ARTIFICIAL INTELLIGENCE

2022/2023 Semester 2

Learning from examples: Chapter 18

提纲

- 相关概念
- 决策树学习
- 人工神经网络
- 深度学习入门

相关概念

➢什么是机器学习

- 学习是一个过程，通过学习可以对Agent的性能进行改进
- Simon：学习就是系统中的变化，这种变化使系统比以前更有效地去做同样的工作

相关概念

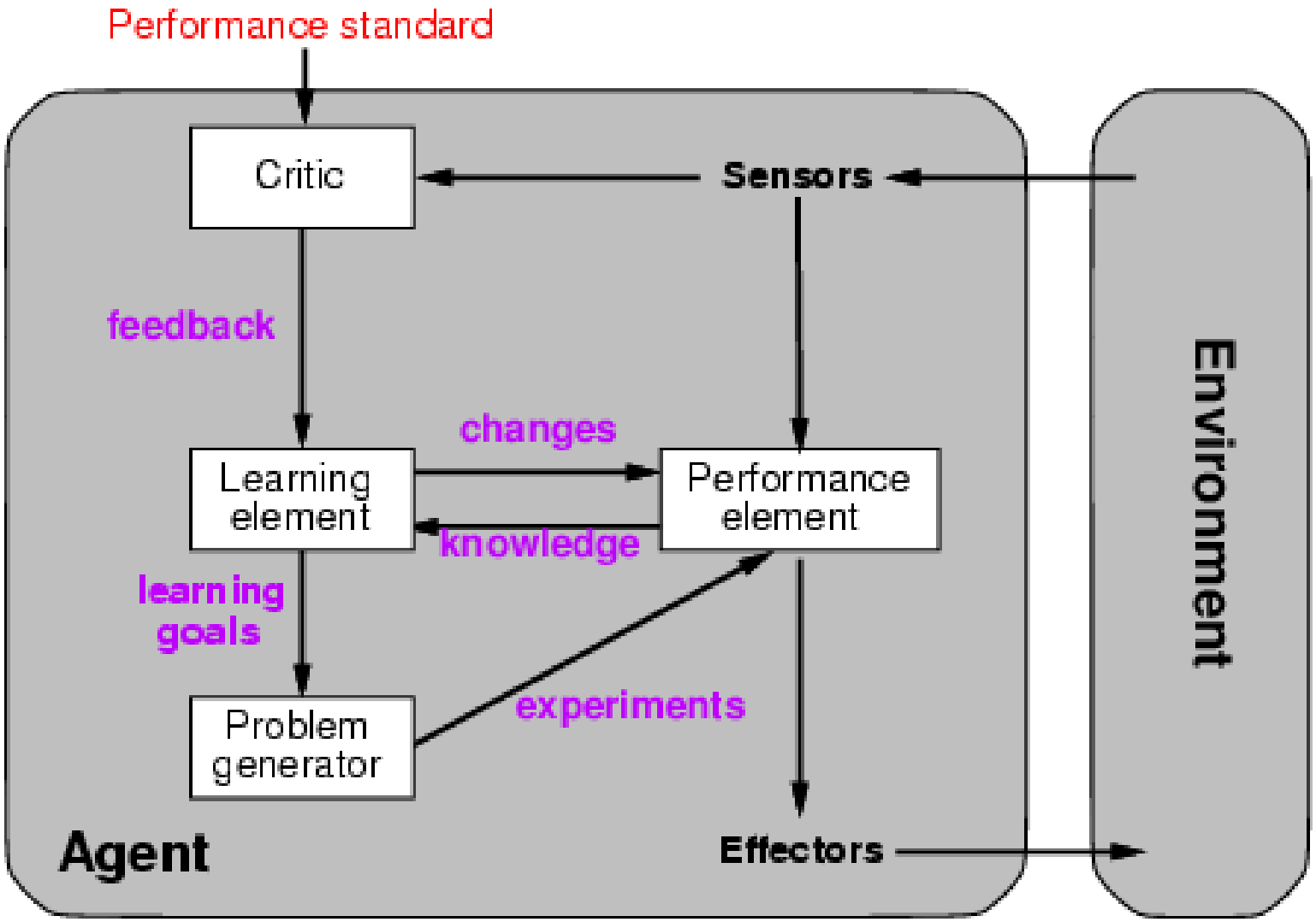
- 什么是机器学习

相关概念

➢ 学习的重要性

- 对未知的环境来讲，设计者缺少全知，学习是必要的
- 学习可以用作一种系统构建的方法
- 通过学习可以修改agent的决策机制，提高其性能， Agent任何部件的性能都可以通过从数据中进行学习来改进

相关概念


Performance standard
Critic
Sensors
feedback
changes
Learning element
Performance element
knowledge
learning goals
Problem generator
experiments
Agent
Effectors
Environment

相关概念

➢ 学习形式

- 智能体性能的改进以及改进所采用的技术依赖以下四个主要因素：

  – 要改进哪个部件?
  – Agent具备什么样的先验知识?
  – 数据和部件使用什么样的表示法?
  – 对学习可用的反馈是什么?

相关概念

➢ 要改进的部件

- 条件到动作的直接映射
- 从感知序列推演世界的合适特征的方法
- 关于世界进化方式的信息以及Agent能执行的可能行动的结果信息
- 表明世界状态的愿望的效用信息
- 表明动作愿望的动作-价值信息
- 描述能最大化成就Agent效用的状态类的目标

相关概念

➢ 表示和先验知识

- 问题求解智能体：状态空间
- 逻辑Agent：命题逻辑或一阶逻辑
- 决策理论智能体：贝叶斯网络进行推理

相关概念

- 对学习可用的反馈
- 有监督学习（Supervised learning）
  - 外部环境的反馈告知agent的期望输出是什么，或当前输出是否正确
  - 对每一个输入模式都有一个正确的目标输出
  - 从一组输入 - 输出的实例数据集中学习出一个函数（模型参数）
  - 当新的数据到来时，可以根据这个函数预测结果
  - 举例

相关概念

- 对学习可用的反馈

- 无监督学习（Unsupervised learning）
  - 没有来自外部环境的直接反馈（输出值），自组织学习输入的模式
  - 需要挖掘数据中的隐含规律
  - 举例

相关概念

- 对学习可用的反馈
- 强化学习（Reinforcement learning）
  - 外部环境仅给出一个对当前输出的一个评价（奖赏或惩罚信号），不会给出具体的期望输出
  - 系统从环境学习以使得奖励最大
  - 举例

相关概念

➢ 有监督学习

- 有监督学习的任务：

  给定一个训练集，N个样本，每个样本是一个输入 - 输出数据对： (x_1, y_1)， (x_2, y_2),...(x_n, y_n)，其中，$y=f(x)$，找到一个函数或假设h （hypothesis），使得 $h \approx f$

  x, y可以是任何形式的值。

- 通过测试集来验证它的泛化能力

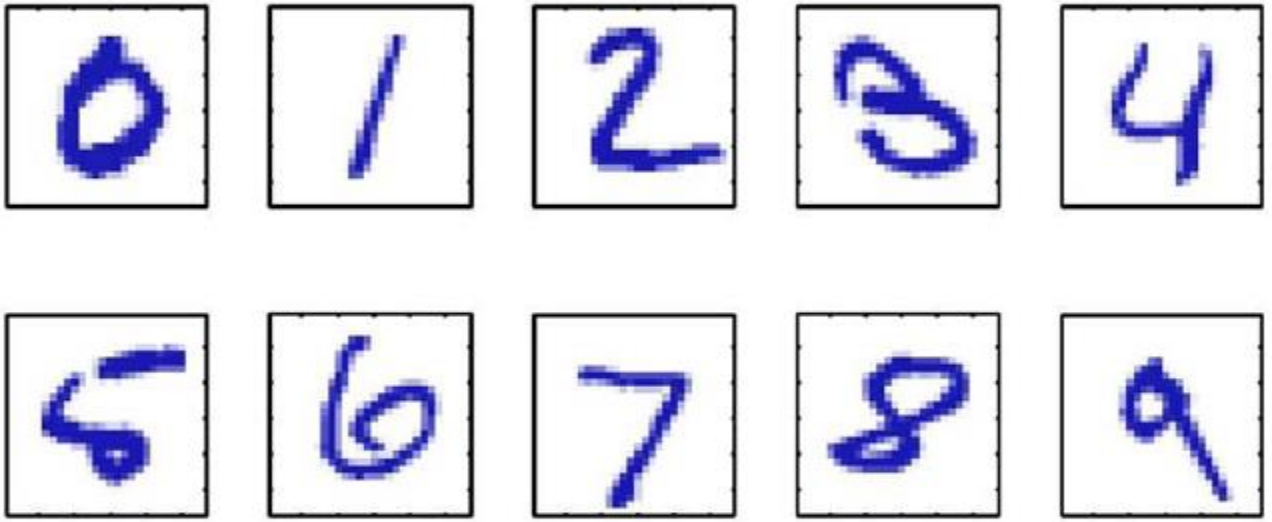
相关概念

➢ 有监督学习

- 分类学习问题：如果输出y是离散的值
- 回归学习问题：如果y是一个连续的数值
- 举例：预测天气

相关概念

Example 1: hand-written digit recognition



Images are 28 x 28 pixels

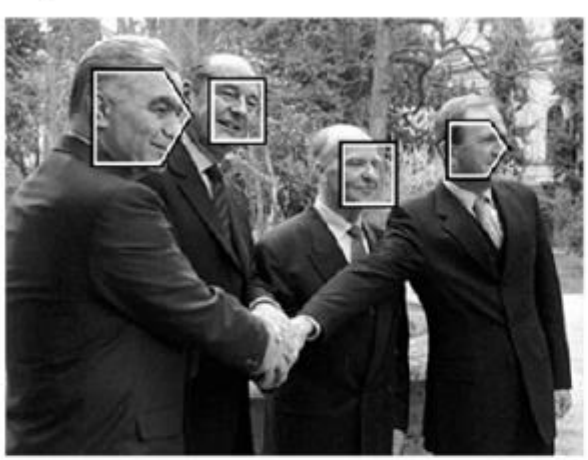
Represent input image as a vector $\mathbf{x} \in \mathbb{R}^{784}$

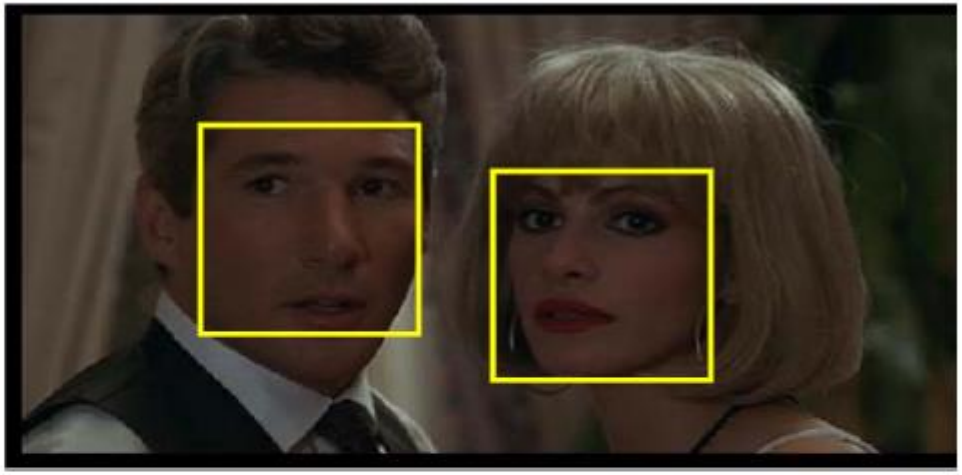
Learn a classifier $f(\mathbf{x})$ such that,

$$f : \mathbf{x} \rightarrow \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

相关概念

Example 2: Face detection

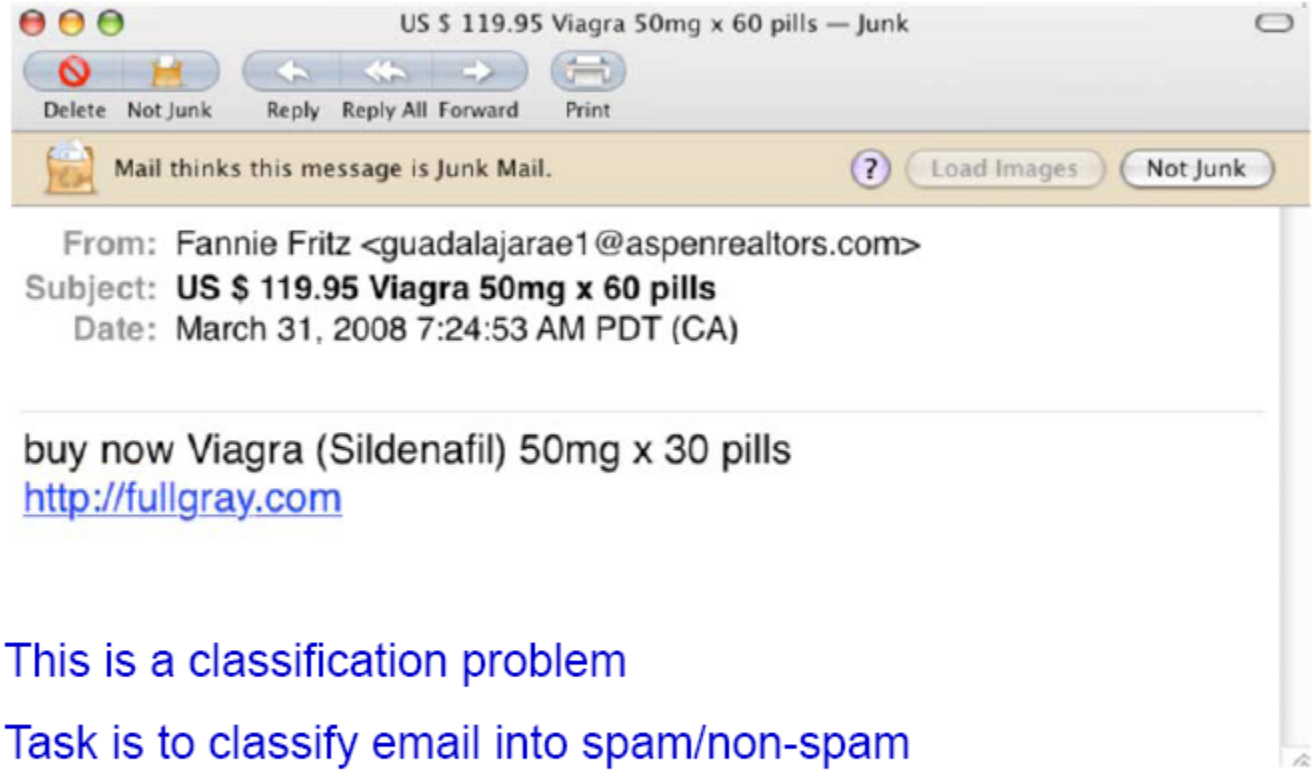




- Again, a supervised classification problem
- Need to classify an image window into three classes:
  - non-face
  - frontal-face
  - profile-face

相关概念

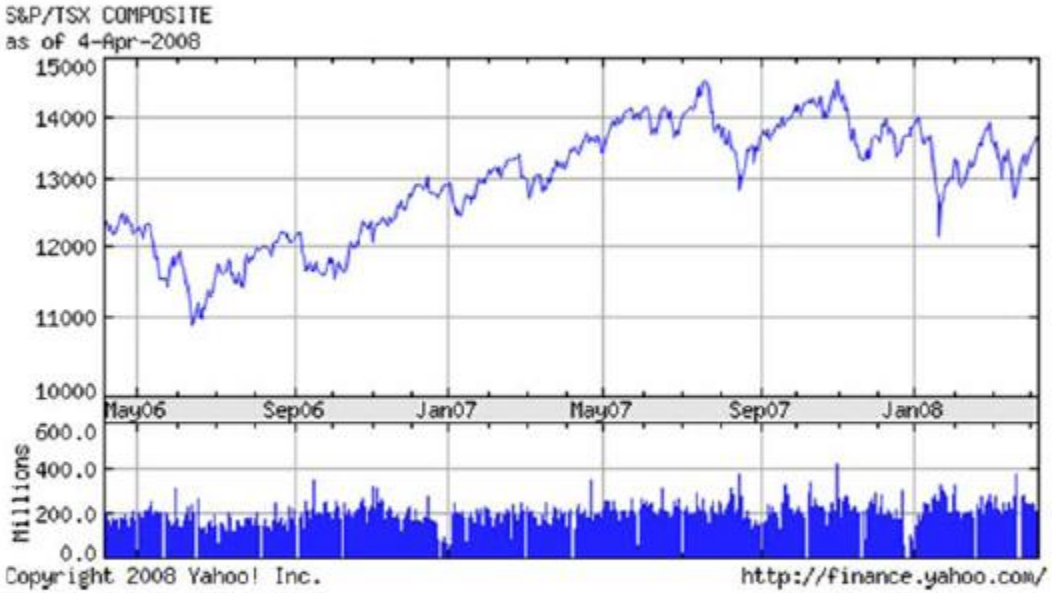
Example 3: Spam detection



- This is a classification problem
- Task is to classify email into spam/non-spam
- Data x_i is word count, e.g. of viagra, outperform, "you may be surprized to be contacted" …
- Requires a learning system as "enemy" keeps innovating

相关概念

Example 4: Stock price prediction



- Task is to predict stock price at future date
- This is a regression task, as the output is continuous

相关概念

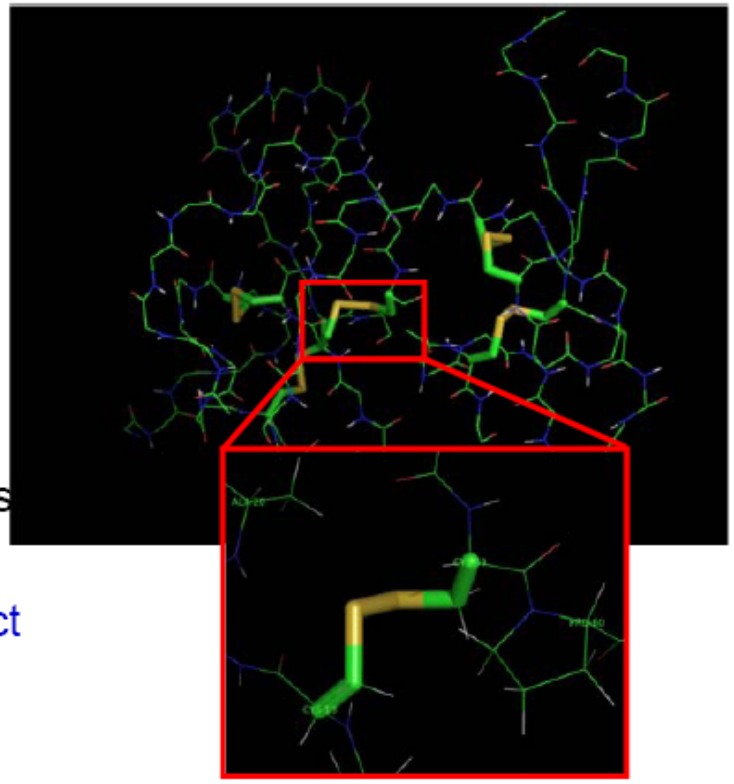
Example 5: Computational biology

x

AVITGACERDLQCG
KGTCCAVSLWIKSV
RVCTPVGTSGEDCH
PASHKIPFSGQRMH
HTCPCAPNLACVQT
SPKKFKCLSK

Protein Structure and Disulfide Bridges

y



Regression task: given sequence predict 3D structure

Protein: 1IMT

相关概念

➢ 有监督学习（续）

- 一致假设：如果一个假设h，对训练集中的每个x，h(x)=f(x)，就称 h 是一致的（consistent）

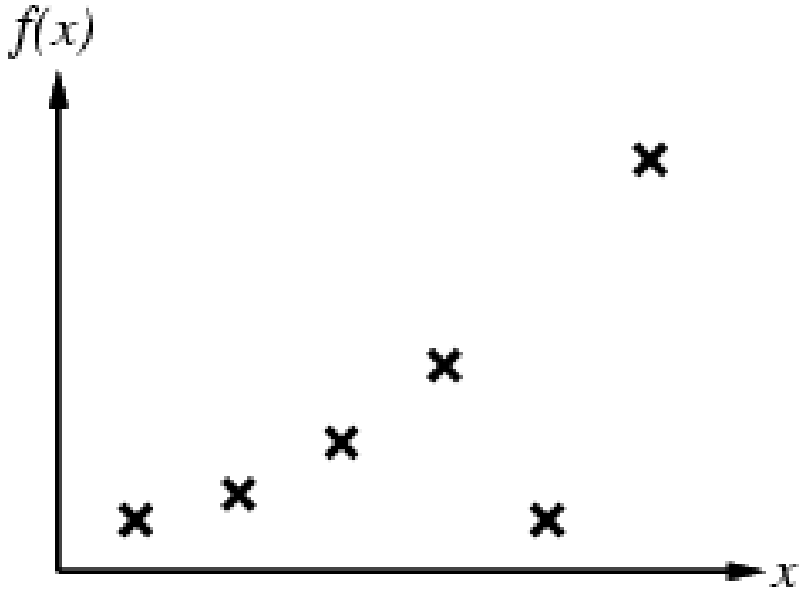
- 不同的假设可能拟合程度不一样，复杂程度也不一样

- 举例：曲线拟合

相关概念

- 有监督学习

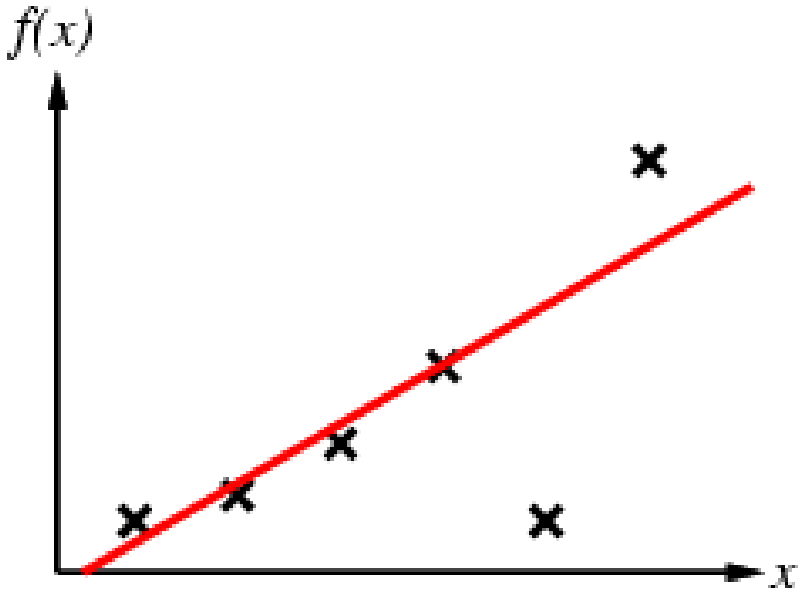
- 例如，曲线拟合：



相关概念

➢ 有监督学习

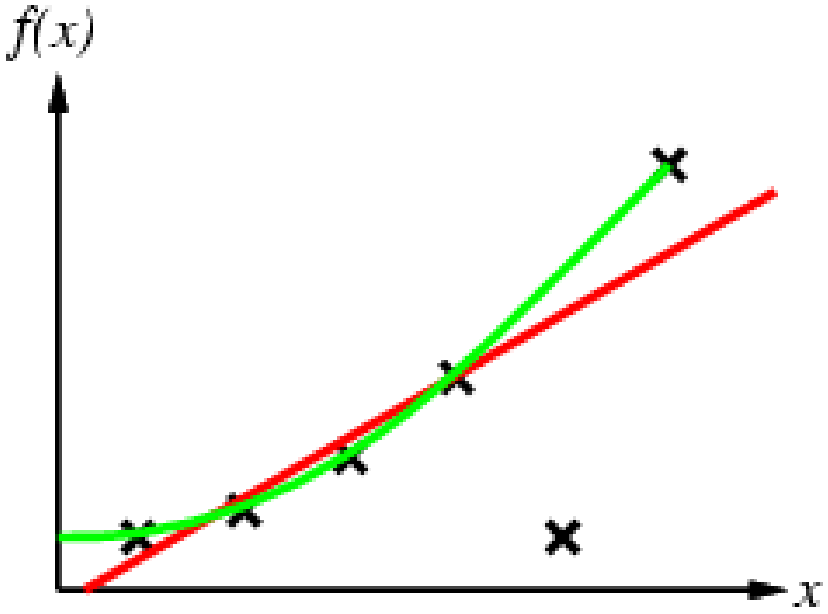
- 例如，曲线拟合：



相关概念

- 有监督学习

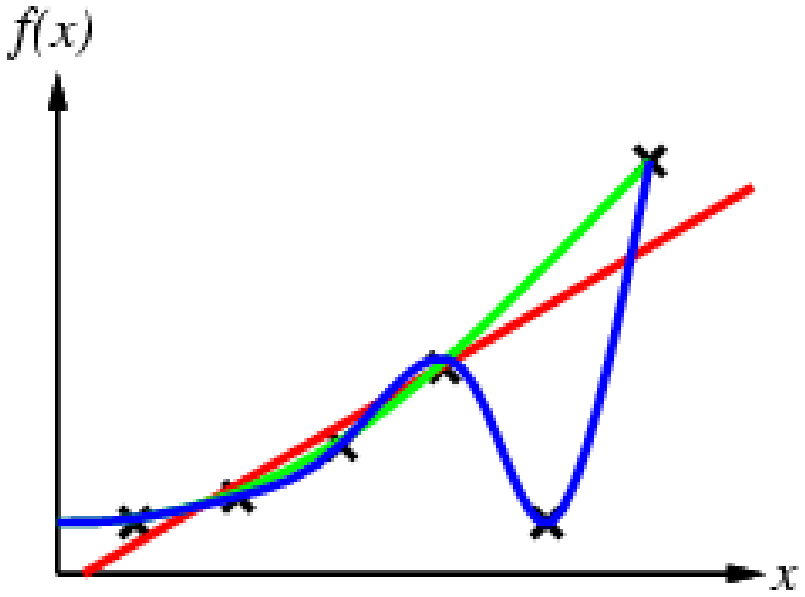
- 例如，曲线拟合：



相关概念

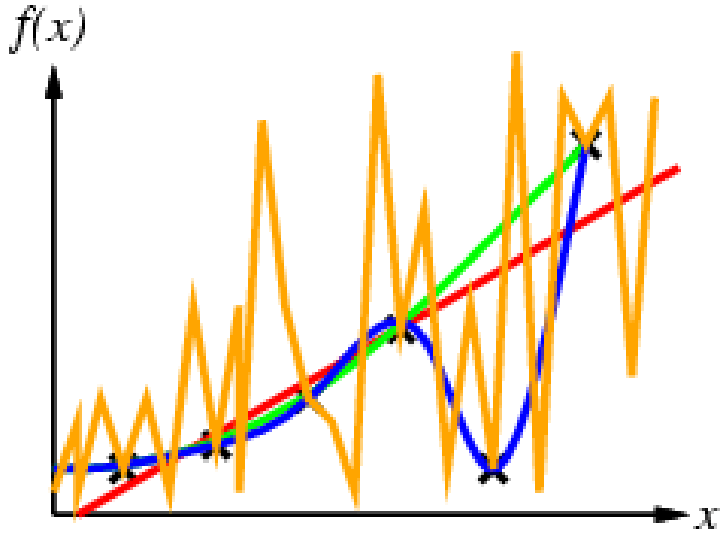
- 有监督学习

- 例如，曲线拟合：



相关概念

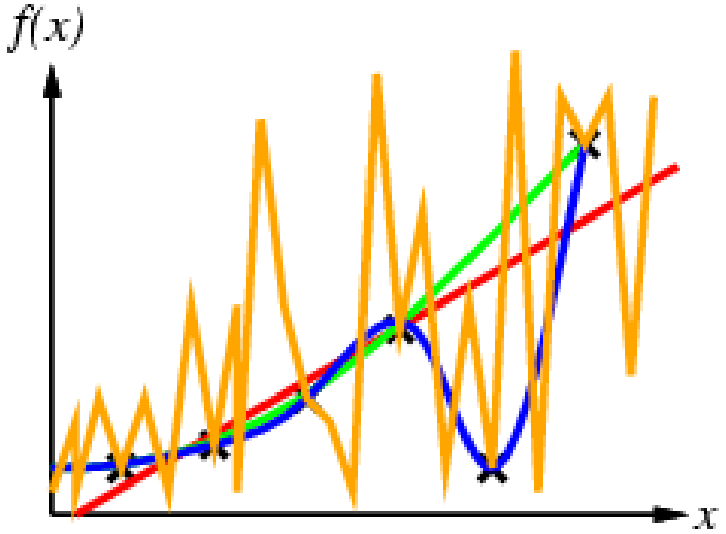
- 有监督学习
- 例如，曲线拟合：



相关概念

➢ 有监督学习

- 例如，曲线拟合：



- 奥卡姆剃刀（Ockham’s razor）原则：优先选择与数据一致的最简单的假设

提纲

- 相关概念
- 决策树学习
- 人工神经网络
- 深度学习入门

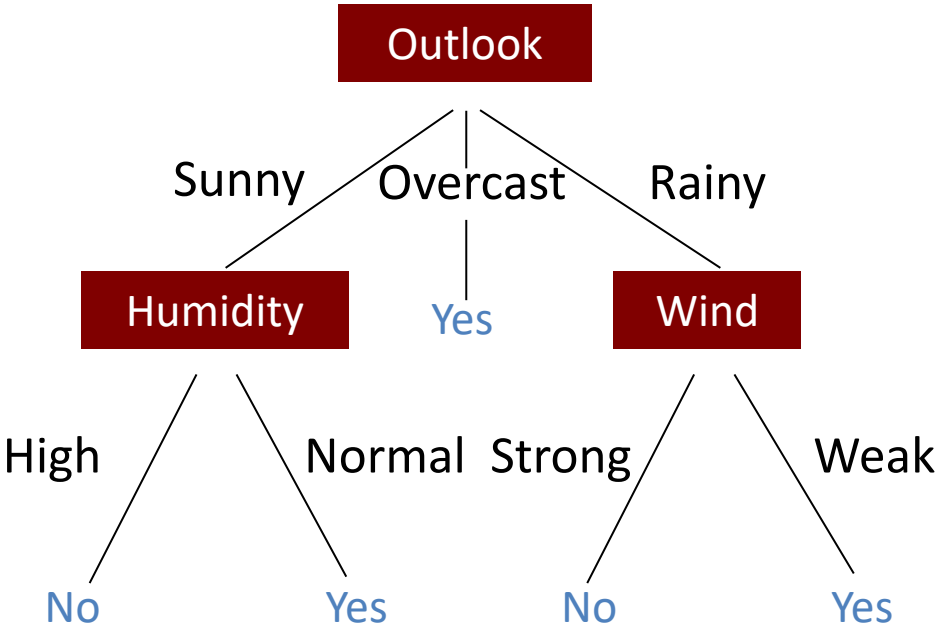
决策树学习

➢ 概述

- 是一种最简单但应用广泛的有监督学习算法
- 将从一组训练数据中学习到的函数被表示为一棵树
- 决策树通过把实例从根节点排列到某个叶子节点来分类实例：
  - 树上的每一个节点指定了对实例的某个属性的测试
  - 节点的每一个后继分支对应于该属性的一个可能值。
  - 叶子节点即为实例所属的分类。

决策树学习

➢ 决策树例子



Play Tennis的决策树

该模型是一棵树：

✓ 每次根据一个特征（或属 性）将样本分成若干部分；

✓ 当不可再分时（叶节点），给出一个决策：

- 通常输出的决策是类标签
- 也可以输出一个概率分布

决策树学习

➢ 决策树实例问题

基于下面的属性，决定一下是否要在餐馆等座位

1. Alternate（候选）：附近是否有另一家合适的餐馆?
2. Bar（酒吧）：该餐馆中供顾客等候的吧区是否舒适?
3. Fri/Sat（周五/周六）若是周五或周六，则为真
4. Hungry（饥饿）我们是否饥饿?
5. Patrons(顾客）：该餐馆中有多少顾客(None, Some, Full)
6. Price（价格）：餐馆的价格范围($, $$, $$$)
7. Raining（下雨）外面是否在下雨?
8. Reservation（预约）：我们是否预约过?
9. Type（类型）：餐馆的种类(French, Italian, Thai, Burger)
10. WaitEstimate（等候时间估计）：估计的等候时间(0-10, 10-30, 30-60, >60)

决策树学习

- 基于属性的表示
- 决定是否要在餐馆等座位的实例集
- 实例是通过属性值描述的

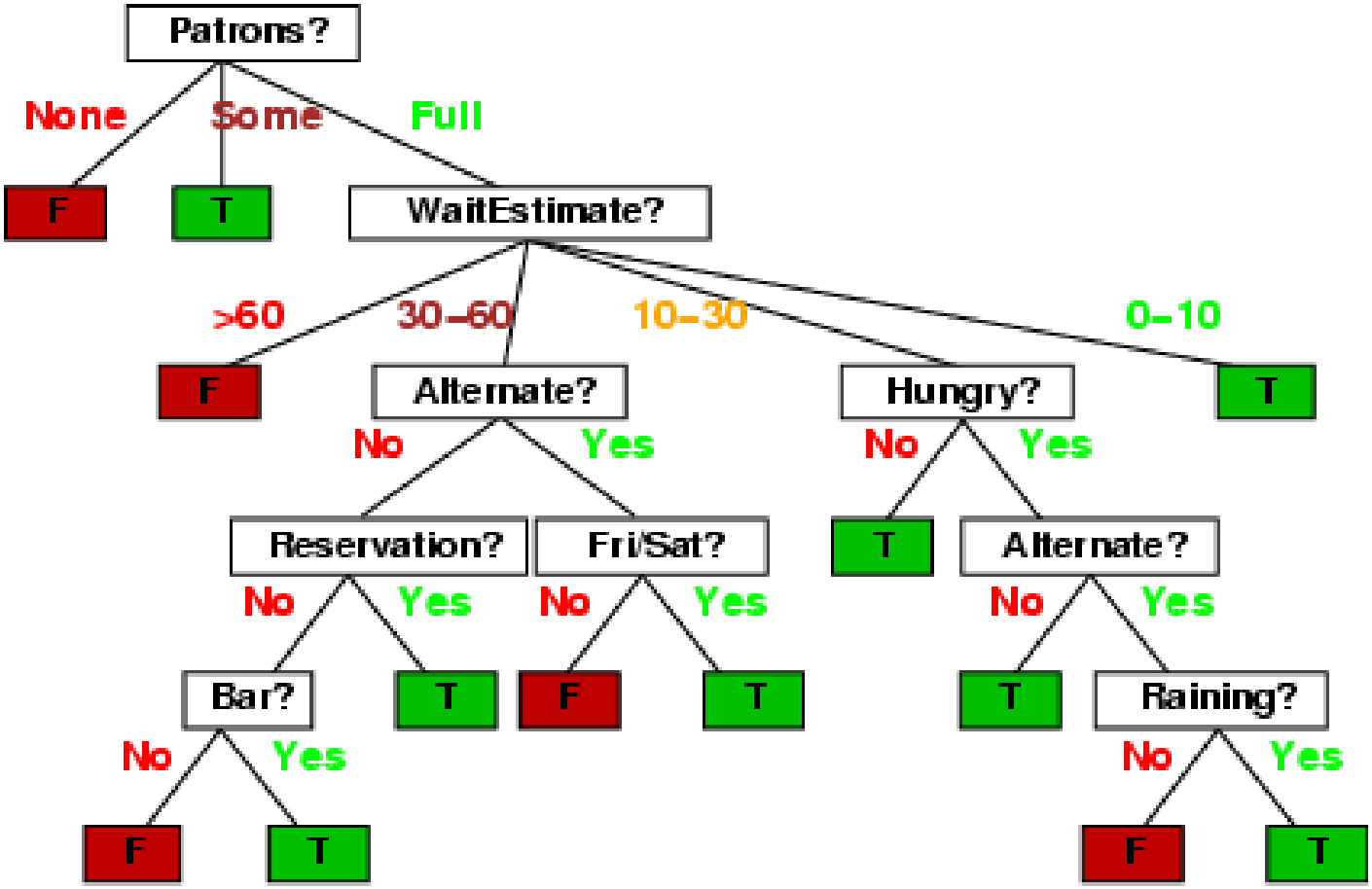
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	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

- 实例分类： 正 (T) 或 负 (F)

决策树学习

➢ 决策树

- 假设空间的一种可能表示
- 实例从树的根节点开始处理，并沿着适当的分支到达叶节点

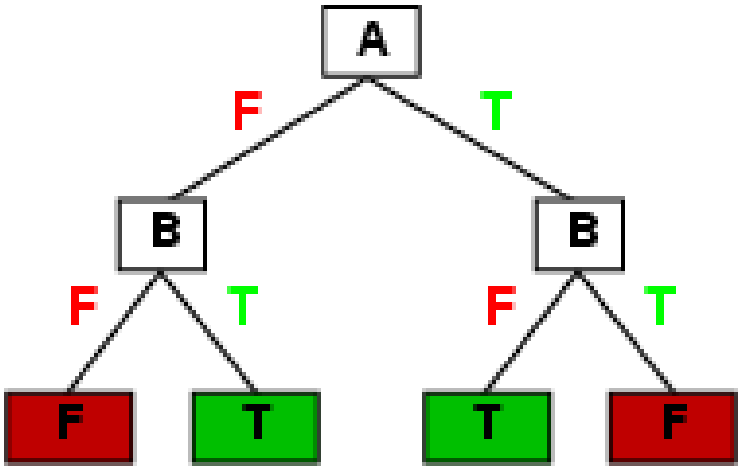


决策树学习

➢ 决策树的表示能力

- 决策树能表示输入属性的任何函数
- 对布尔型函数, 真值表中每一行对应树中到叶节点的一条路径:

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



- n个布尔属性， 真值表有 2^n 行，有 2^{2^n}个函数
- 决策树个数一般等于或大于布尔函数的个数
- 例：6个布尔属性， 有18,446,744,073,709,551,616 个不同的函数

决策树学习

➢ 如何找到一致的假设

- 学习：寻找和完整的实例集合（训练集）一致的决策树
- 针对每个实例，树中都有一条到达叶节点的路径，可以得到一棵一致的决策树，但可能不能推广到新的样例上。
- 需要寻找更“紧凑” (compact)的决策树

决策树学习

➢ 决策树学习要点

- 目标: 找到和训练集一致的较小的树（树中所有的路径都很短，整棵树的规模比较小）
- 思想: 递归地选择“最好”或“最佳”的属性作为树或子树的根，通过较少数量的测试就能得到正确的分类
- 最好：分类能力最好

决策树学习

- 决策树学习算法

```
function DECISION-TREE-LEARNING(examples, attributes, default) returns a decision tree
  inputs: examples, set of examples
          attributes, set of attributes
          default, default value for the goal predicate

  if examples is empty then return default
  else if all examples have the same classification then return the classification
  else if attributes is empty then return MAJORITY-VALUE(examples)
  else
      best ← CHOOSE-ATTRIBUTE(attributes, examples)
      tree ← a new decision tree with root test best
      for each value v_i of best do
          examples_i ← {elements of examples with best = v_i}
          subtree ← DECISION-TREE-LEARNING(examples_i, attributes − best,
                                           MAJORITY-VALUE(examples))
          add a branch to tree with label v_i and subtree subtree
      end
      return tree
```

决策树学习

➢ 决策树学习算法 - 结束条件

①如果所有剩余的实例属于同一类。

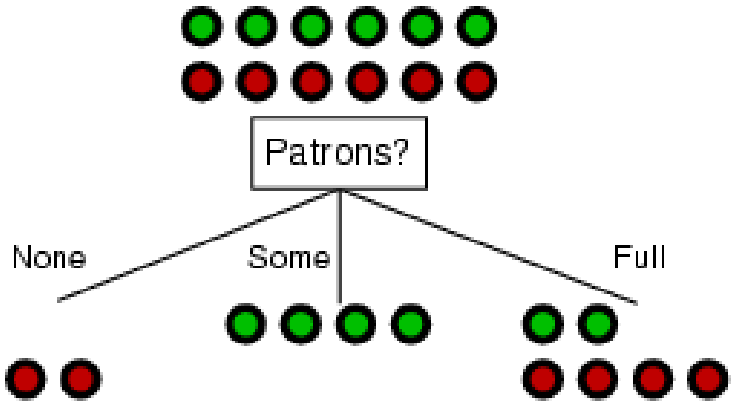
②没有剩下的实例，则返回一个缺省值，以父节点中实例的多数类别创建一个树叶。

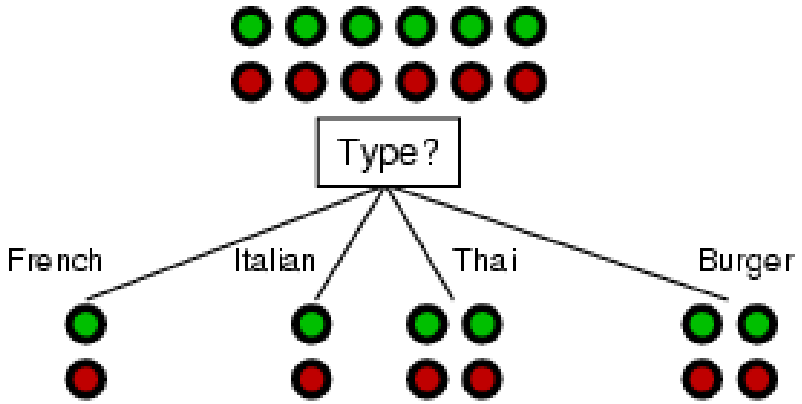
③没有剩余属性可以用来进一步划分实例。在这种情况下，使用多数表决，将给定的节点转换成树叶，并以实例中元组个数最多的类别作为类别标记，同时也可以存放该节点样本的类别分布。

决策树学习

- 选择一个属性

- 思想: 理想的属性是将实例分为只包含正例或只包含反例的集合





- *Patrons?* 不理想，但是很不错

决策树学习

- 如何选择属性

- 使用信息论中信息增益的概念
- 信息增益通过信息内容或熵来定义
- 熵（Entropy）:是随机变量的不确定性度量。信息的获取对应熵的减少，不确定性越小，熵越小

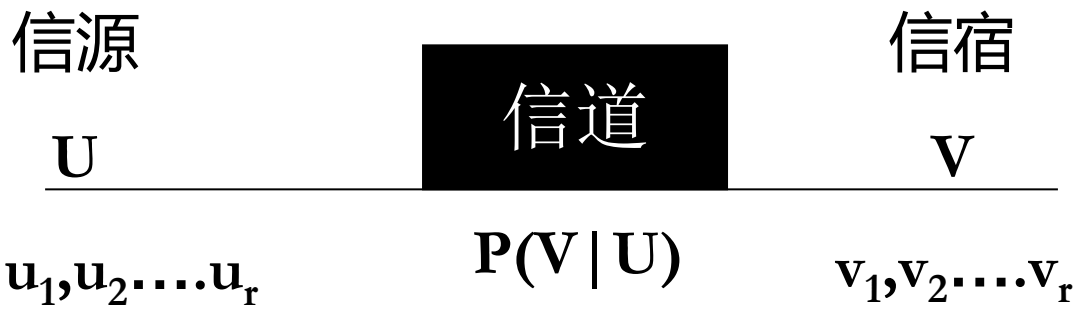
决策树学习

➢ 信息论原理

信息论是C.E.Shannon为解决信息传递（通信）过程问题而建立的理论，也称为统计通信理论。

- 信道模型

一个传递信息的系统是由发送端（信源）和接收端（信宿）以及连接两者的通道（信道）三者组成。



决策树学习

- 在进行实际的通信之前，收信者（信宿）不可能确切了解信源究竟会发出什么样的具体信息，不可能判断信源会处于什么样的状态。

- 这种情形就称为信宿对于信源状态具有不确定性。而且这种不确定性是存在于通信之前的。因而又叫做先验不确定性，表示成

信息熵 H（U）

决策树学习

- 在进行了通信之后，信宿收到了信源发来的信息，这种先验不确定性才会被消除或者被减少。
- 如果干扰很小，不会对传递的信息产生任何可察觉的影响，信源发出的信息能够被信宿全部收到，在这种情况下，信宿的先验不确定性就会被完全消除。

决策树学习

- 在一般情况下，干扰总会对信源发出的信息造成某种破坏，使信宿收到的信息不完全。
- 先验不确定性不能全部被消除，只能部分地消除。
- 通信结束之后，信宿仍然具有一定程度的不确定性。这就是后验不确定性，用条件熵表示**H（U/V）**。
- 后验不确定性总要小于先验不确定性:

$$H(U/V) < H(U)$$

决策树学习

- 如果后验不确定性的大小正好等于先验不确定性的大小，这就表示信宿根本没有收到信息。
- 如果后验不确定性的大小等于零，这就表示信宿收到了全部信息。
- 可见，信息是用来消除（随机）不确定性的度量。信息量用互信息来表示，即：

$$I(U, V) = H(U) - H(U/V)$$

决策树学习

➢ 信息增益（互信息）的计算

1．定义

（1）设S为训练集，有n个特征（属性），表示为（A_1，A_2，…，，A_n）。｜S｜表示例子总数。

（2）S中有U_1，U_2两类。｜U_i｜表示U_i类例子数。

（3）特征A_k处有m个取值，分别为（V_1，V_2，…，，V_m）。

2．U_i类出现概率为:

$$P(U_i) = |U_i| / |S| \qquad (1)$$

自然有 $\sum_{i=1}^{2} P(U_i) = 1$

决策树学习

3. U_i类中在特征A_k处取值V_j的例子集合V_{ij}的条件概率为:

$$P(V_j \mid U_i) = |V_{ij}| / |U_i| \qquad (2)$$

自然有 $\sum_{j=1}^{m} P(V_j|U_i) = 1$

4. 在特征A_k处，取V_j值的例子集合的概率为:

$$P(V_j) = |V_j| / |S| \qquad (3)$$

自然有 $\sum_{j=1}^{m} P(V_j) = 1$

决策树学习

5．在特征A_k处取V_j值的例子，属于U_i类的例子集合U_{ij}的条件概率为:

$$P(U_i \mid V_j) = |U_{ij}| / |V_j| \qquad (4)$$

自然有 $$\sum_{i=1}^{2} P(U_i \mid V_j) = 1$$

决策树学习

6．信息熵

(1) 消息传递系统由消息的发送端（信源）和接收端（信宿）以及连接两者的通道（信道）三者组成。

(2) 消息（符号）U_i（i=1，2，…，q）的发生概率P（U_i）组成信源数学模型（样本空间或概率空间）

$$[U,P]=\begin{bmatrix} U_1 & U_2 & \cdots & U_q \\ P(U_1) & P(U_2) & \cdots & P(U_q) \end{bmatrix} \qquad (5)$$

决策树学习

（3）**自信息**:消息U_i发生后所含有的信息量。它反映了消息U_i发生前的不确定性（随机性）。定义为:

$$I(U_i) = log\frac{1}{P(U_i)} = -log\,P(U_i) \qquad (6)$$

以2为底，所得的信息量单位为bit。以e为底，所得的信息量单位为nat.

（4）**信息熵**:自信息的数学期望。即信源输出后，每个消息所提供的信息量，也反映了信源输出前的平均确定性。定义为:

$$H(U) = \sum_i P(U_i)\log\frac{1}{P(U_i)} = -\sum_i P(U_i)\log P(U_i) \qquad (7)$$

决策树学习

例如:两个信源，其概率空间分别为:

$$\begin{array}{c|cc} X & a_1 & a_2 \\ P(X) & 0.99 & 0.01 \end{array} \qquad \begin{array}{c|cc} Y & b_1 & b_2 \\ P(Y) & 0.5 & 0.5 \end{array}$$

则信息熵分别为:

H（X）= - 0.99 log0.99 - 0.01 log0.01 = 0.08 bit

H（Y）= - 0.5 log0.5 - 0.5 log0.5 = 1bit

可见　　H（Y）>H（X）

故信源Y比信源X的平均不确定性要大。

决策树学习

信息熵H（U）是信源输出前的平均不确定性，也称先验熵。

H（U）的性质:

（1）H（U）=0时，说明只存在着唯一的可能性，不存在不确定性。

（2）如果n种可能的发生都有相同的概率，即所有的U_i有P（U_i）=1/n，H（U）达到最大值log n，系统的不确定性最大。

P（U_i）互相接近，H（U）就大。P（U_i）相差大，则H（U）就小。

决策树学习

7．信息增益

(1)后验熵和条件熵

当没有接收到输出符号V时，已知输入符号U的概率分布为P（U），而当接收到输出符号V=V_j 后，输入符号的概率分布发生了变化,变成后验概率分布P（U|V_j）。其后验熵为：

$$H(U|V_j)=\sum_i P(U_i|V_j)\log\frac{1}{P(U_i|V_j)}$$

那么接收到输出符号V=V_j后，关于U的平均不确定性为：

$$H(U|V)=\sum_j P(V_j)\sum_i P(U_i|V_j)\log\frac{1}{P(U_i|V_j)}$$

这是接收到输出符号V_j后关于U的条件熵

决策树学习

这个条件熵称为信道疑义度。它表示在输出端收到全部输出符号 *V*后，对于输入端的符号集 *U*尚存在的不确定性（存在疑义）。

从上面分析可知：条件熵小于无条件熵，即

$$H(U|V) < H(U)。$$

说明接收到符号集 *V*的所有符号后，关于输入符号 *U*的平均不确定性减少了。即总能消除一些关于输入端 *X*的不确定性，从而获得了一些信息。

决策树学习

(2) 信息增益

定义:

$$I(U,V) = H(U) - H(U|V) \quad (8)$$

I（U,V）称为U和V之间的平均互信息.它代表接收到符号集V后获得的关于U的信息量。

可见，熵（H（U）、H（U|V））只是平均不确定性的描述。熵差（H（U） – H（U|V））是不确定性的消除，即互信息才是接收端所获得的信息量。

对输入端U只有U_1，U_2两类，互信息的计算公式为:

$$H(U) = \sum_{i=1}^{2} P(U_i) \log \frac{1}{P(U_i)}$$

$$H(U|V) = \sum_{j=1}^{m} \sum_{i=1}^{2} P(V_j) P(U_i|V_j) \log \frac{1}{P(U_i|V_j)}$$

$$I(U,V) = H(U) - H(U|V)$$

决策树学习

➢ 基于信息增益的决策树构造算法（ID3算法）

- ID3算法是当前国际上最有影响的示例学习方法
- ID3算法引进了信息论中的信息增益（互信息）作为特征判别能力的度量，并且将建树的方法嵌在一个迭代的外壳之中
- 首先找出最有判别力的特征，把数据分成多个子集，每个子集又选择最有判别力的特征进行划分，一直进行到所有子集仅包含同一类型的数据为止。最后得到一棵决策树。

决策树学习

➢ 建树算法

- 对当前例子集合，计算各特征的互信息；

选择互信息最大的特征A_k；

- 把在A_k处取值相同的例子归于同一子集，A_k取几个值就得几个子集；
- 对既含正例又含反例的子集，递归调用建树算法；
- 若子集仅含正例或反例，对应分枝标上P或N，返回调用处。

NO.	属性				类别
	天气	气温	湿度	风	
1	晴	热	高	无风	N
2	晴	热	高	有风	N
3	多云	热	高	无风	P
4	雨	适中	高	无风	P
5	雨	冷	正常	无风	P
6	雨	冷	正常	有风	N
7	多云	冷	正常	有风	P
8	晴	适中	高	无风	N
9	晴	冷	正常	无风	P
10	雨	适中	正常	无风	P
11	晴	适中	正常	有风	P
12	多云	适中	高	有风	P
13	多云	热	正常	无风	P
14	雨	适中	高	有风	N

决策树学习

➢ 实例计算

对于气候分类问题进行具体计算有：

1 信息熵的计算

信息熵：$$H(U) = -\sum_i P(u_i)\log P(u_i)$$

类别出现概率：$$P(u_i) = \frac{|u_i|}{|S|}$$

|S|表示例子集*S*的总数，$|u_i|$表示类别u_i的例子数。

对9个正例和5个反例有：

P（u_1）=9/14　　P（u_2）=5/14

H（U）=（9/14）log（14/9）+（5/14）log（14/5）=0.94bit

决策树学习

2 条件熵计算

条件熵： $H(U/V)=-\sum_j P(v_j)\sum_i P(u_i/v_j)\log P(u_i/v_j)$

属性A_1取值v_j时，类别u_i的条件概率：$P(u_i/v_j)=\frac{|u_i|}{|v_j|}$

A_1=天气 取值 v_1=晴，v_2=多云，v_3=雨

在A_1处取值晴的例子5个，取值多云的例子4 个，取值雨的例子5 个，故：

P（v_1）=5/14 P（v_2）=4/14 P（v_3）=5/14

取值为晴的5 个例子中有2 个正例、3个反例，故：

P（u_1/v_1）=2/5， P（u_2/v_1）=3/5

同理有：P（u_1/v_2）=4/4， P（u_2/v_2）=0

P（u_1/v_3）=2/5， P（u_2/v_3）=3/5

H(U/V)=(5/14)((2/5)log(5/2)+(3/5)log(5/3))+(4/14)((4/4)log(4/4)+0)+(5/14)((2/5)log(5/2)+(3/5)log(5/3)) = 0.694bit

决策树学习

3 互信息计算

对 A1=天气 处有:

I（天气）=H（U）- H（U|V）= 0.94 - 0.694 = 0.246 bit

类似可得:

I（气温）=0.029 bit

I（湿度）=0.151 bit

I（风）=0.048 bit

4 建决策树的树根和分枝

ID3算法将选择互信息最大的特征天气作为树根，在14个例子中对天气的3个取值进行分枝，3 个分枝对应3 个子集，分别是:

F1={1，2，8，9，11}，F2={3，7，12，13}，F3={4，5，6，10，14}

其中F2中的例子全属于P类，因此对应分枝标记为P，其余两个子集既含有正例又含有反例，将递归调用建树算法。

决策树学习

5 递归建树

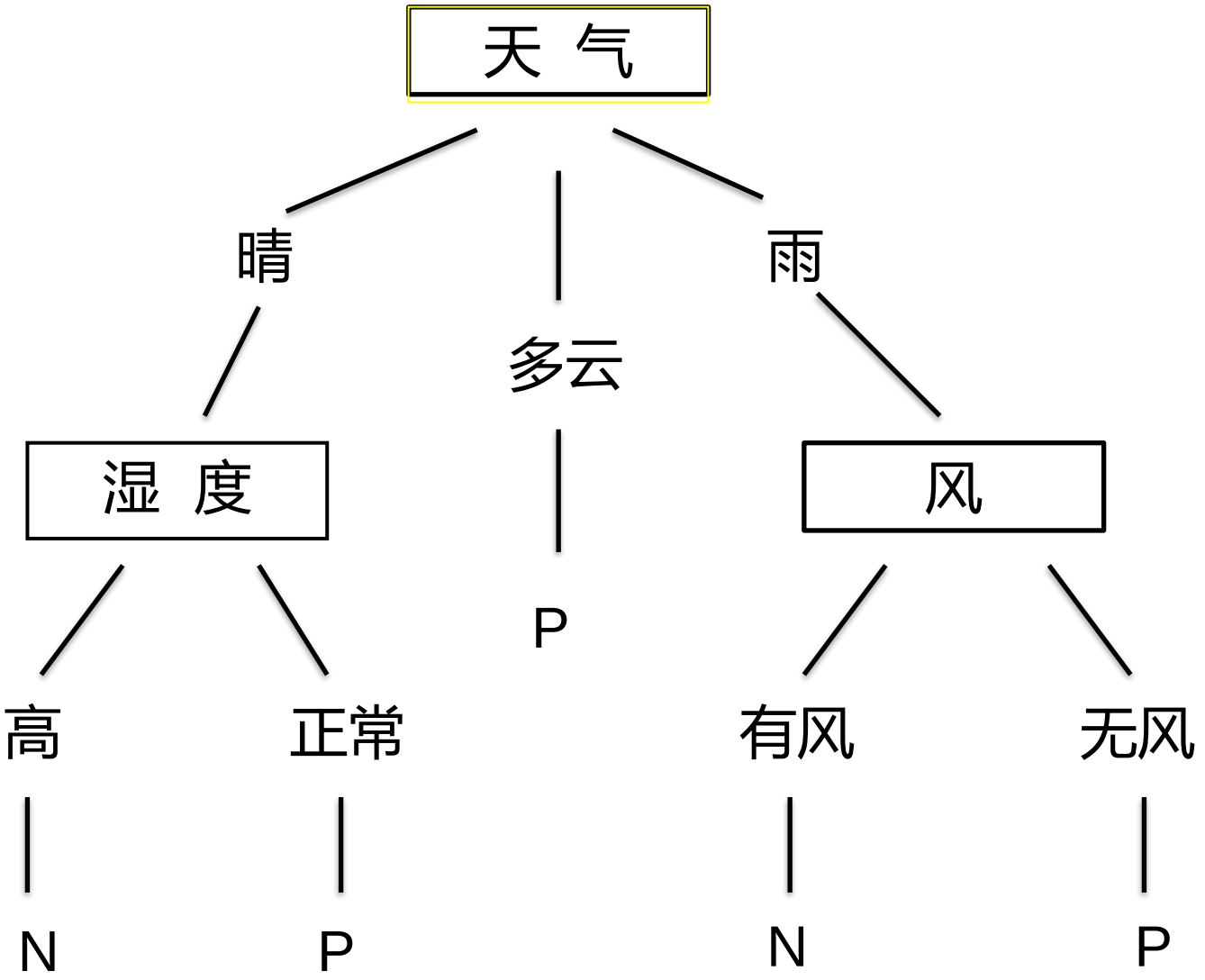
分别对F1和F3子集利用ID3算法，在每个子集中对各特征求互信息.

（1）F1中的天气全取晴值，则H（U）=H（U|V），有I（U|V）=0，在余下三个特征中求出湿度互信息最大，以它为该分枝的根结点，再向下分枝。湿度取高的例子全为N类，该分枝标记N。取值正常的例子全为P类，该分枝标记P。

（2）在F3中，对四个特征求互信息，得到风特征互信息最大，则以它为该分枝根结点。再向下分枝，风取有风时全为N类，该分枝标记N。取无风时全为P类，该分枝标记P。

这样就得到下图的决策树

决策树学习



ID3决策树

决策树学习

➢ 餐馆座位问题实列

对于餐馆是否等座位的数据集, $p = n = 6$, H（U）$= 1$ bit

考虑Patrons 和 Type 两个属性:

I(Patrons)= H（U）- H（U|V）=1-0.46= 0.54

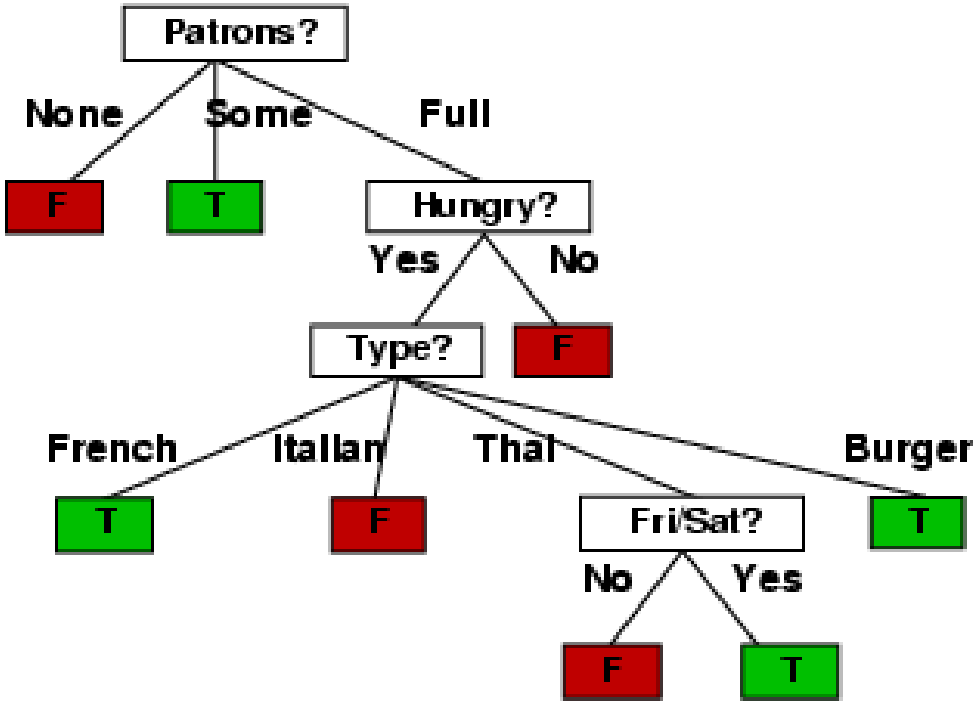
I(Type)= H（U）- H（U|V）=1-1= 0

其他的属性的信息增益也可以进行类似的计算

结论：Patrons 具有最高的信息增益，被决策树算法选为决策树的根

决策树学习

- 从12个实例中学习到的决策树:



- 与训练数据集一致；比原始树要简单得多。

决策树学习

➢ 对应的逻辑表达式

- 从根节点到叶节点的每一条路经都对应着一条合理的规则，规则间各个部分的关系是合取关系。整个决策树就对应着一组析取的规则。

- 该决策树对应以下逻辑表达式：

（ Patrons=Some) ∨ (Patrons=Full ∧ Hungry=Yes ∧ Type=French) ∨ (Patrons= Full ∧ Hungry=Yes ∧ Type=Burger) ∨ (Patrons= Full ∧ Hungry=Yes ∧ Type=That ∧ Fri/Sat=Yes)

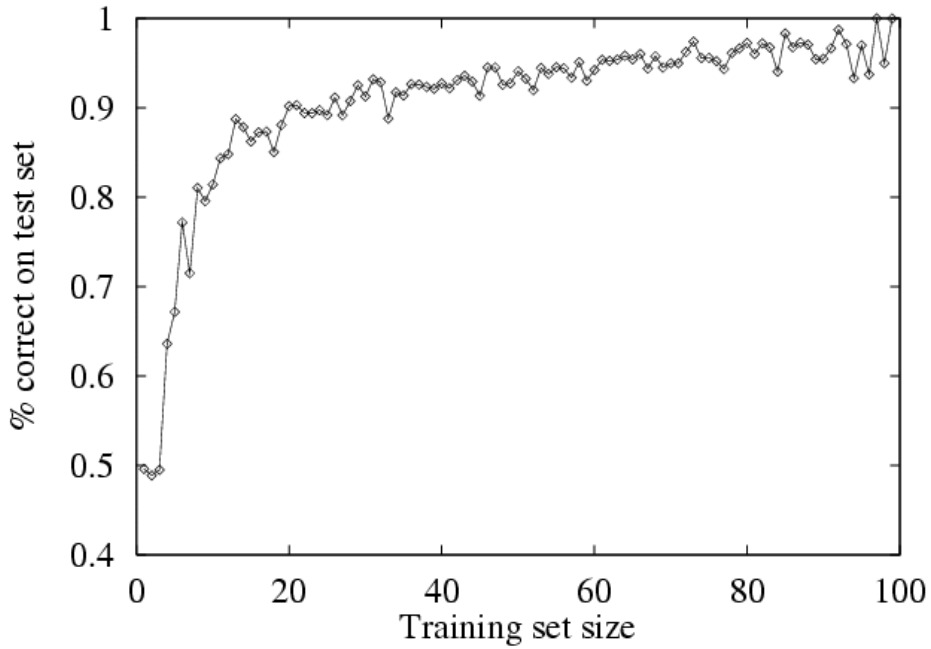
决策树学习

- **学习算法的性能评估**

- 如何知道 $h \approx f$？ h是好是坏?

1. 使用计算 /统计学习理论
2. 在新的测试集上进行评估（分类预测未见的实例）
   (使用和训练集的样例空间相同的分布)

学习曲线= % 在测试集上分类正确的百分比（对不同规模的训练集）



提纲

- 相关概念
- 决策树学习
- 人工神经网络
- 深度学习入门

人工神经网络

主要问题：

1）什么是神经网络?

2）如何建立神经网络?

3）如何应用神经网络?

(人工)神经网络（ARTIFICIAL NEURAL NETWORK 简记ANN)：

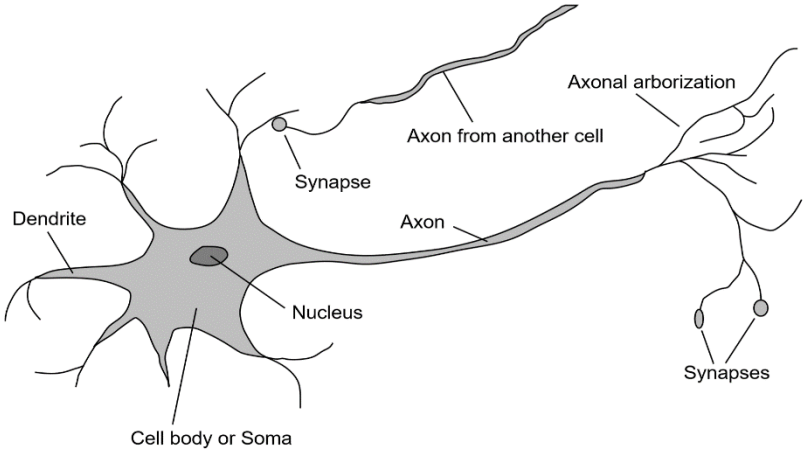
一种模仿大脑神经网络结构和功能而建立的信息处理系统。

是大量的神经元广泛互连形成的一个复杂网络。

表示神经网络的输入与输出变量之间关系的模型，称为神经网络模型。

人工神经网络

➢ 生物神经元

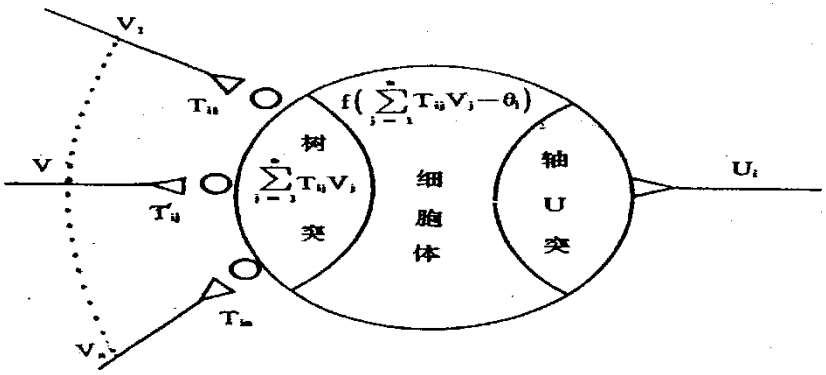


神经元由细胞体、树突和轴突三部分组成。

细胞体，对接收到的信息进行处理；

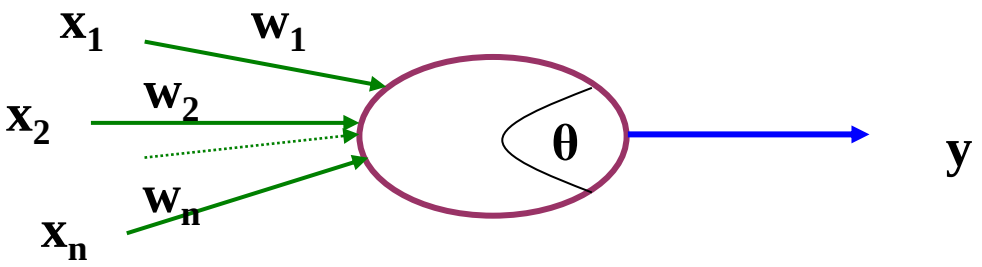
轴突，发出信息；

树突，接收信息



人工神经网络

➢ 人工神经元网络模型



M-P神经元模型

在图中，x_1，x_2，…，x_n表示某一神经元的n个输入；w_i表示第i个输入的连接强度，称为连接权值，正数权值表示兴奋性输入，负数权值表示抑制性输入；θ为神经元兴奋时的阈值，当神经元输入的加权和大于θ时，神经元处于兴奋状态；y为神经元的输出。可以看出，人工神经元是一个具有多输入，单输出的非线性器件。

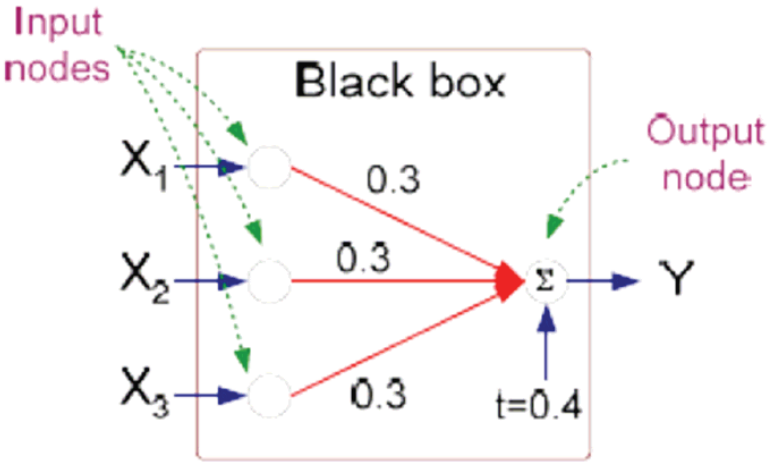
人工神经网络

M-P神经元模型的输入是：　　$\sum w_i * x_i$　　$(i=1,2,\ldots\ldots,n)$

输出是：　　$y=f(\sigma)=f(\sum w_i * x_i - \theta)$

其中f 称之为神经元激活函数、激励函数。

X_1	X_2	X_3	Y
1	0	0	-1
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	-1
0	1	0	-1
0	1	1	1
0	0	0	-1



- $Y=\text{sign}(0.3x_1+0.3x_2+0.3x_3-0.4)$
- 其中：

$$\text{Sign}(x)=\begin{cases} 1 & \text{if } x>0 \\ -1 & \text{if } x<0 \end{cases}$$

  - Sign(x)为激活函数

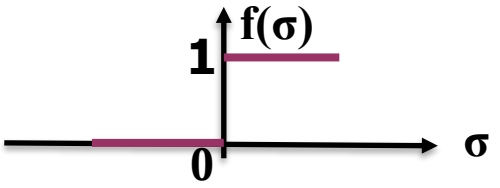
人工神经网络

➢ 常用的激活函数

激活函数f是表示神经元输入与输出之间关系的函数，根据功能函数的不同，可以得到不同的神经元模型。常用的神经元模型有以下几种。

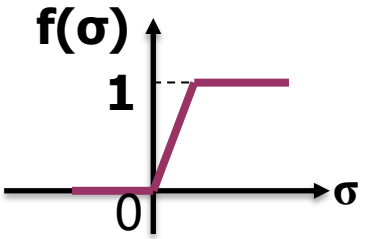
（1）阶跃函数或阈值函数（Threshold

$$\text{Sign}(x)=\begin{cases}1 & \text{if } x>0\\ 0 & \text{if } x<0\end{cases}$$



阶梯型神经元的输入 / 输出特性

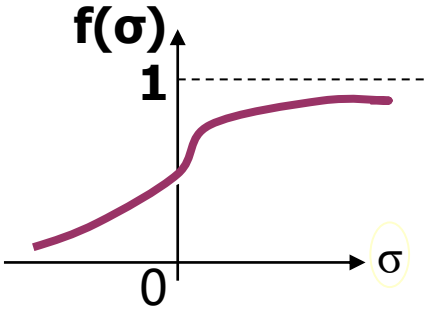
（2）分段线性强饱和型（Linear Saturation）



性饱和型神经元的输入 / 输出特性

（3）S型（Sibmoid）

$$Y=\text{Sign}(x)=1/(1+e^{-x})$$



S型神经元的输入 / 输出特性

人工神经网络

➢ 神经网络结构

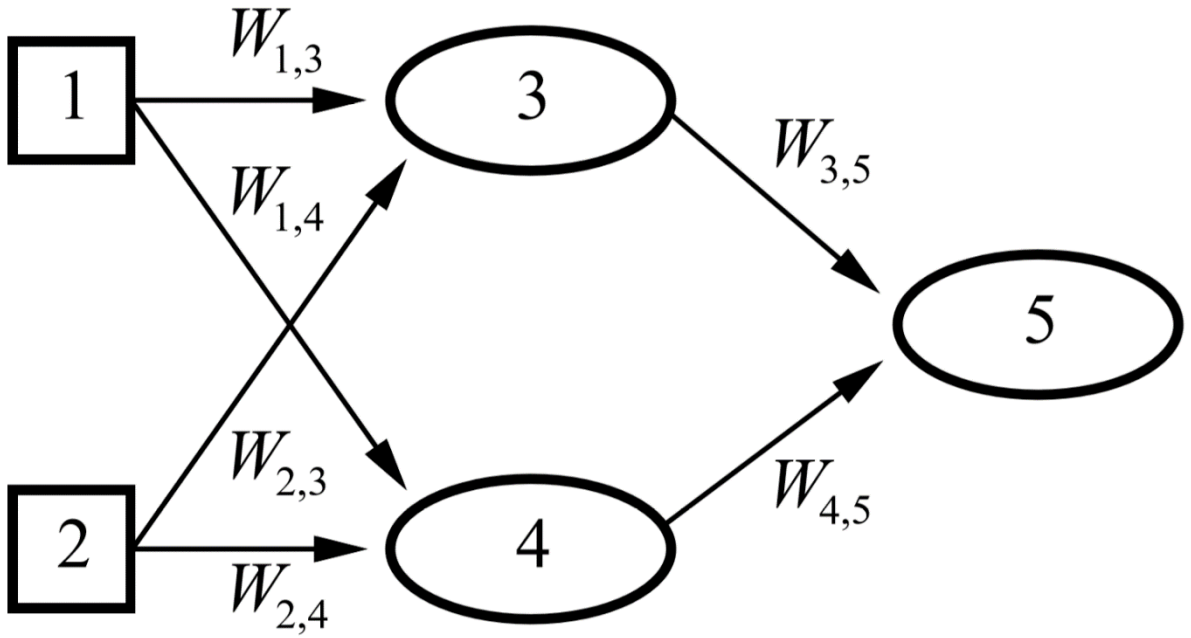
- 前馈网络：排列成层次，每层神经元都仅仅从其直接前一层的神经元接收输入
  - 单层前馈网络（感知器）
  - 多层前馈网络
  - 前馈网络表示输入的一个函数，网络没有内部状态
- 循环网络：输出反馈回输入
  - Hopfield网络：具有对称权值 $(W_{i,j} = W_{j,i})$

    $g(x) = \text{sign}(x),\ a_i = \pm 1$

    联想记忆
  - RNN：具有内部状态，可能会振荡

人工神经网络

一个简单的前馈网络



$$
\begin{aligned}
a_5 &= g(W_{3,5} \cdot a_3 + W_{4,5} \cdot a_4) \\
&= g(W_{3,5} \cdot g(W_{1,3} \cdot a_1 + W_{2,3} \cdot a_2) + W_{4,5} \cdot g(W_{1,4} \cdot a_1 + W_{2,4} \cdot a_2))
\end{aligned}
$$

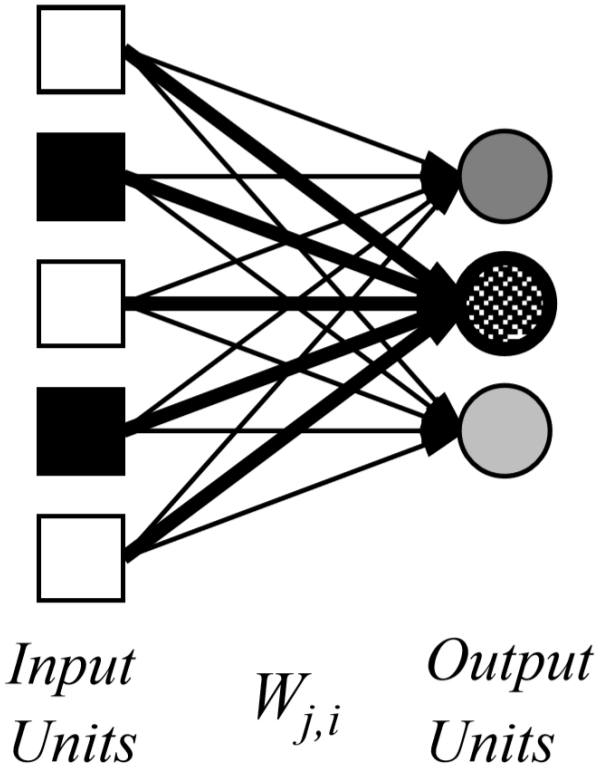
一个前馈网络＝输入的一组非线性函数

通过调整权值，可以改变这个网络所表示的函数

神经网络中的学习：调整参数

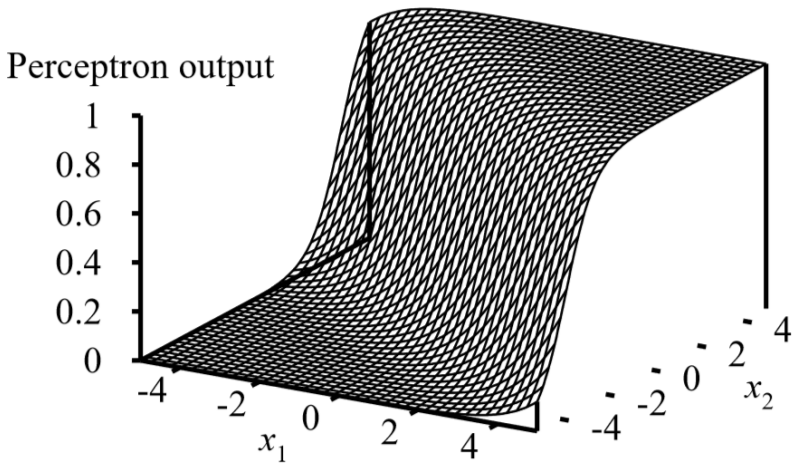
人工神经网络

➢ 单层前馈神经网络（感知器）



(a)

一个感知器网络



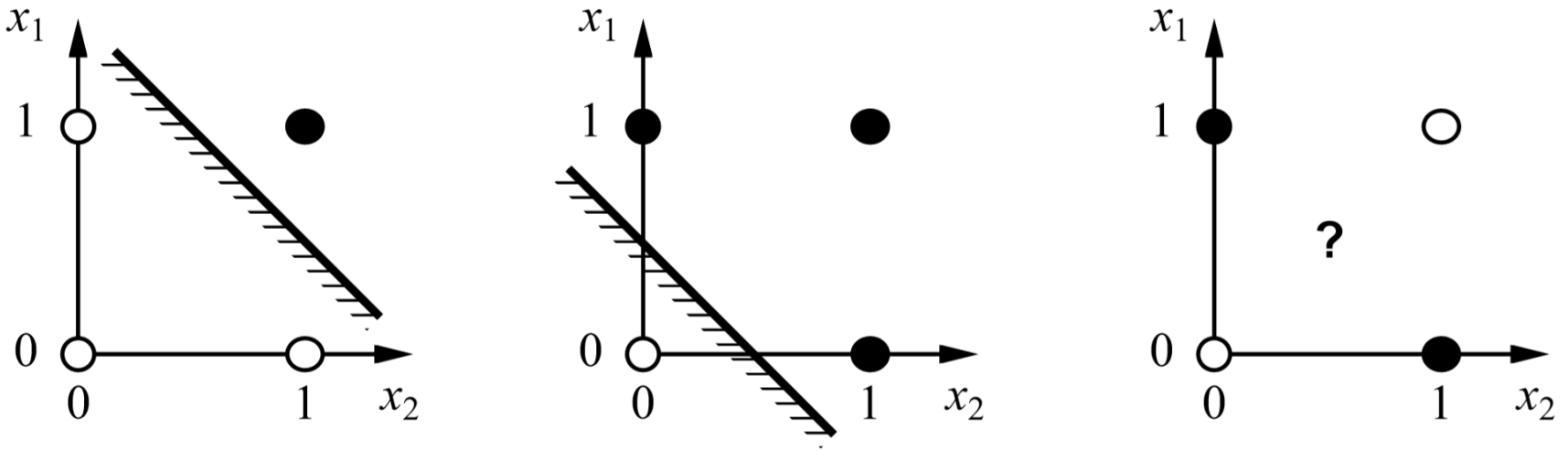
(b)

两输入的S型激活函数感知器的一 个输出结果图

人工神经网络

➢ 感知器的表示能力

- 一个采用阈值激活函数的感知器，可以表示逻辑与、或、非、多数函数等，但不能表示异或（XOR）函数
- 只能表示线性可分的函数



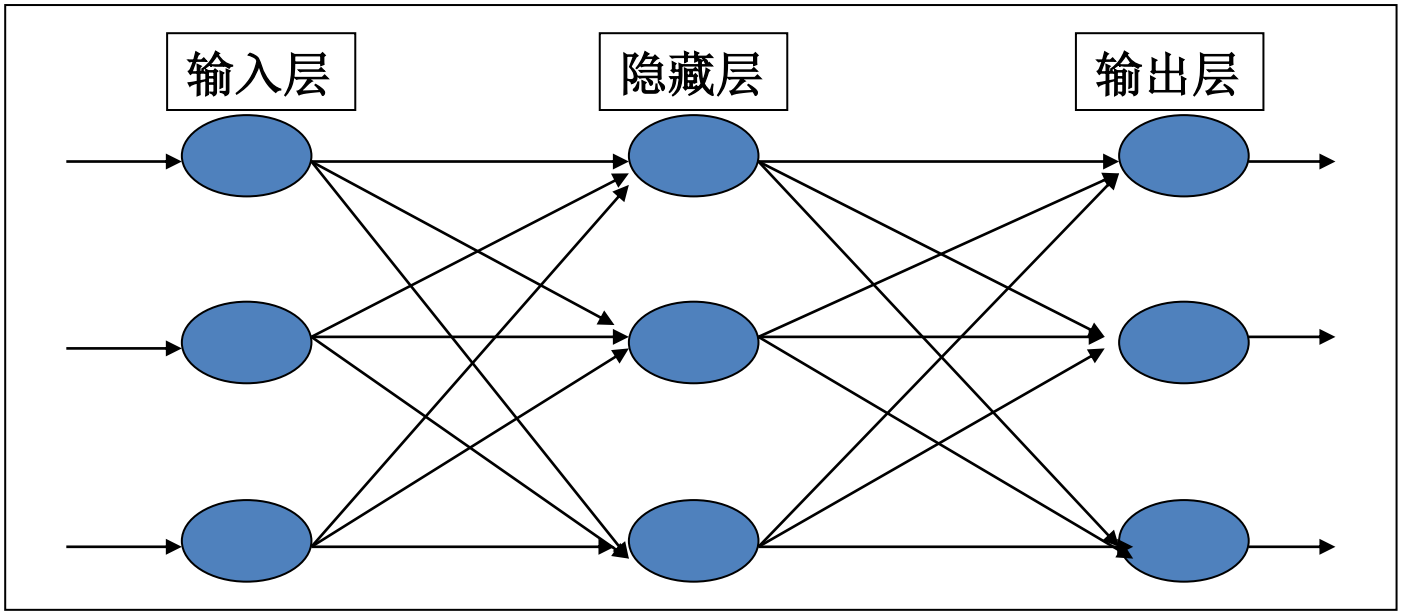
(a) x_1 **and** x_2　　(b) x_1 **or** x_2　　(c) x_1 **xor** x_2

人工神经网络

➢ 多层前馈神经网络

1、最流行的神经网络算法是20世纪80年代提出的后向传播算法。

2、这种算法是在多层前馈神经网络基础上运行的。



前馈神经网络是分层网络模型，具有一个输入层、一个输出层，输入层和输出层之间有一个或多个隐藏层。每个层具有若干个单元，前一层单元与后一层单元之间通过有向加权边相连。

人工神经网络

输入层

节点的数据与训练样本的非类别属性数目对应，通常一个连续属性对应一个输入层单元，一个p值离散属性对应p个输入层单元；

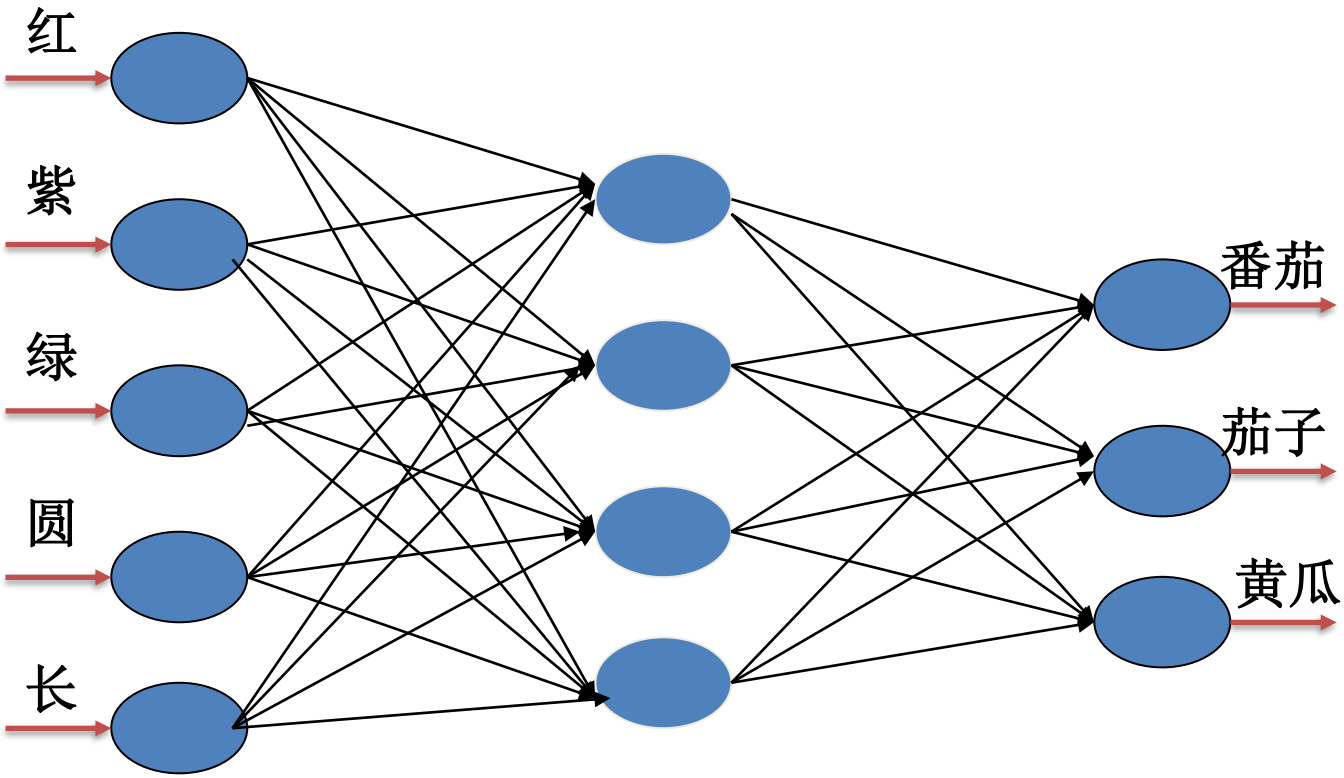
输出层

节点的数据与训练样本的类别数目对应；

隐藏层

层数及隐藏层节点的数目目前尚无理论指导，凭经验选取。但是，至今也没有一个明确的理论来指导我们如何确定每个隐含层的节点数和网络的层数。在实践中通常用试错法。

人工神经网络

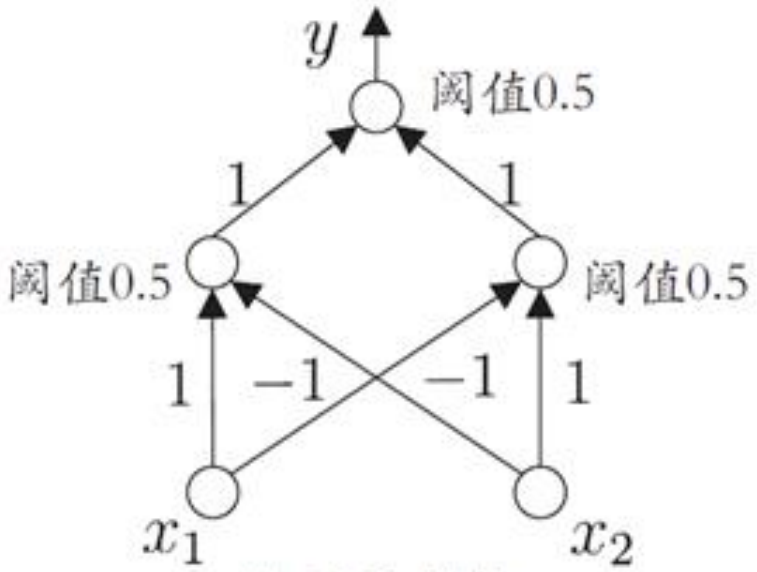


人工神经网络

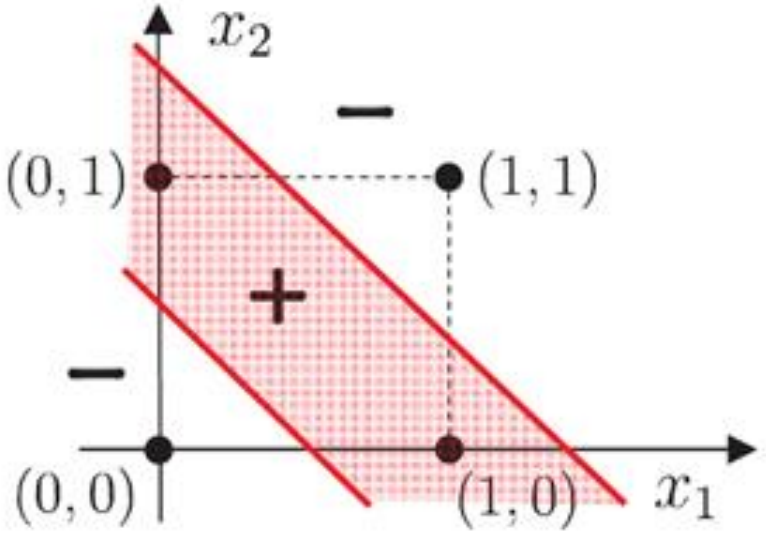
多层感知器表示能力

- 只要一个足够大的隐层，一个隐层网络可以任意精度表示关于输入的任何连续函数
- 两个隐层网络可表示任何函数
- 要点：所需隐单元数会随输入数目成指数级增长

人工神经网络



(a) 网络结构



(b) 分类区域

能解决异或问题的两层感知机

人工神经网络

实例：是否在餐馆等位置问题

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
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

如何设计一个神经网络模型？

人工神经网络

- 确定神经网络结构
  - 10-？-1
  - 例，10-4-1

- 确定神经网络参数
  - 权值、阈值
  - 如何确定？
  - 涉及学习算法

人工神经网络

误差反向传播（Error Back-Propagation, BP)学习

BP神经网络模型是一种具有三层或三层以上的前馈型的、按梯度下降算法使计算输出与实际输出的误差沿反向传播修正各连接权值的神经网络模型。

学习（训练）的最终目的： 利用训练集获得权重的集合，使得网络能够正确地将训练集分类。

人工神经网络

➢ 误差反向传播方法

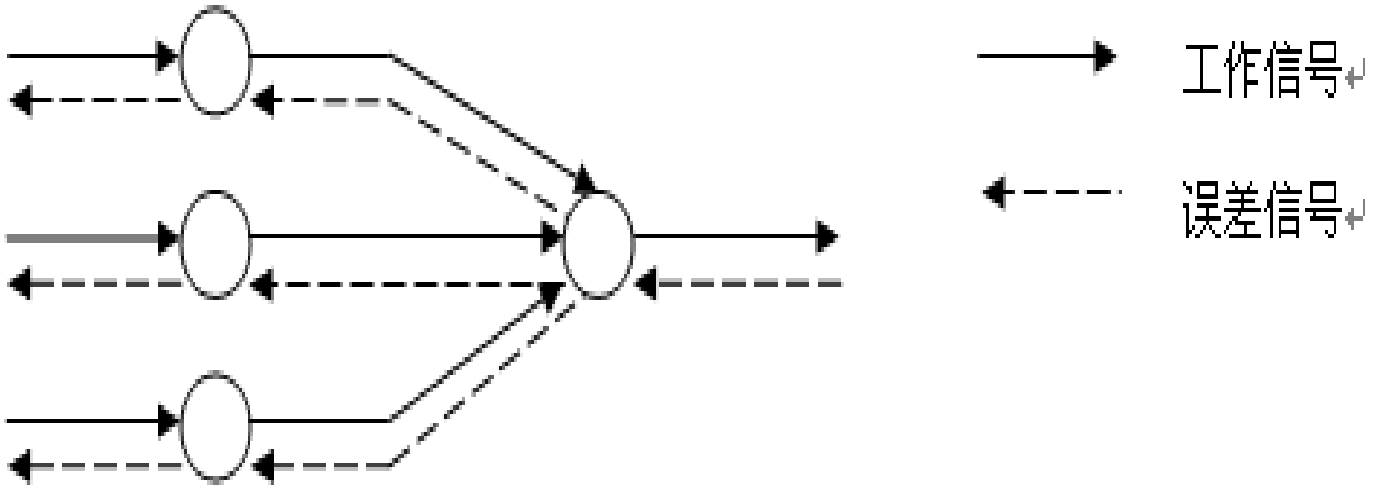
- 通过迭代地处理一组训练样本，将每个样本的网络预测与实际指导的类标号比较，进行学习。对于每个训练样本，修改权值，使得网络预测和实际类之间的均方误差最小。这种修改“后向”进行。即由输出层，经由每个隐藏层，到第一个隐藏层。一般地，权将最终收敛，学习过程停止。

人工神经网络

➢ BP学习原理

工作信号正向传播/前向传播

误差信号反向传播/后向传播



人工神经网络

- 正向传播：当给定网络一组输入模式时，BP网络将依次对这组输入模式中的每个输入模式按如下方式进行学习：把输入模式从输入层传到隐含层单元，经隐含层单元逐层处理后，产生一个输出模式传至输出层，这一过程称为正向传播。
- 反向传播：如果经正向传播在输出层没有得到所期望的输出模式，则转为误差反向传播过程，即把误差信号沿原连接路径返回，并通过修改各层神经元的连接权值，使误差信号为最小。
- 重复正向传播和反向传播过程，直至得到所期望的输出模式为止。

人工神经网络

➢ BP网络的学习算法思想：

1) 初始化网络及学习参数，即将隐含层和输出层各节点的连接权值、神经元阈值赋予[- 1，1]或[- 0.5，0.5]区间的一个小的随机数。
2) 提供训练模式，即从训练模式集合中选出一个训练模式，将其输入模式和期望输出送入网络。
3) 正向传播过程，即对给定的输入模式，从第一隐含层开始，计算网络的输出模式，并把得到的输出模式与期望模式比较，若有误差，则执行第4）步；否则，返回第2）步，提供下一个训练模式；
4) 反向传播过程，即从输出层反向计算到第一隐含层，逐层修正各单元的连接权值。
5) 返回第2）步，对训练模式集中的每一个训练模式重复第2）到第3）步，直到训满足结束条件为止。

人工神经网络

- 更新方法：
  - 实例更新：每处理一个样本就更新权和偏置。
  - 周期更新：权和偏置的增量也可以累积到变量中，使得可以在处理完训练集中的所有样本之后再更新权和偏置。
- 终止条件：

  前一周期所有的w_{ij}的变化量都很小，小于某个指定的阈值

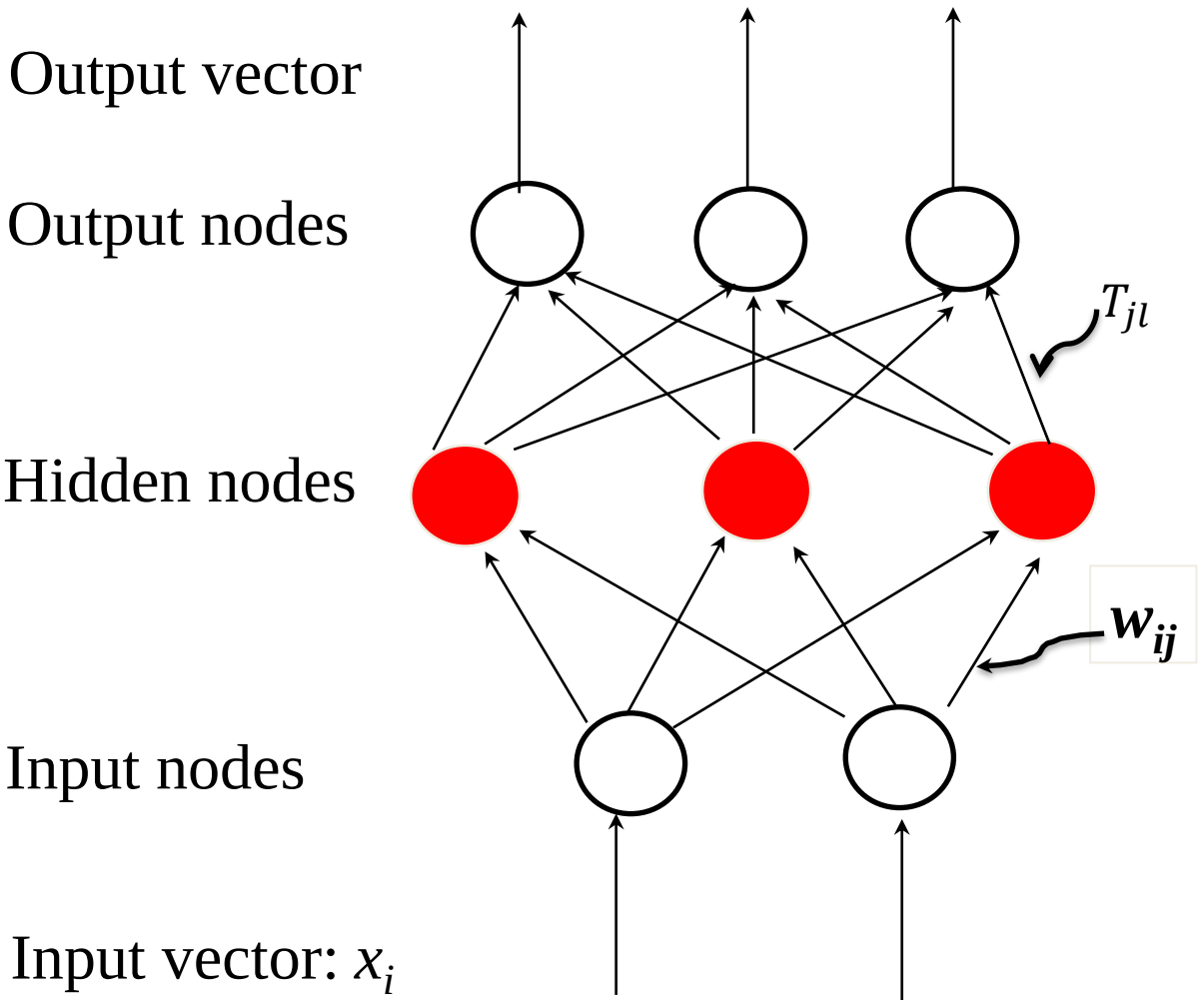
  前一周期未正确分类的样本百分比小于某个阈值

  超过预先指定的周期数

  实践中，权收敛可能需要数十万个周期。

人工神经网络

- BP算法


Output vector
Output nodes
Hidden nodes
Input nodes
Input vector: x_i
T_{jl}
w_{ij}

人工神经网络

➢ 相关概念

设(Xp, Tp) 表示输入样本，p {1, 2,..., N}，N为输入样本的个数。W表示网络权向量。

误差函数：E (W)=g (f (W, Xp, Tp))，E称为误差（测度）函数。用误差函数来判别网络的实际输出向量Yp与教师信号向量Tp的误差。常采用二乘误差函数加以判别（m为输出向量的维数）：

$$E=\frac{1}{2}\sum_{p=1}^{N}E_p=\frac{1}{2}\sum_{p=1}^{N}\left(\mathbf{T}_p-\mathbf{Y}_p\right)^2=\frac{1}{2}\sum_{p=1}^{N}\sum_{i=1}^{m}\left(T_{ip}-Y_{ip}\right)^2$$

人工神经网络

➢ BP算法权值的修正量

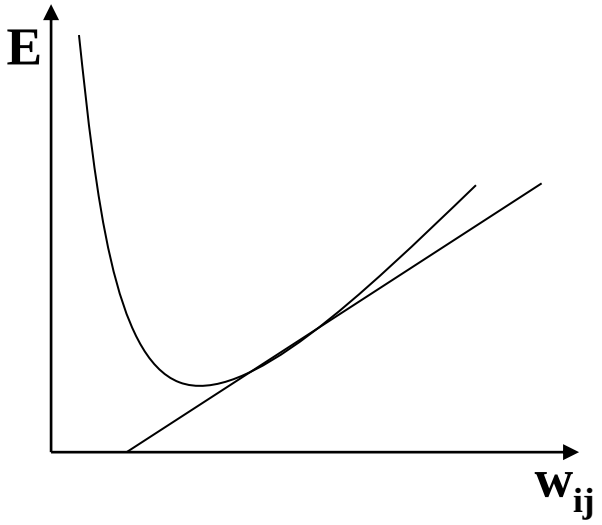
- BP算法基于梯度下降算法。在梯度下降算法中，权值的修正量正比于误差函数E(W)对W的负梯度，即：
- **W (t+1)= W (t) +ΔW (t)**

$$\Delta \mathbf{W}(t) = -\eta \frac{\partial E(\mathbf{W})}{\partial(\mathbf{W})}$$

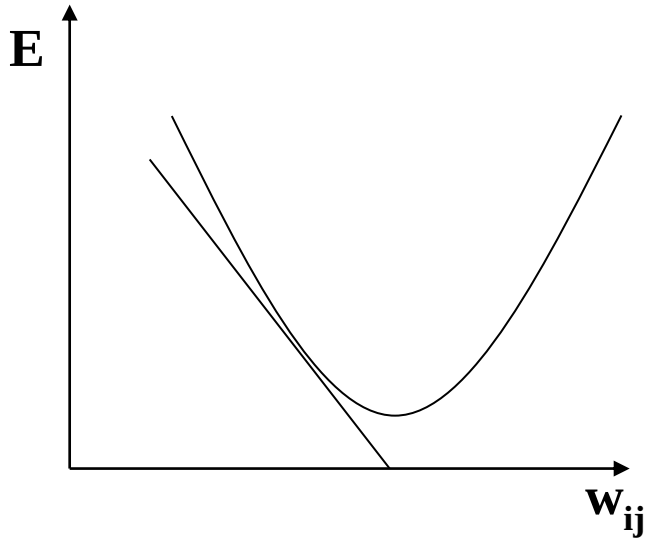
η 是学习率，**0<η<1**

人工神经网络

取 $\Delta w_{ij} \propto -\frac{\partial E}{\partial w_{ij}}$



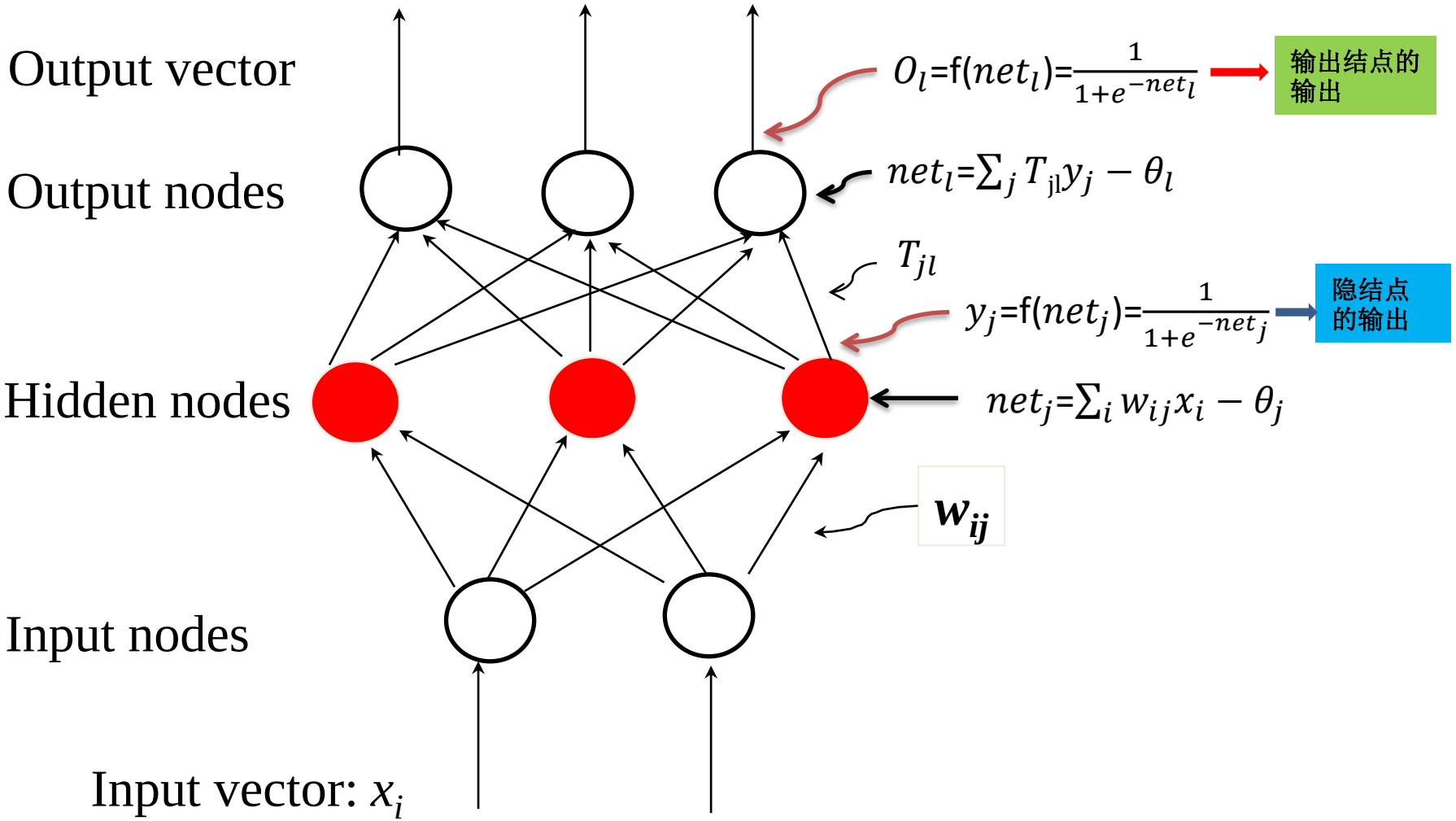
$\frac{\partial E}{\partial w_{ij}} > 0$ ， 此时$\Delta w_{ij}<0$



$\frac{\partial E}{\partial w_{ij}} < 0$，此时$\Delta w_{ij}>0$

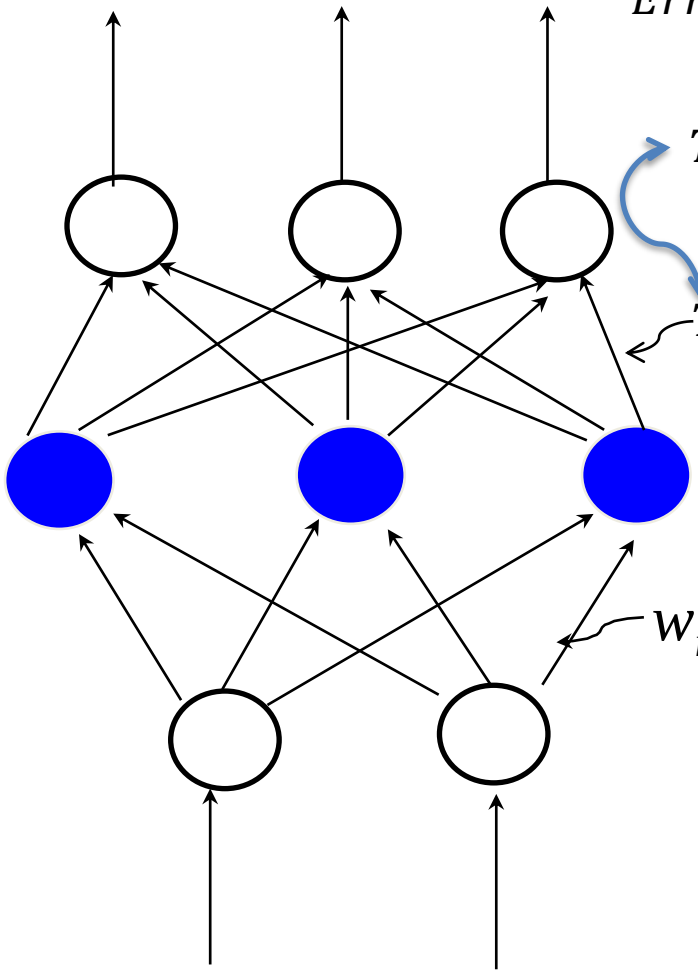
人工神经网络

- 信号正向传播计算各层输出



人工神经网络

- 反向传播进行权值修正



$f'(net_l)$

$Err_l=O_l(1-O_l)(T_l-O_l)$ 输出层误差

$T_{jl}(k+1)=T_{jl}(k)+\delta Err_l y_j$ 权值修正

$\theta_l(k+1)=\theta_l(k)-\delta Err_l$ 阈值修正

$f'(net_j)$

$Err_j=y_j(1-y_j)\sum_l Err_l T_{jl}$

$W_{ij}(k+1)=W_{ij}(k)+\delta' Err_j x_i$

$\theta_j(k+1)=\theta_j(k)-\delta' Err_j$

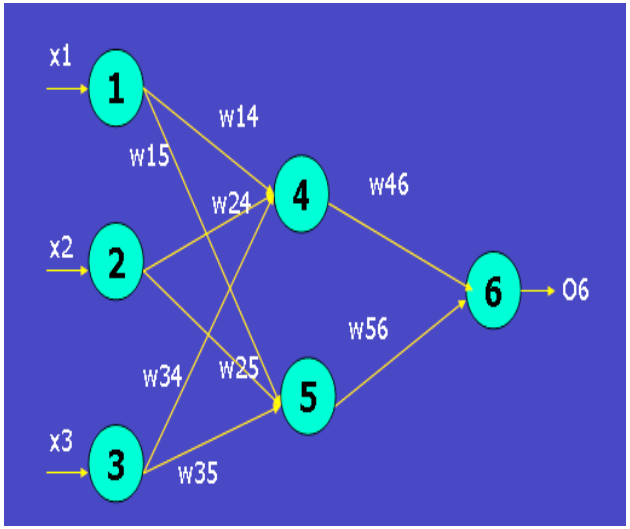
人工神经网络

➢ 实例1

$Err_l=O_l(1-O_l)(T_l-O_l)$

$Err_j=y_j(1-y_j)\sum_l Err_l\, T_{jl}$

X1	X2	X3	W14	W15	W24	W25	W34	W35	W46	W56	θ4	θ5	θ6
1	0	1	0.2	-0.3	0.4	0.1	-0.5	0.2	-0.3	-0.2	0.4	-0.2	-0.1



单元j	输入 net	输出o
4	0.2*1+0.4*0+(-0.5)*1-0.4=-0.7	1/(1+e-(-0.7))=0.332
5	(-0.3)*1+0.1*0+(0.2)*1-(-0.2)=0.1	1/(1+e(-0.1))=0.525
6	(-0.3)*0.332+(-0.2)*0.525-(-0.1)=-0.105	1/(1+e-(-0.105))=0.474

单元j	误差
6	0.474*(1-0.474)*(1-0.474)=0.1311
5	0.525*(1-0.525)*(0.1311*(-0.2))=-0.0065
4	0.332*(1-0.332)*(0.1311*(-0.3))=-0.0087

训练样本x={1, 0, 1}

类标号为1

$T_{jl}(k+1)=T_{jl}(k)+\delta Err_l y_j$

$W_{ij}(k+1)=W_{ij}(k)+\delta' Err_j x_i$

$\theta_l(k+1)=\theta_l(k)-\delta Err_l$

$\theta_i(k+1)=\theta_i(k)-\delta' Err_j$

w46	-0.3+0.9*0.1311*0.332=-0.216
w56	-0.2+0.9*0.1311*0.525=-0.138
w14	0.2+0.9*(-0.0087)*1=0.192
w15	-0.3+0.9*(-0.0065)*1=-0.306
w24	0.4+0.9*(-0.0087)*0=0.1
w25	0.1+0.9*(-0.0065)*0=0.1
w34	-0.5+0.9*(-0.0087)*1=-0.508
w35	0.2+0.9*(-0.0065)*1=-0.194
Θ6	-0.1-0.9*0.1311=-0.218
Θ5	-0.2-0.9*(-0.0065)=-0.194
Θ4	0.4-0.9*(-0.0087)=0.408

本例只有一个训练样本，只示例了网络学习过程中的一次迭代过程。

人工神经网络

神经网络的学习过程虽然漫长，但分类速度却很快。学习结束后，网络得到一组固定的权值及偏置值。未知类别的样本到来后，将其属性送入输入层各节点，从输入层到输出层正向传播，计算输出层各单元的值，输出值最大的单元所代表的类就是该样本所属的类别。

人工神经网络

➢ 实例2

通过实例归纳算法的基本流程

假设某汽车销售公司希望对购买不同档次汽车的客户进行分类，分类的依据是客户的年龄、收入和学历。公司希望通过这三个分类标准，推断出客户会购买哪种档次的汽车。

人工神经网络

年龄划分成四档：　30岁以下，31-40岁，
41-50岁，50岁以上。

收入划分成四档：　1500元以下，1501-2500元，
2501-3500元，3500元以上。

学历也划分成四档：大学及大学以上，大专，
高中及中专，初中及初中以下。

车的档次分为两档：高档和低档。

人工神经网络

- 实现步骤：

1、输入数据的预处理

输入到神经网络中的数据必须是在[0，1]闭区间内的数字，所以需要用[0，1]上的数字形式来表示分类数据。

30岁以下用[0，0]表示，31-40岁用[0，1]表示，41-50岁用[1，0]表示，50岁以上用[1，1]表示。

1500元以下用[0，0]表示，1501-2500元用[0，1]表示，2501-3500元用[1，0]表示，3500元以上用[1，1]表示。

大学及大学以上用[0，0]表示，大专用[0，1]表示，高中及中专用[1，0]表示，初中及初中以下用[1，1]表示。

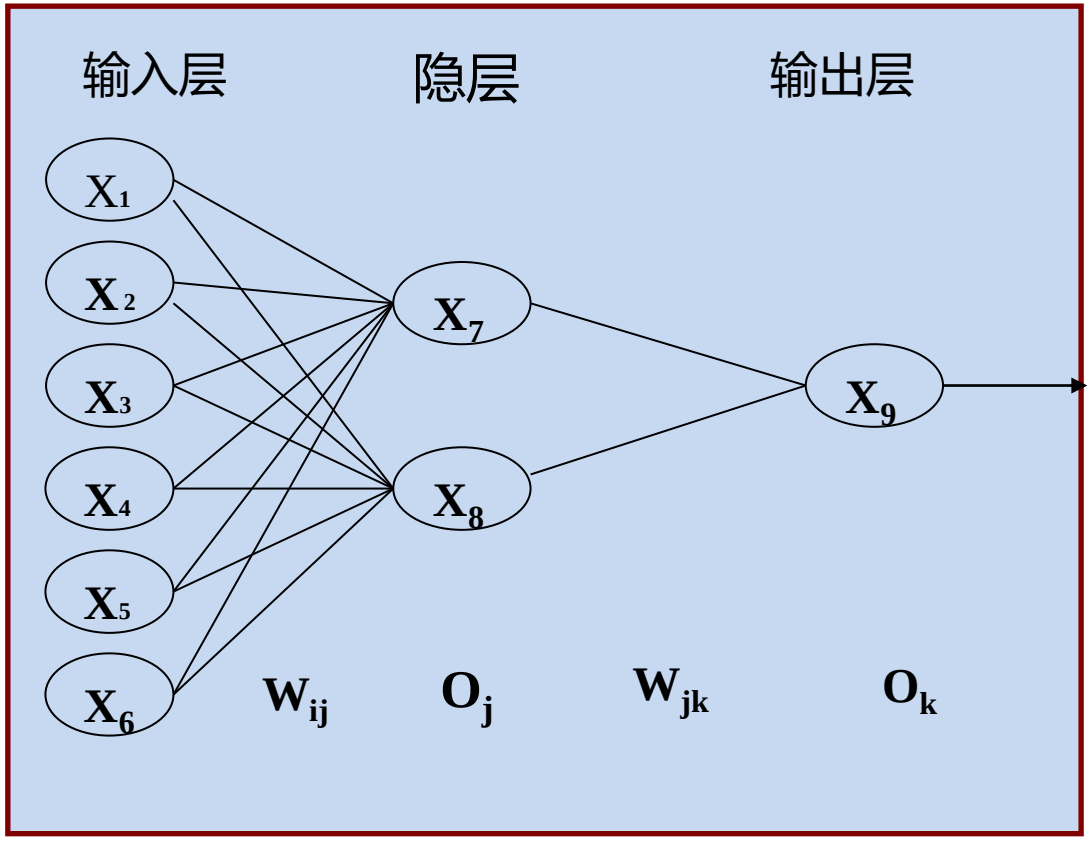
人工神经网络

2、确定网络拓扑图

1）输入数据有三个：年龄、收入和学历。

2）设网络中只有一个隐层，而且隐层中包含两个结点

3）输出层中有一个输出结点。若输出值为1，则表示购买高档车，若输出值为0，则表示购买低档车。



人工神经网络

3、BP算法的具体流程

（1）权重阈值初始化

一般初始化权重的范围在一个较小的区间内，如[-1.0，1.0]或[-0.5，0.5]。

结点的初始权重

W	W_{17}	W_{27}	W_{37}	W_{47}	W_{57}	W_{67}
权值	0.25	0.1	0.2	0.3	0.4	0.5
W	W_{18}	W_{28}	W_{38}	W_{48}	W_{58}	W_{68}
权值	-0.2	-0.1	-0.15	0.2	0.3	0.1
W	W_{79}	W_{89}				
权值	0.15	0.05				

结点的初始阈值

θ	θ_7	θ_8	θ_9
阈值	0.1	0.5	-0.2

人工神经网络

(2) 正向传播

计算各个层次的每个结点的输入及输出情况。

输入层：对于输入层中的结点i来说，其输出值等于输入值，即 $O_i=I_i$。

设有一个样本是：年龄在41-50岁之间，收入在3500元以上，学历是大学水平，已知的输出是它会购买高档汽车。此样本的输入为：

I	I_1	I_2	I_3	I_4	I_5	I_6
输入	1	0	1	1	0	0

期望的输出是：1。

人工神经网络

隐层、输出层：每个结点的输入都与上一层的输出有关，是上一层的输出与相应权值的复合。

$$I_j = \sum_i w_{ij} O_i - \theta_j$$

其中，W_{ij}表示从上一层的结点i到当前层中的结点j的连接权重。O_i表示i结点的输出，θ_j表示结点j的阈值。

人工神经网络

$$I_j = \sum_i w_{ij} O_i - \theta_j$$

W	W_{17}	W_{27}	W_{37}	W_{47}	W_{57}	W_{67}
权值	0.25	0.1	0.2	0.3	0.4	0.5

O	0_1	0_2	0_3	0_4	0_5	0_6
输出	1	0	1	1	0	0

根据公式计算得到(隐层的输入)：

I_7=0.25*1+0.1*0+0.2*1+0.3*1+0.4*0+0.5*0-0.1=0.65

I_8=-0.2*1-0.1*0-0.15*1+0.2*1+0.3*0+0.1*0-0.5=-0.65

人工神经网络

对于结点 j，若给定其输入为 I_j，则其输出为

$$O_j = f(I_j) = \frac{1}{1+e^{-I_j}}$$

根据公式计算得到：

$$O_7 = \frac{1}{1+e^{-0.65}} = 0.657 \qquad O_8 = \frac{1}{1+e^{0.65}} = 0.343$$

$I_9 = 0.15*0.657 + 0.05*0.343 - (-0.2) = 0.3157$

$$O_9 = \frac{1}{1+e^{-0.3157}} = 0.366$$

人工神经网络

（3）反向传播

反向传播过程实际上是一个误差反向传播修正的过程。在这个过程中，根据输出层的实际输出和期望输出之间的误差，修改权重及相应的阈值以减少网络预测中存在的误差。

人工神经网络

对于输出层误差的计算公式：

$Err_j=f'(I_j)\ (T_j-O_j)$

其中，I_j 表示 j 结点的输入，T_j 表示目标输出，O_j 表示j结点的实际输出，$f'(I_j)$是S型函数的一阶导数。

S型函数在I_j的导数可以方便地计算为$O_j\ (1-O_j)$。

$Err_j= O_j\ (1-O_j)\ (T_j-O_j)$

对于隐层误差的计算公式为：

$$Err_j = f'(I_j)\sum_k Err_k W_{jk}$$

$$Err_j = O_j\ (1-O_j)\sum_k Err_k W_{jk}$$

人工神经网络

权值的修正公式为：

$$\Delta w_{ij}=\eta Err_j O_i$$

$$w_{ij}=w_{ij}+\Delta w_{ij}$$

阈值的修正公式为：

$$\Delta \theta_j=-\eta Err_j$$

$$\theta_j=\theta_j+\Delta \theta_j$$

人工神经网络

$Err_j = O_j (1 - O_j)(T_j - O_j)$ $\quad Err_j = O_j (1 - O_j) \sum_k Err_k W_{jk}$

根据公式计算得到：

$Err_9 = O_9 (1 - O_9)(T_9 - O_9)$

$= 0.366 * (1 - 0.366) * (1 - 0.366) = 0.147$

$Err_7 = O_7 (1 - O_7) * Err_9 * W_{79}$

$= 0.657 * (1 - 0.657) * 0.147 * 0.15 = 0.00497$

$Err_8 = O_8 (1 - O_8) * Err_9 * W_{89}$

$= 0.343 * (1 - 0.343) * 0.147 * 0.05 = 0.00166$

人工神经网络

$$\Delta w_{ij}=\eta Err_j O_i$$

$$w_{ij}=w_{ij}+\Delta w_{ij}$$

设$\eta=0.5$，则对权值的修正为：

$$\Delta W_{17}=\eta Err_7 O_1=0.5*0.00497*1=0.0025$$

$$\Delta W_{27}=0.5*0.00497*0=0$$

$$\Delta W_{37}=0.5*0.00497*1=0.0025$$

$$\Delta W_{47}=0.5*0.00497*1=0.0025$$

$$\Delta W_{57}=0.5*0.00497*0=0$$

$$\Delta W_{67}=0.5*0.00497*0=0$$

人工神经网络

$\Delta w_{ij}=\eta Err_j O_i$

$w_{ij}=w_{ij}+\Delta w_{ij}$

$W_{17}=W_{17}+\Delta W_{17}=0.25+0.0025=0.2525$

$W_{27}= 0.1+0=0.1$

$W_{37}= 0.2+0.0025=0.2025$

$W_{47}= 0.3+0.0025=0.3025$

$W_{57}= 0.4+0=0.4$

$W_{67}= 0.5+0=0.5$

人工神经网络

$\Delta W_{18}=\eta Err_8 O_1=0.5*0.00166*1=0.0008$

$\Delta W_{28}=0.5*0.00166*0=0$

$\Delta W_{38}=0.5*0.00166*1=0.0008$

$\Delta W_{48}=0.5*0.00166*1=0.0008$

$\Delta W_{58}=0.5*0.00166*0=0$

$\Delta W_{68}=0.5*0.00166*0=0$

人工神经网络

$W_{18}=W_{18}+\Delta W_{18}=-0.2+0.0008=-0.1992$

$W_{28}= -0.1+0=-0.1$

$W_{38}= -0.15+0.0008=-0.1492$

$W_{48}= 0.2+0.0008=0.2008$

$W_{58}= 0.3+0=0.3$

$W_{68}= 0.1+0=0.1$

人工神经网络

$\Delta W_{79}=\eta Err_9 O_7=0.5*0.147*0.657=0.0483$

$\Delta W_{89}=\eta Err_9 O_8=0.5*0.147*0.55=0.0404$

$W_{79}=W_{79}+\Delta W_{79}=0.15+0.0483=0.1983$

$W_{89}=W_{89}+\Delta W_{89}=0.05+0.0404=0.0904$

人工神经网络

对阈值的修正为：

$$\Delta\theta_7=-\eta Err_7=-0.5*0.00497=-0.0025$$

$$\Delta\theta_8=-\eta Err_8=-0.5*0.00166=-0.0008$$

$$\Delta\theta_9=-\eta Err_9=-0.5*0.147=-0.0735$$

$$\theta_7=\theta_7+\Delta\theta_7=0.1-0.0025=0.0975$$

$$\theta_8=\theta_8+\Delta\theta_8=0.5-0.0008=0.492$$

$$\theta_9=\theta_9+\Delta\theta_9=-0.2-0.0735=-0.2735$$

人工神经网络

(4) BP算法的终止条件

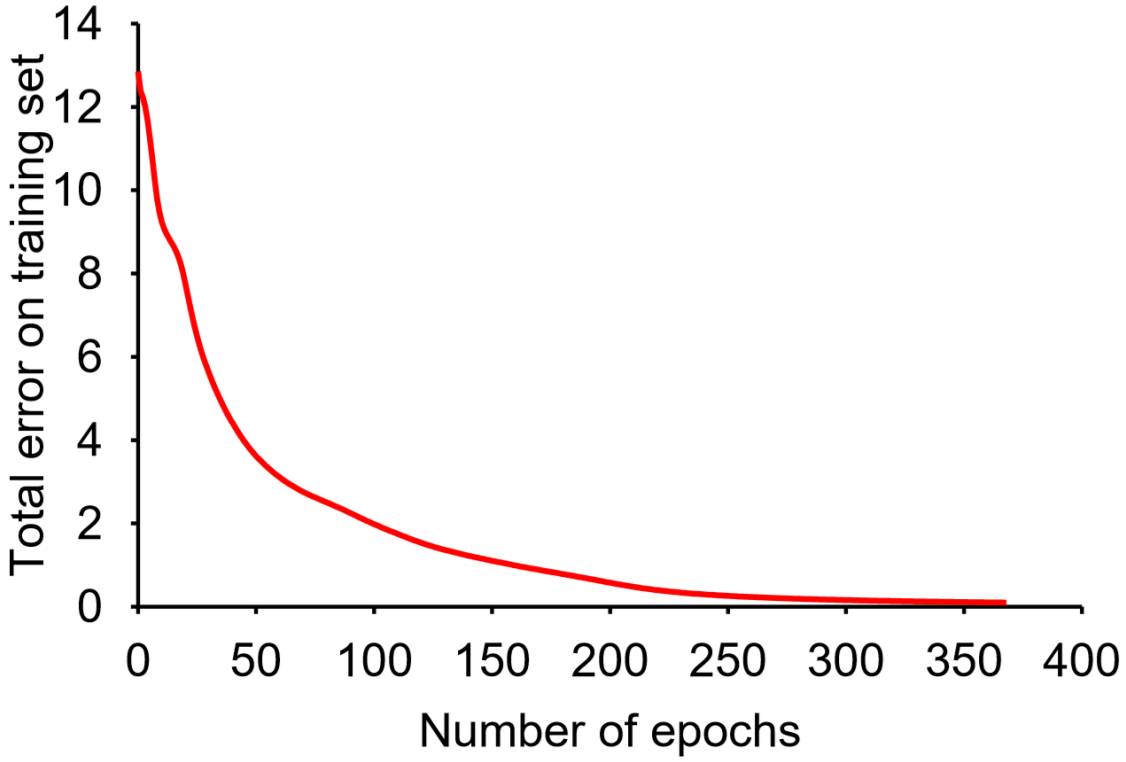
重复次数超过指定的周期数。

对于此题，即输入一百个样本，重复上述计算，对权值和阈值进行训练，当一百个样本输入完毕后，网络中的权值和阈值即为所求。

人工神经网络

单隐层网络在餐馆问题上的应用

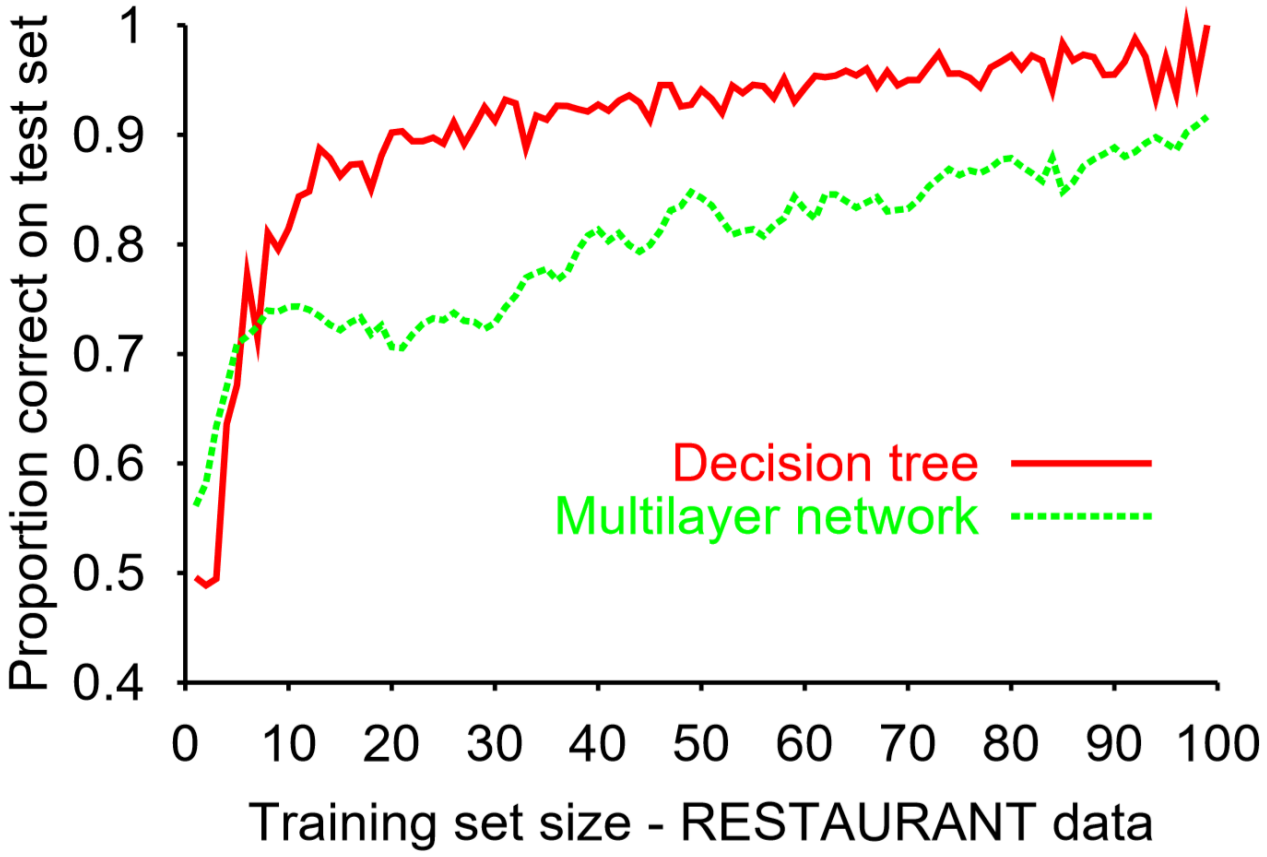
- 神经网络结构：10-4-1
- 数据集：100个样本
- BP算法训练，收敛到了对训练数据的一个很好的拟合



人工神经网络

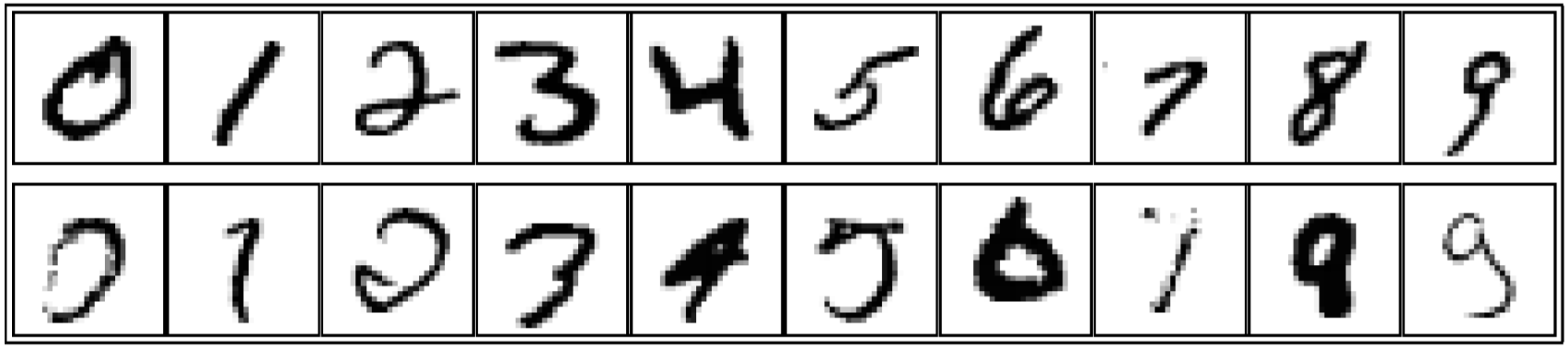
单隐层网络在餐馆问题上的应用（续）

学习曲线比较：



人工神经网络

单隐层网络在手写体数字识别中的应用



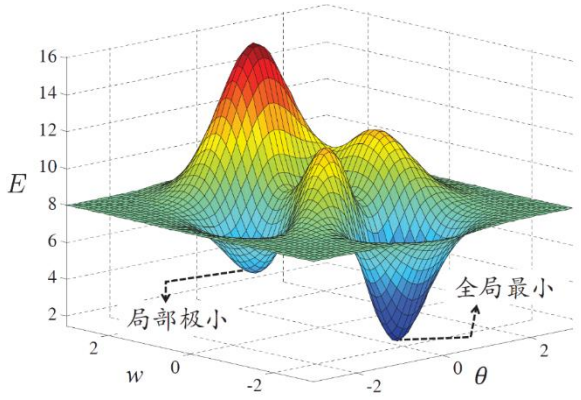
MNIST数据集：60000个训练样本、10000个测试样本
数字图像：20×20
多层感知器：400–300–10
测试集上的误差：1.6%

人工神经网络

多层前馈网络总结

- 优点及适合问题
  - 表达能力强
  - 适合复杂的模式识别任务
  - 很多成功应用

- 局限性
  - 训练过拟合：表现为：训练误差持续降低，但测试误差却可能上升
  - 收敛速度慢
  - 局部极小值
  - 解释能力差
  - 隐藏层神经元个数设置无理论上指导
  - 。。。



人工神经网络

多层前馈网络总结

- 缓解过拟合、加快收敛策略
  - 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部分，例如连接权值与阈值的平方和
  - Dropout
  - 加动量项
  - 优化策略：自适应学习率
  - 新的激活函数，如ReLU
  - 。。。

人工神经网络

多层前馈网络总结

- 跳出“局部最小”策略
  - 多组不同的初始参数优化神经网络，选取误差最小的解作为最终参数
  - 模拟退火：每一步都以一定的概率接受比当前解更差的结果，从而有助于跳出局部极小
  - 随机梯度下降： 与标准梯度下降法精确计算梯度不同，随机梯度下降法在计算梯度时加入了随机因素
  - 遗传算法：遗传算法也常用来训练神经网络以更好地逼近全局极小
  - 。。。