X})$$

前馈神经网络

*M*类问题判决规则(神经元的输出函数) 为

$$y_j = f(\boldsymbol{W}_j^{\mathrm{T}}\boldsymbol{X}) = \begin{cases} +1, & 若\boldsymbol{X} \in \omega_j \\ -1, & 若\boldsymbol{X} \notin \omega_j \end{cases} \quad 1 \le j \le M$$

* 正确判决的关键：

输出层每个神经元必须有一组合适的权值。

* 感知器采用监督学习算法得到权值；

* 权值更新方法：*δ*学习规则。

算法描述

第一步：设置初始权值$w_{ij}(1)$，$w_{(n+1)j}(1)$为第*j*个神经元的阈值。

第二步：输入新的模式向量。

第三步：计算神经元的实际输出。

前馈神经网络

设第k次输入的模式向量为X_k，与第j个神经元相连的权向量为

$$\boldsymbol{W}_j(k)=[w_{1j},w_{2j},\cdots,w_{(n+1)j}]^{\mathrm{T}}$$

第j个神经元的实际输出为

$$y_j(k)=f[\boldsymbol{W}_j^{\mathrm{T}}(k)\boldsymbol{X}_k] \qquad 1\le j\le M$$

第四步：修正权值。

$$\boldsymbol{W}_j(k+1)=\boldsymbol{W}_j(k)+\eta[d_j-y_j(k)]\boldsymbol{X}_k$$

d_j：第j个神经元的期望输出。

$$d_j=\begin{cases}+1, & \boldsymbol{X}_k\in\omega_j\\ -1, & \boldsymbol{X}_k\notin\omega_j\end{cases} \qquad 1\le j\le M$$

第五步：转到第二步。

当全部学习样本都能正确分类时，学习过程结束。

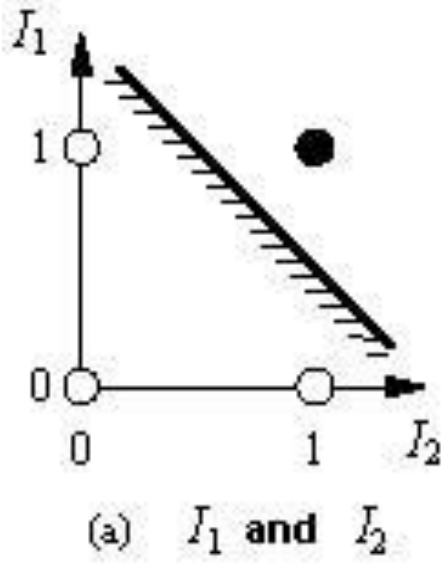
经验证明，当η随k的增加而减小时，算法一定收敛。

感知器的表达能力

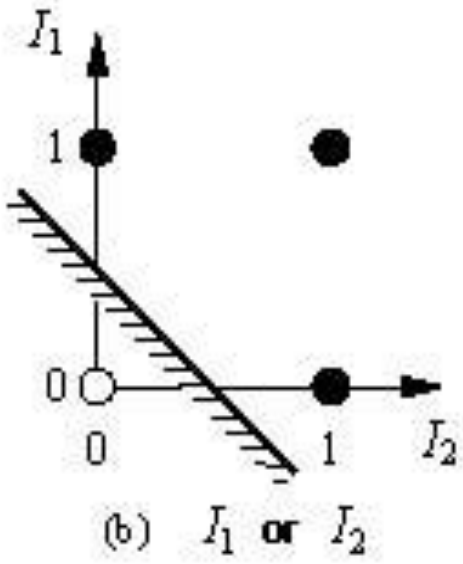
考虑一个f = 阶跃函数的感知器，可以表达AND，OR，NOT，多数函数等。

感知器的输出：$y_i = f\left(net_i\right) = \text{step}\left(\sum_j w_{j,i} x_j\right)$

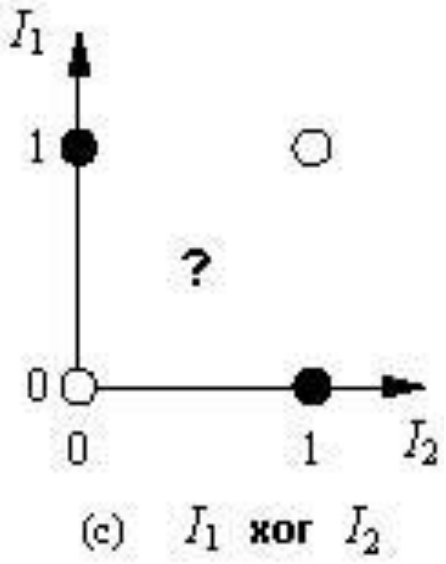
表达输入空间的一种**线性分割**：$\sum_j w_j x_j > 0 \quad \text{or} \quad \mathbf{W}\cdot\mathbf{X} > 0$



(a) I_1 and I_2



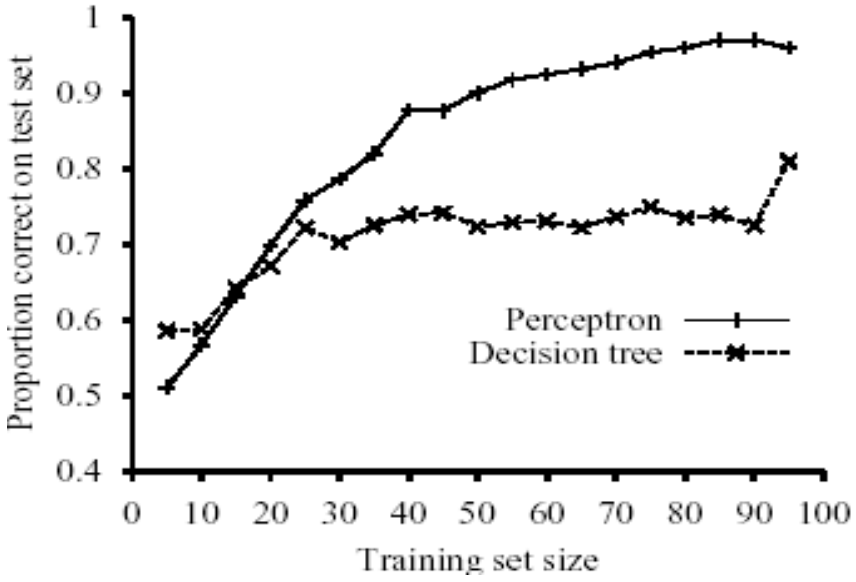
(b) I_1 or I_2



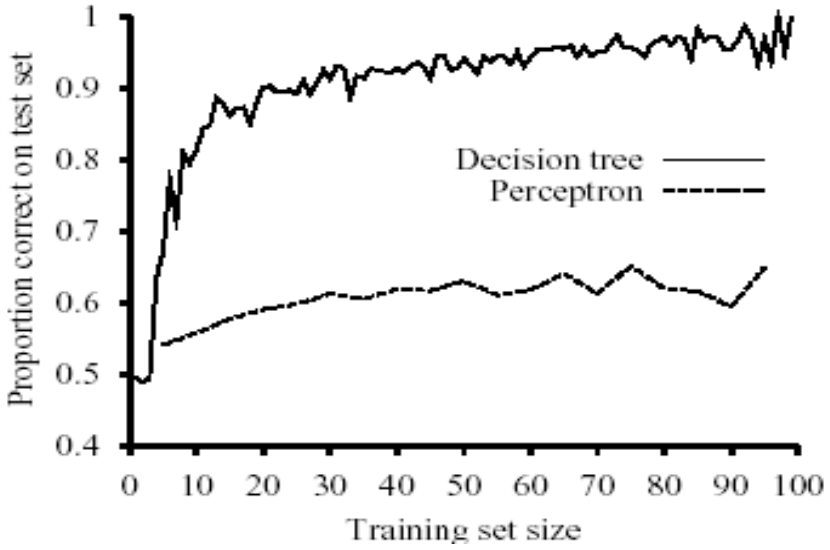
(c) I_1 xor I_2

感知器的表达能力

对于任意线性可分数据集，感知器学习规则收敛于一个一致函数



11个布尔输入的多数函数



餐厅例子

BP网络

BP网络：采用BP算法（Back-Propagation Training Algorithm）的多层感知器。

认识最清楚、应用最广泛。

误差反向传播算法

性能优势：识别、分类

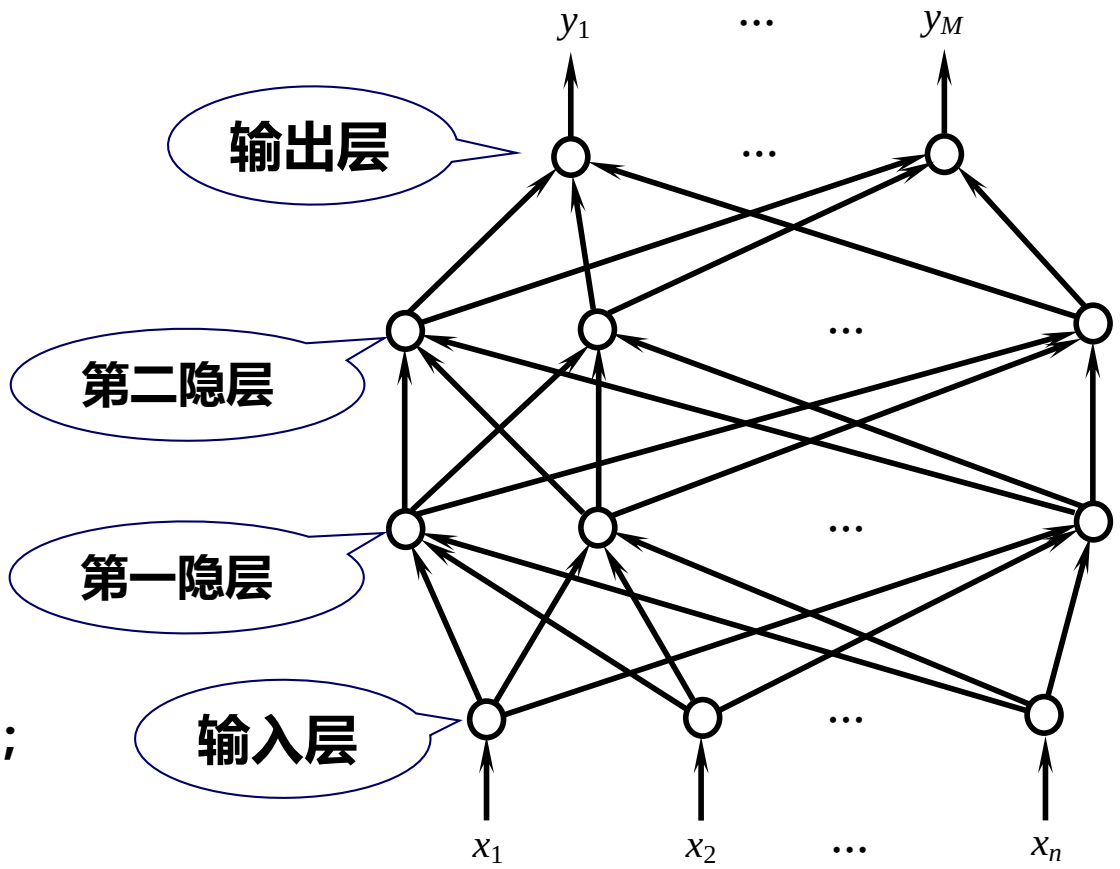
1．多层感知器

针对感知器学习算法的局限性：模式类必须线性可分。

结构：

前馈网络；

中间层为一层或多层处理单元；



BP 算法

两个阶段：
- 正向传播阶段：逐层状态更新
- 反向传播阶段：误差

BP算法的学习过程

设：某层任一神经元 j 的

输入为 net_j，输出为 y_j；

相邻低一层中任一

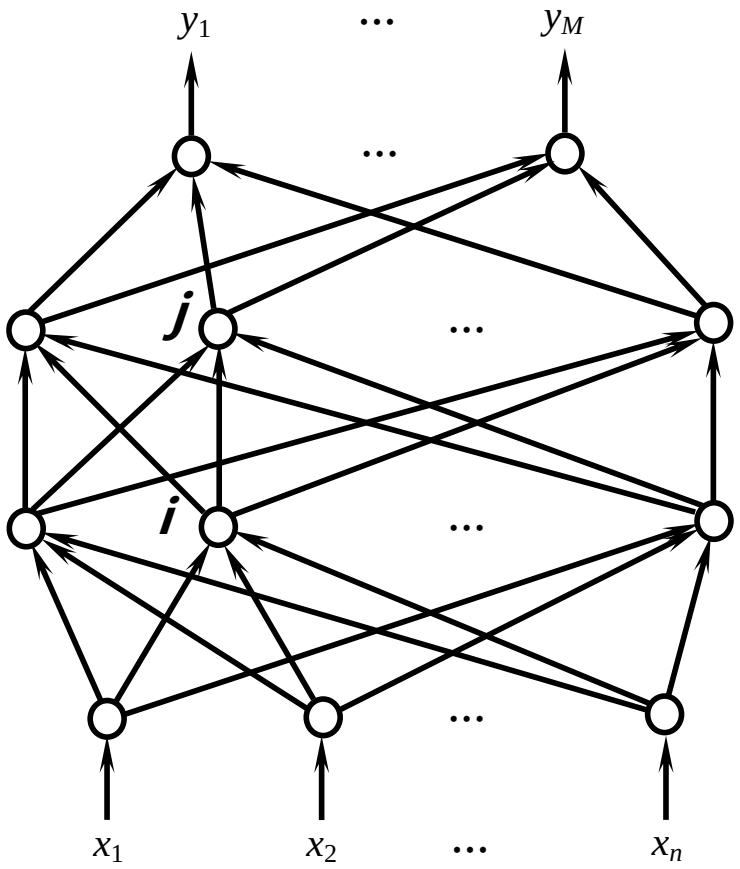
神经元 i 的输出为 y_i。

$$net_j = \sum_i w_{ij} y_i$$

$$y_j = f(net_j)$$

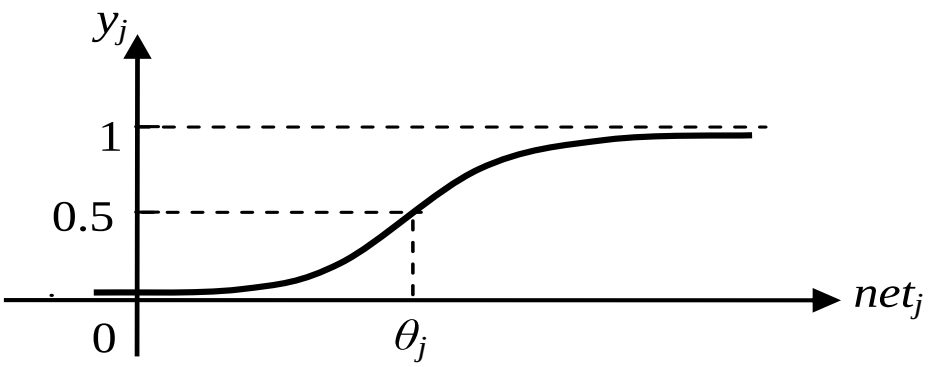
w_{ij}：神经元 i 与 j 之间的连接权；

$f(\bullet)$：神经元的输出函数。



BP 算法

S型输出函数：



$$y_j = f(net_j) = \frac{1}{1+e^{-(net_j+\theta_j)/h_0}}$$

θ_j：神经元阈值；

h_0：修改输出函数形状的参数。

设：输出层中第k个神经元的实际输出为y_k，输入为net_k；
与输出层相邻的隐层中任一神经元j的输出为y_j。

$$net_k = \sum_j w_{jk} y_j$$

$$y_k = f(net_k)$$

BP 算法

对输入样本X_p，若输出层中第k个神经元的期望输出为d_{pk}，实际输出为y_{pk}。输出层的输出方差：

$$E_p = \frac{1}{2}\sum_k (d_{pk} - y_{pk})^2$$

若输入N个样本，网络的系统均方差为：

$$E = \frac{1}{2N}\sum_p \sum_k (d_{pk} - y_{pk})^2 = \frac{1}{N}\sum_p E_p$$

当输入X_p时，w_{jk}的修正增量：

$$\Delta_p w_{jk} = -\eta \frac{\partial E_p}{\partial w_{jk}}$$

其中，

$$-\frac{\partial E_p}{\partial w_{jk}} = -\frac{\partial E_p}{\partial net_k} \cdot \frac{\partial net_k}{\partial w_{jk}}$$

由 $net_k = \sum_j w_{jk} y_j$ 式得到：

$$\frac{\partial net_k}{\partial w_{jk}} = \frac{\partial}{\partial w_{jk}} \sum_j w_{jk} y_{pj} = y_{pj}$$

BP 算法

令 $\delta_{pk} = -\partial E_p / \partial net_k$ ，可得

输出单元的误差：$\delta_{pk} = (d_{pk} - y_{pk}) y_{pk} (1 - y_{pk})$

输出单元的修正增量：$\Delta_p \omega_{jk} = \eta \delta_{pk} y_{pj}$

对于与输出层相邻的隐层中的神经元j和该隐层前低一层中的神经元i：

$$\delta_{pj} = y_{pj}(1 - y_{pj}) \sum_k \delta_{pk} w_{jk}$$

$$\Delta_p w_{ij} = \eta \delta_{pj} y_{pj}$$

输出层中神经元输出的误差反向传播到前面各层，对各层之间的权值进行修正。

BP算法步骤

第一步：对权值和神经元阈值初始化：(0，1)上分布的随机数。

第二步：输入样本，指定输出层各神经元的希望输出值。

$$d_j = \begin{cases} +1, & \boldsymbol{X} \in \omega_j \\ -1, & \boldsymbol{X} \notin \omega_j \end{cases} \qquad j = 1, 2, \cdots, M$$

第三步：依次计算每层神经元的实际输出，直到输出层。

第四步：从输出层开始修正每个权值，直到第一隐层。

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j y_i \qquad 0 < \eta < 1$$

若j是输出层神经元，则： $\delta_j = y_j(1-y_j)(d_j - y_j)$

若j是隐层神经元，则： $\delta_j = y_j(1-y_j)\sum_k \delta_k w_{jk}$

第五步：转到第二步，循环至权值稳定为止。

BP算法步骤

改进的权值修正：

$$w_{ij}(t+1) = w_{ij}(t) + \eta\delta_j y_i + \alpha[w_{ij}(t) - w_{ij}(t-1)]$$

α：平滑因子，0<α<1。

—— 收敛快、权值平滑变化

BP算法存在问题：

* **存在局部极小值问题；**
* **算法收敛速度慢；**
* **隐层单元数目的选取无一般指导原则；**
* **新加入的学习样本影响已学完样本的学习结果。**

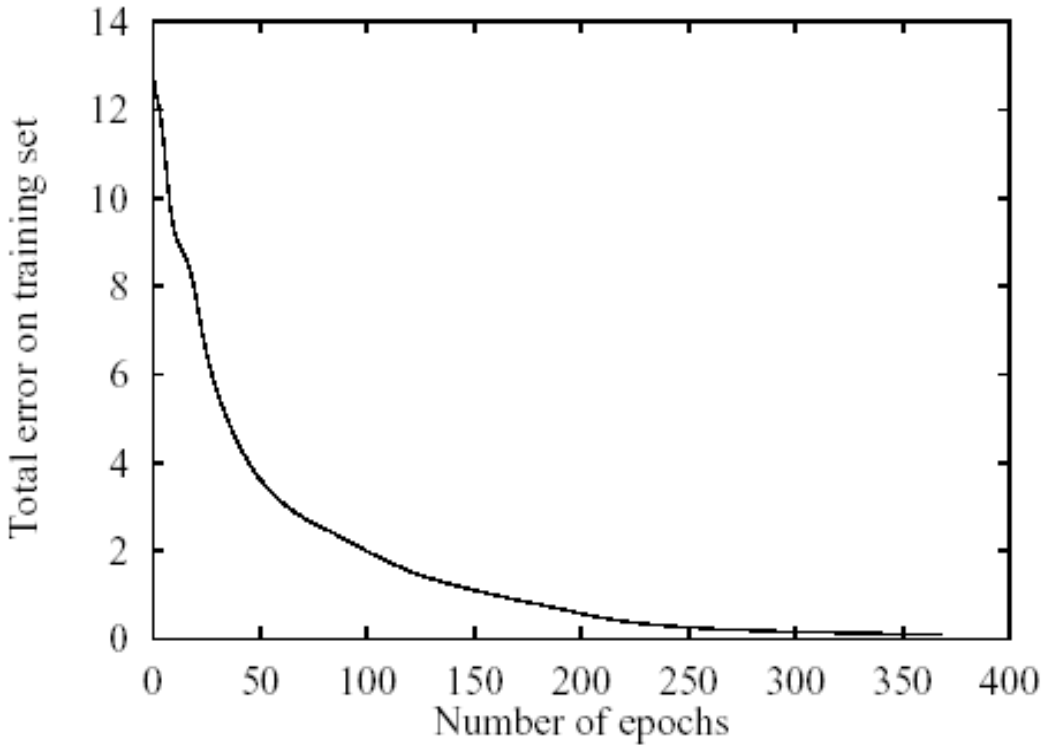
BP算法代码

function BACK-PROP-LEARNING(*examples*, *network*) **returns** a neural network
inputs: *examples*, a set of examples, each with input vector $\mathbf{x}$ and output vector $\mathbf{y}$
network, a multilayer network with L layers, weights $w_{i,j}$, activation function g
local variables: Δ, a vector of errors, indexed by network node

repeat
for each weight $w_{i,j}$ in *network* **do**
$w_{i,j} \leftarrow$ a small random number
for each example $(\mathbf{x}, \mathbf{y})$ **in** *examples* **do**
/* *Propagate the inputs forward to compute the outputs* */
for each node i in the input layer **do**
$a_i \leftarrow x_i$
for $\ell = 2$ **to** L **do**
for each node j in layer ℓ **do**
$in_j \leftarrow \sum_i w_{i,j}\, a_i$
$a_j \leftarrow g(in_j)$
/* *Propagate deltas backward from output layer to input layer* */
for each node j in the output layer **do**
$\Delta[j] \leftarrow g'(in_j) \times (y_j - a_j)$
for $\ell = L - 1$ **to** 1 **do**
for each node i in layer ℓ **do**
$\Delta[i] \leftarrow g'(in_i) \sum_j w_{i,j}\, \Delta[j]$
/* *Update every weight in network using deltas* */
for each weight $w_{i,j}$ in *network* **do**
$w_{i,j} \leftarrow w_{i,j} + \alpha \times a_i \times \Delta[j]$
until some stopping criterion is satisfied
return *network*

BP 算法

对每一批次学习，将所有样本的梯度更新加和并执行一次权值更新



通常收敛很慢，且易陷入局部极小点

模式识别研究用计算机模拟生物、人的感知，对模式信息，如图像、文字、语音等，进行识别和分类。

传统人工智能的研究部分地显示了人脑的归纳、推理等智能。但是，对于人类底层的智能，如视觉、听觉、触觉等方面，现代计算机系统的信息处理能力还不如一个幼儿园的孩子。

神经网络模型模拟了人脑神经系统的特点：处理单元的广泛连接；并行分布式信息储存、处理；自适应学习能力等。

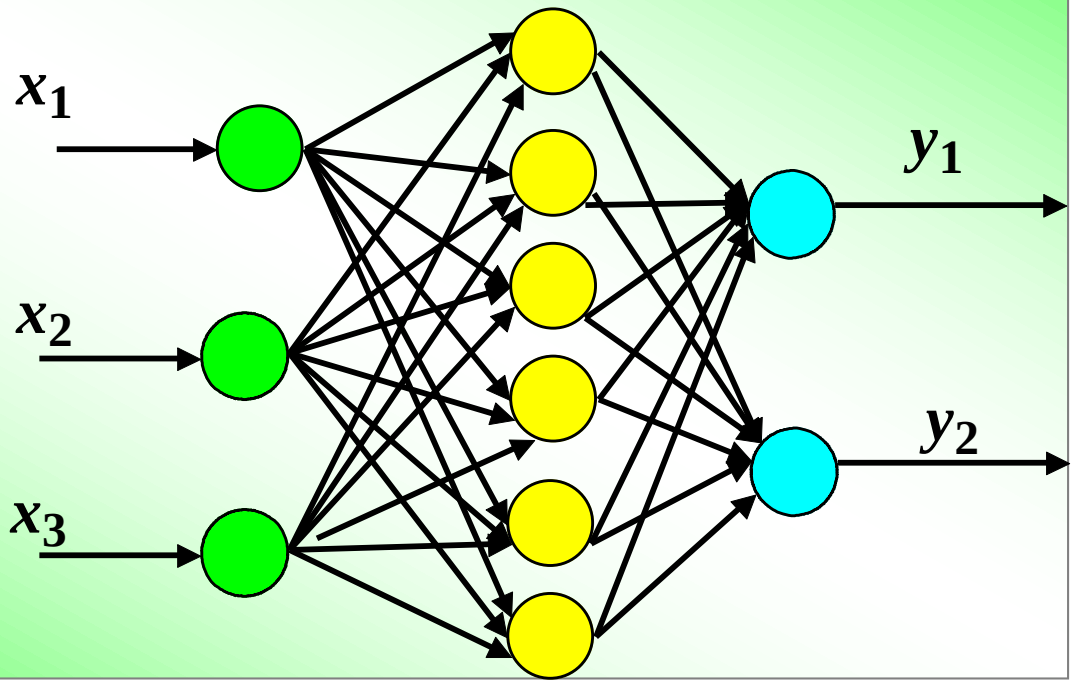
神经网络模式识别方法具有较强的容错能力、自适应学习能力、并行信息处理能力。

- 例 输入输出样本：

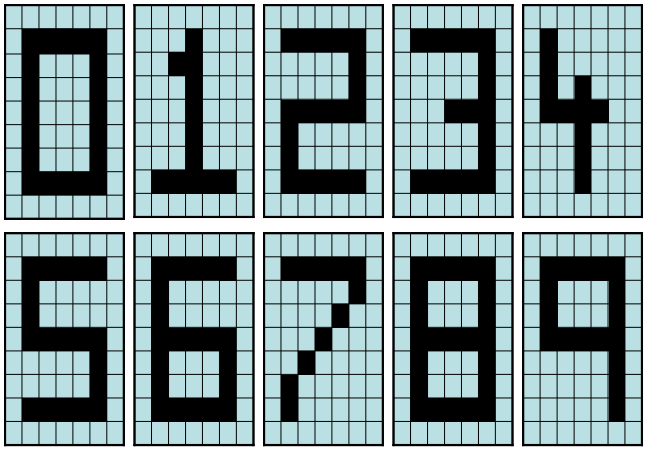
输入			输出	
1	0	0	1	0
0	1	0	0	0.5
0	0	1	0	1

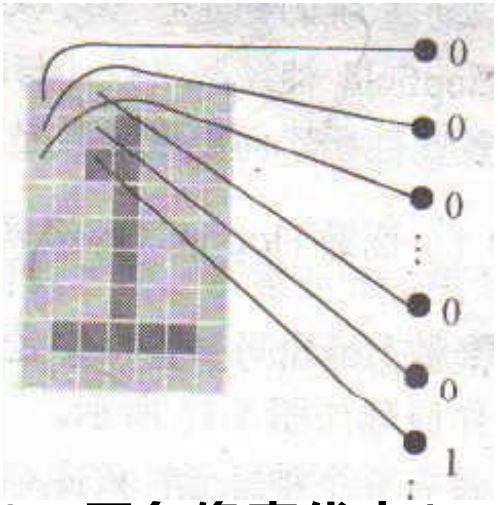
- 测试数据：

输入		
0.97	0.001	0.001
0	0.98	0
0.002	0	1.04
0.5	0.5	0.5
1	0	0
0	1	0
0	0	1



例 设计一个三层BP网络对数字0至9进行分类。



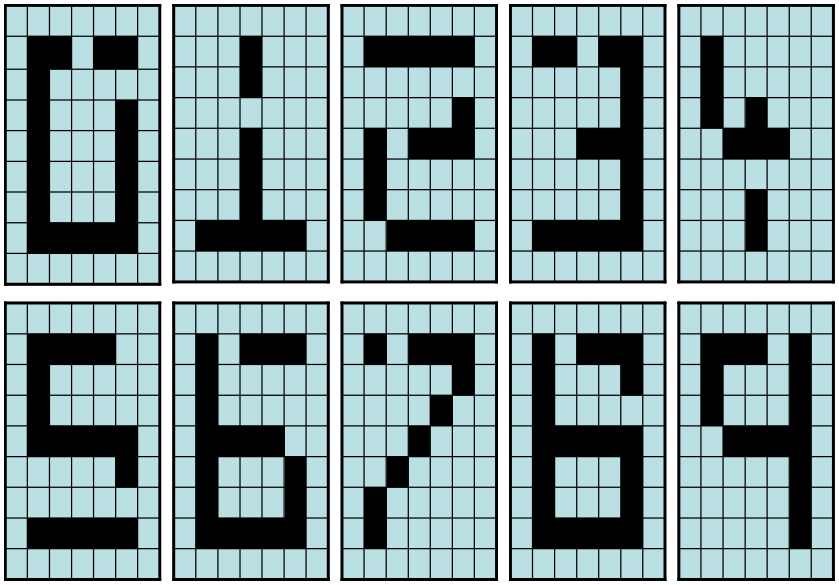


每个数字用9×7的网格表示，灰色像素代表0，黑色像素代表1。将每个网格表示为0，1的长位串。位映射由左上角开始向下直到网格的整个一列，然后重复其他列。

选择BP网络结构为63-6-9。97个输入结点，对应上述网格的映射。9个输出结点对应10种分类。

使用的学习步长为0.3。训练600个周期，如果输出结点的值大于0.9，则取为ON，如果输出结点的值小于0.1，则取为OFF。

当训练成功后，对如图所示测试数据进行测试。测试数据都有一个或者多个位丢失。



测试结果表明：除了8以外，所有被测的数字都能够被正确地识别。

对于数字8，神经网络的第6个结点的输出值为0.53，第8个结点的输出值为0.41，表明第8个样本是模糊的，可能是数字6，也可能是数字8，但也不完全确信是两者之一。

问题?