

# 人工智能技术

## Artificial Intelligence

——人工智能:经典智能+智能计算+机器学习

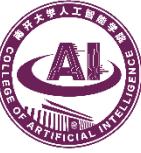
AI: Classical Intelligence + Computing Intelligence + Machine Learning

王鸿鹏

南开大学人工智能学院

hpwang@nankai.edu.cn





## Section IV

---

# 第四部分 机器学习

## Machine Learning

——第十一章：机器学习  
Chapter 11: Machine Learning



# 机器学习

## Machine Learning

- 顾名思义，机器学习是研究如何使用机器来模拟人类活动的一门学科
- 稍微严格一点的提法是：机器学习是一门研究机器获取新知识和新技能，并识别现有知识的学问。

# 定义

## 学习

- 西蒙(Simon): 学习就是系统在不断重复的工作中对本身能力的增强或者改进，使得系统在下一次执行同样任务或类似任务时，会比现在做得更好或效率更高。

Learning denotes **changes** in the system that are adaptive in the sense that they enable the system to do the task or tasks drawn from the same population more efficiently and more effectively the next time.

by Herbert Simon



赫伯特·西蒙  
(Herbert Simon)



# 定义

## 学习

- 维纳(Wiener, 1965): 一个具有生存能力的动物在它的一生中能够被其经受的环境所改造。一个能够繁殖后代的动物至少能够生产出与自身相似的动物(后代), 即使这种相似可能随着时间变化。如果这种变化是自我可遗传的, 那么就存在一种能受自然选择影像的物质。如果该变化是以行为形式出现的, 并假定这种行为是无害的, 那么这种变化就会世代相传下去。这种从一代至下一代的变化形式称为**种族学习**(racial learning)或**系统发育学习**(system growth learning), 而发生在特定个体上的这种行为变化或行为学习, 则称为**个体发育学习**(individual growth learning)



# 定义

## 学习

- 香农(C. Shannon, 1953): 假设 ①一个有机体或一部机器处在某类环境中，或者同该环境有联系； ②对该环境存在一种“成功的”度量或“自适应”度量； ③这种度量在时间上是比较局部的，也就是说，人们能够用一个比有机体生命期短的时间来测试这种成功的度量。对于所考虑的环境，如果这种全局的成功度量能够随时间而改善，那么我们就说，对于所选择的成功度量，该有机体或机器正为适应这类环境而学习。



# 定义

## 学习

- 米切尔(Mitchell): 对于某类任务 $T$  和性能度量 $P$ ，如果一个计算机程序在 $T$  上以 $P$  衡量的性能随着经验 $E$  而自我完善，那么就称这个计算机程序从经验 $E$  中学习。

Machine Learning  $\langle T, P, E \rangle$

Any computer program that improves its  
performance  $P$  at some task  $T$  through experience  $E$ .

by Tom M. Mitchell



汤姆·米切尔  
(Tom M. Mitchell)



# Examples of the learning tasks:

---

- Playing checkers

- T: Playing checkers
  - P: Percentage of games won against an arbitrary opponent
  - E: Playing practice games against itself

- Recognizing

- T: Recognizing hand-written words
  - P: Percentage of words correctly classified
  - E: Database of human-labeled images of handwritten words

- Automatically driving

- T: Driving on four-lane highways using vision sensors
  - P: Average distance traveled before a human-judged error
  - E: A sequence of images and steering commands recorded while observing a human driver.



# 定义

---

## 学习系统

- 学习系统(Learning system)是一个能够学习有关过程的未知信息，并用所学信息作为进一步决策或控制的经验，从而逐步改善过程性能的系统。
- 如果一个系统能够学习某一过程或环境的未知特征固有信息，并用所得经验进行估计、分类、决策和控制，使系统的品质得到改善，那么称该系统为学习系统。
- 学习系统是一个能在其运行过程中逐步获得过程及环境的非预知信息，积累经验，并在一定的评价标准下进行估值、分类、决策和不断改善系统品质的智能系统。



# 定义

## 机器学习

- 顾名思义，机器学习是研究如何使用机器来模拟人类活动的一门学科。
- 稍微严格一点的提法是：机器学习是一门研究机器获取新知识和新技能，并识别现有知识的学问。
- 综合上述两定义，可给出如下定义：

机器学习是研究机器模拟人类的学习活动，  
获取知识和技能的理论和方法，以改善系统性能  
的学科。



# 机器学习的发展史

---

## ◆ 机器学习的发展分为4个时期

### ■ 热烈时期（50年代中叶到60年代中叶）

- 神经元模型研究

- 罗森勃拉特1957年提出的感知器模型

### ■ 冷静时期（60年代中叶至70年代中叶）

- 符号概念获取

- 模拟人类的概念学习过程

### ■ 复兴时期（70年代中叶至80年代中叶）

- 知识强化学习

- 与各种实际应用相结合，尤其是专家系统在知识获取方面

### ■ 最新阶段（1986年至今）

- 连接学习和混合型学习

- 符号学习和连接学习结合



# 机器学习的发展史

---

## ● 机器学习进入新阶段的表现

- 机器学习已成为新的边缘学科并在高校形成课程。
- 综合各种学习方法。
- 机器学习与人工智能问题的统一性观点正在形成。
- 各种学习方法的应用范围不断扩大。
- 数据挖掘和知识发现的研究已形成热潮。
- 与机器学习有关的学术活动空前活跃。



# 机器学习的主要策略

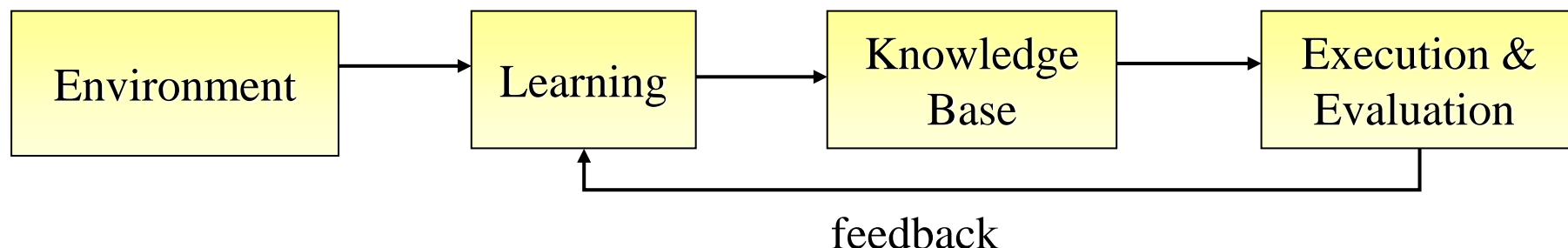
- 学习过程与推理过程是紧密相连的，按照学习中使用推理的多少，机器学习所采用的策略大体上可分为4种——**机械学习、示教学习、类比学习和示例学习**。学习中所用的推理越多，系统的能力越强。
  - 机械学习就是记忆，是最简单的学习策略。这种学习策略不需要任何推理过程。
  - 比机械学习更复杂一点的学习是示教学习策略。系统在接受外部知识时需要一点推理，翻译和转化工作。
  - 类比学习系统只能得到完成类似任务的有关内容，因此，它比上述两种学习策略需要更多的推理。
  - 采用示例学习策略的计算机系统，事先完全没有完成任务的任何规律性的信息，因此需要推理是最多的。



# 机器学习系统的基本结构

以西蒙的学习定义为出发点，建立起简单的学习模型：

环境向系统的学习部分提供某些信息，学习部分利用这些信息修改知识库，以增进系统执行部分完成任务的效能，执行部分根据知识库完成任务，同时把获得的信息反馈给学习部分。在具体的应用中，环境，知识库和执行部分决定了具体的工作内容，学习部分所需要解决的问题完全由上述3部分确定。





# 影响学习系统设计的要素

---

## ◆ 环境向系统提供的信息

- 或者更具体地说是信息的质量
- 最重要因素

## ◆ 知识库

- 是影响学习系统设计的第二个因素
- 知识的表示有特征向量、一阶逻辑语句、产生式规则、语义网络和框架等多种形式



# 机器学习概述

研究的主要内容：

在计算机上从数据中产生“**模型**（model）”的算法——**学习算法**（learning algorithm）

模型：泛指从数据中学得的结果

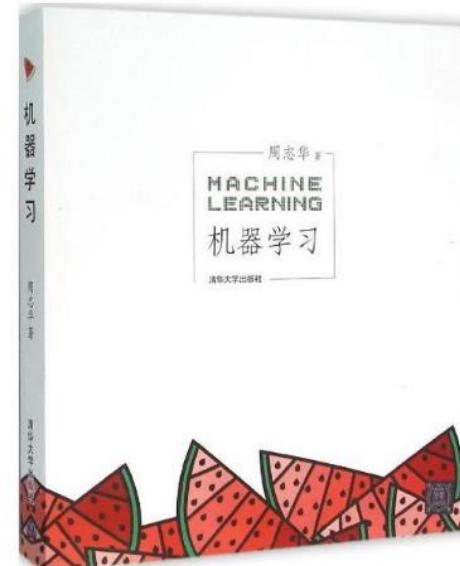
5.1 术语与模型评估

5.2 线性模型

5.3 神经网络

5.4 支持向量机

5.5 聚类

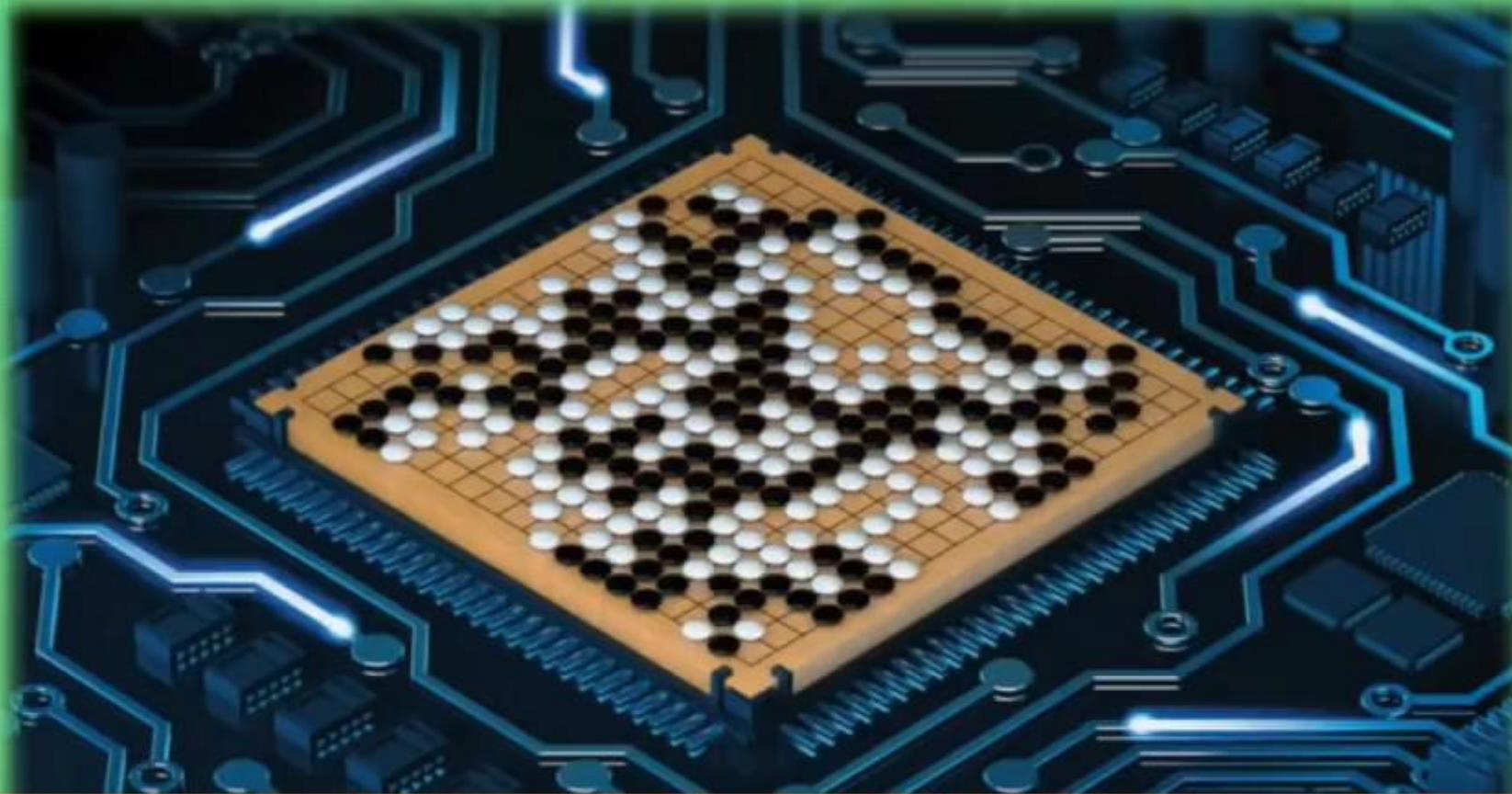


<http://open.163.com/special/opencourse/machinelearning.htm> (斯坦福大学机器学习课程)

# 什么是机器学习

## *Machine Learning*





# 什么是强化学习 (Reinforcement Learning)

What is Reinforcement Learning?



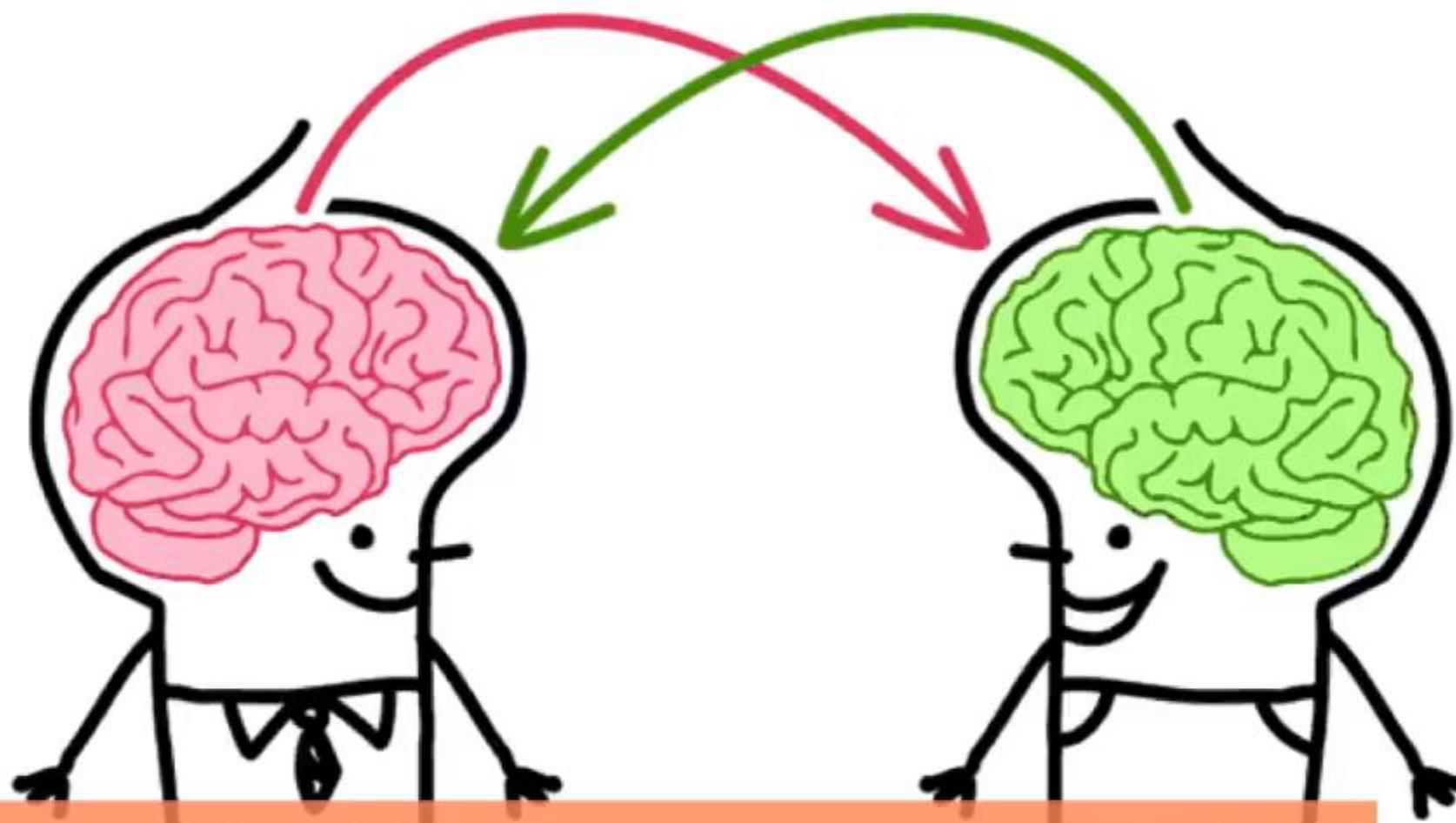
强化学习方法汇总

Reinforcement Learning Methods

什么是 Q Learning ?

What is Q Learning?





为什么迁移学习火了？

Why does Transfer Learning become popular?



## Subsection:

# 7.1 术语与模型评估

- 7.1.1 机器学习术语
- 7.1.2 经验误差与过拟合
- 7.1.3 评估方法
- 7.1.4 性能度量



## 7.1.1 机器学习术语

### ➤ 数据与数据集 (data set)

- 在数据集中，每一条记录是关于一个事件或者对象的描述，称为示例 (instance) 或样本 (sample)
- 反映事件或对象在某方面的表现或者性质的事项，称为属性 (attribute) 或特征 (feature)
- 属性上的取值，称为属性值 (attribute value)

编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	浊响	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否



## 7.1.1 机器学习术语

### ➤ 数据与数据集 (data set)

- 属性张成的空间称为属性空间 (attribute space)、样本空间 (sample space) 或输入空间，每个示例都可以在属性空间中找到自己的坐标位置
- 空间中的每个点对应一个坐标向量，示例也称为特征向量 (feature vector)

编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	浊响	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否



## 7.1.1 机器学习术语

### ➤ 数据集的数学描述

- 包含m个样本的数据集:  $D = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$
- 每个样本由d个属性描述, 样本是d维空间X中的一个向量

$$\mathbf{x}_i = (x_{i1}; x_{i2}; \dots; x_{id})$$

- d称为样本的维数 (dimensionality)

编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	浊响	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否



## 7.1.1 机器学习术语

### ➤ 机器学习的基本概念

- 从数据中学得模型的过程称为**学习**（learning）或**训练**（training），这个过程通过执行某个学习算法完成
- 训练中使用的数据称为**训练数据**（training data），每个样本称为**训练样本**（training sample），训练样本组成的集合称为**训练集**（training set）
- 学得的模型对应了关于数据的某种潜在规律，称为**假设**（hypothesis），潜在规律自身称为**真相或事实**（ground truth），学习过程就是为了找出或逼近真相。
- 模型又称为**学习器**（learner），可看作学习算法在给定数据和参数空间上的实例化
- 通过模型，可以进行**预测**（prediction）获得标记信息，使用其进行预测的过程称为**测试**（testing）
- 被预测的样本称为**测试样本**（testing sample）



## 7.1.1 机器学习术语

### ➤ 标记信息

- 样本的结果信息，称为标记（label）
- 拥有标记信息的样本，称为样例（example） $(x_i, y_i)$
- 标记的集合称为标记空间（label space）或输出空间

编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	浊响	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否



## 7.1.1 机器学习术语

- 学习任务的分类
  - 监督学习 (supervised learning)：分类与回归
  - 无监督学习 (unsupervised learning)：聚类
- 机器学习的目标
  - 使学得模型能够很好的适用于新样本，而不是仅仅在训练样本上工作好
  - 学得模型适用于新样本的能力称为泛化 (generalization) 能力
- 训练集与样本空间
  - 通常假设样本空间中全体样本服从一个未知分布 (distribution)  $\mathcal{D}$ ，获得的每一个样本独立的从这个分布中采样获得，即**独立同分布** (independent and identically distributed, IID)



## 7.1.2 经验误差与过拟合

### ➤ 基本概念

➤ 错误率 (error rate) : 分类错误的样本数占样本总数的比例

$$E = a/m$$

➤ 精度 (accuracy) :  $E = 1 - a/m$

➤ 学习器的实际预测输出与样本的真实输入之间的差异称为“误差” (error)

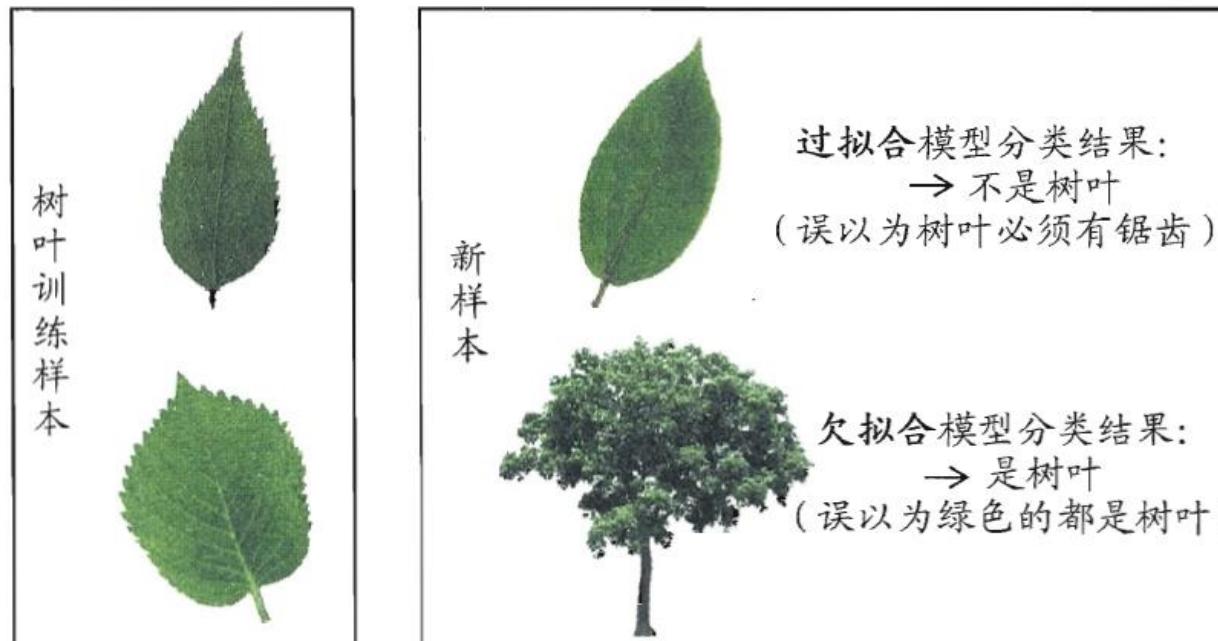
➤ 学习器在训练集合上的误差称为训练误差 (training error) 或经验误差 (empirical error)

➤ 学习器在新样本上的误差称为泛化误差 (generalization error)

## 7.1.2 经验误差与过拟合

### ➤ 过拟合 (overfitting) 与欠拟合 (underfitting)

- 过拟合：从训练样本学习得过好了
- 欠拟合：对于训练样本的一般性质尚未学好
- 过拟合是机器学习面临的关键障碍





什么是过拟合?

What is overfitting?



## 7.1.3 评估方法

- 通过实验测试对学习器的泛化误差进行评估
  - 使用测试集（testing set）测试学习器对新样本的判别能力
  - 使用测试集上的测试误差（testing error）作为泛化误差的近似
  - 注意：测试集应尽可能与训练集互斥
- 对于包含了m个样例的测试集，完成训练和测试的方法

$$D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$$



## 7.1.3 评估方法

### ➤ 1. 留出法 (hold-out)

- 直接将数据集D划分为两个互斥的集合，一个集合作为训练集S，另一个作为测试集T       $D = S \cup T, S \cap T = \emptyset$
- 训练测试集的划分要尽可能保持数据分布的一致性，一般采用保留类别比例的采样方式——分层采样
- 为了使留出法的估计结果更加稳定可靠，一般采用若干次随机划分、重复进行试验评估后取平均作为留出法的评估结果

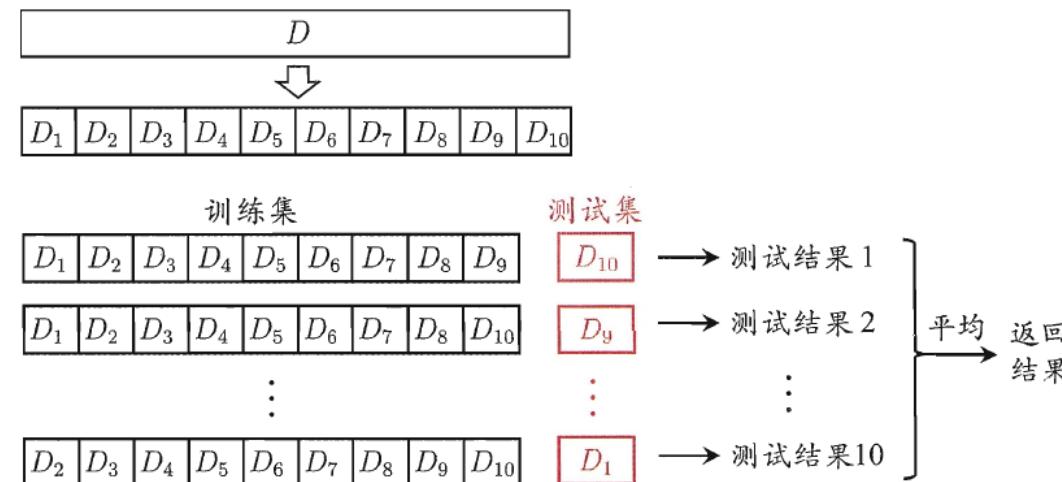
## 7.1.3 评估方法

### ➤ 2. 交叉验证法 (cross validation)

- 将数据集D划分为k个大小相似的互斥子集

$$D = D_1 \cup D_2 \cup \dots \cup D_k, D_i \cap D_j = \emptyset \quad (i \neq j).$$

- 每个子集尽可能保持数据分布的一致性
- K折交叉验证：每次用k-1个子集的并集作为训练集，余下的集合作为测试集，最终返回k个测试结果的均值





## 7.1.3 评估方法

### ➤ 3. 自助法 (bootstrapping)

- 以自主采样法为基础，对于包含 $m$ 个样本的数据集 $D$ ，采样产生数据集 $D'$ ：
  - 每次随机从 $D$ 中挑选一个样本，将其复制到 $D'$ 中，再将该样本放回初始数据集 $D$ ，从而使该样本在下次采样中仍有可能被采到
  - 重复上述过程 $m$ 次，得到包含 $m$ 个样本的数据集 $D'$
- 将 $D'$ 作为训练集， $D \setminus D'$ 作为测试集
- 自助法在数据集较小、难以有效划分训练/测试集时很有用
- 自动法产生的数据集改变了初始数据集的分布，会引入估计误差

$$\lim_{m \rightarrow \infty} \left(1 - \frac{1}{m}\right)^m \approx 0.368$$



## 7.1.4 性能度量(Performance Measure)

- 回归任务的性能度量

- 均方误差 (mean squared error)

$$E(f; D) = \frac{1}{m} \sum_{i=1}^m (f(\mathbf{x}_i) - y_i)^2 .$$

- 分类任务的性能度量

- 分类错误率: 分类错误的样本数占样本总数的比例

$$E(f; D) = \frac{1}{m} \sum_{i=1}^m \mathbb{I}(f(\mathbf{x}_i) \neq y_i) .$$

- 精度:

$$\begin{aligned} \text{acc}(f; D) &= \frac{1}{m} \sum_{i=1}^m \mathbb{I}(f(\mathbf{x}_i) = y_i) \\ &= 1 - E(f; D) . \end{aligned}$$



# 1、查准率、查全率与F1

- 二分问题的分类结果“混淆矩阵”

真实情况	预测结果	
	正例	反例
正例	$TP$ (真正例)	$FN$ (假反例)
反例	$FP$ (假正例)	$TN$ (真反例)

- 查准率 (precision)  $P = \frac{TP}{TP + FP}$  ,

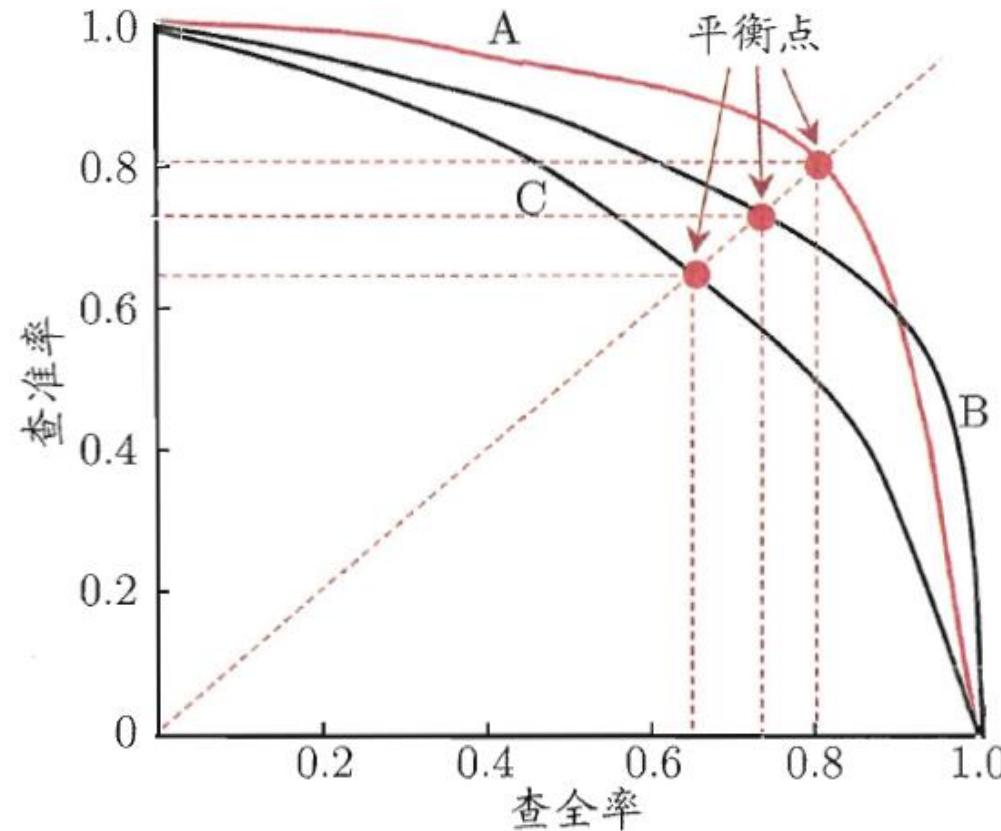
- 查全率 (recall)

$$R = \frac{TP}{TP + FN} .$$

- 查准率和查全率是一对矛盾的度量

# 1、查准率、查全率与F1

➤ P-R曲线（查准率-查全率曲线）评价学习器





# 1、查准率、查全率与F1

## ➤ 基于P-R曲线的性能度量

➤ 平衡点 (break-event point, BEP) : 查全率=查准率时的取值

## ➤ F1度量

$$F1 = \frac{2 \times P \times R}{P + R} = \frac{2 \times TP}{\text{样例总数} + TP - TN} .$$

## ➤ $F_\beta$ 度量

$$F_\beta = \frac{(1 + \beta^2) \times P \times R}{(\beta^2 \times P) + R}$$

➤ 基于P-R曲线的性能度量

➤ 平衡点 (break-event point, BEP) : 查全率=查准率时的取值

➤ F1度量

➤  $F_\beta$ 度量



# 1、查准率、查全率与F1

- 多个二分类混淆矩阵的查全/查准率
  - 宏查准率、宏查全率与宏F1
  - 微查准率、微查全率与微F1

$$\text{macro-}P = \frac{1}{n} \sum_{i=1}^n P_i ,$$

$$\text{macro-}R = \frac{1}{n} \sum_{i=1}^n R_i ,$$

$$\text{macro-}F1 = \frac{2 \times \text{macro-}P \times \text{macro-}R}{\text{macro-}P + \text{macro-}R}$$

$$\text{micro-}P = \frac{\overline{TP}}{\overline{TP} + \overline{FP}} ,$$

$$\text{micro-}R = \frac{\overline{TP}}{\overline{TP} + \overline{FN}} ,$$

$$\text{micro-}F1 = \frac{2 \times \text{micro-}P \times \text{micro-}R}{\text{micro-}P + \text{micro-}R} .$$



## 2、ROC与AUC

### ➤ ROC曲线

- ROC (受试者工作特征, Receiver Operating Characteristic)
- 将测试样本进行排序, 最可能是正例的排在最前面, 最不可能是正例的排在最后面, 设置分类阈值(截断点), 将样本分为两部分
- 排序本身质量的好坏, 体现了学习器在不同任务下的期望泛化能力
- ROC曲线用来研究学习器的泛化性能

## 2、ROC与AUC

- ROC曲线

- 横轴: 假正例率FPR

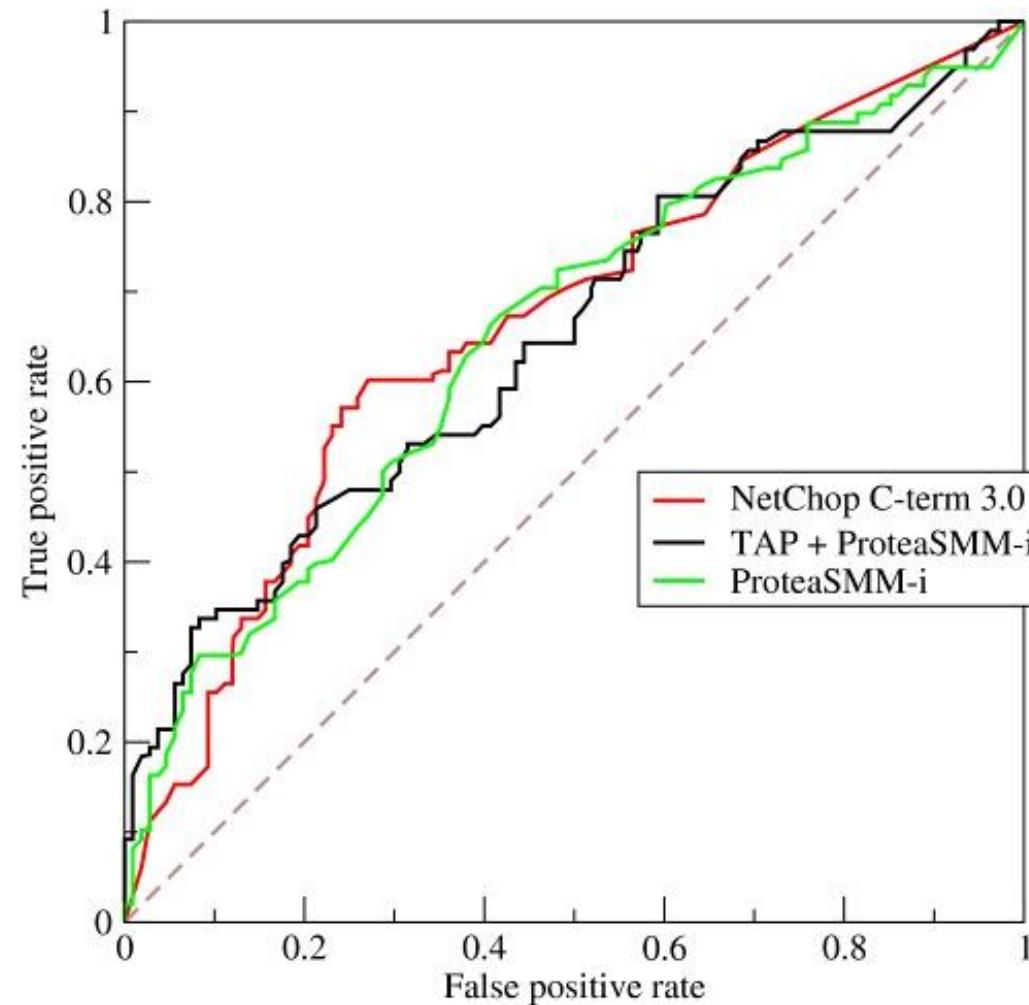
- 纵轴: 真正例率TPR

$$TPR = \frac{TP}{TP + FN} ,$$

$$FPR = \frac{FP}{TN + FP} .$$

- AUC

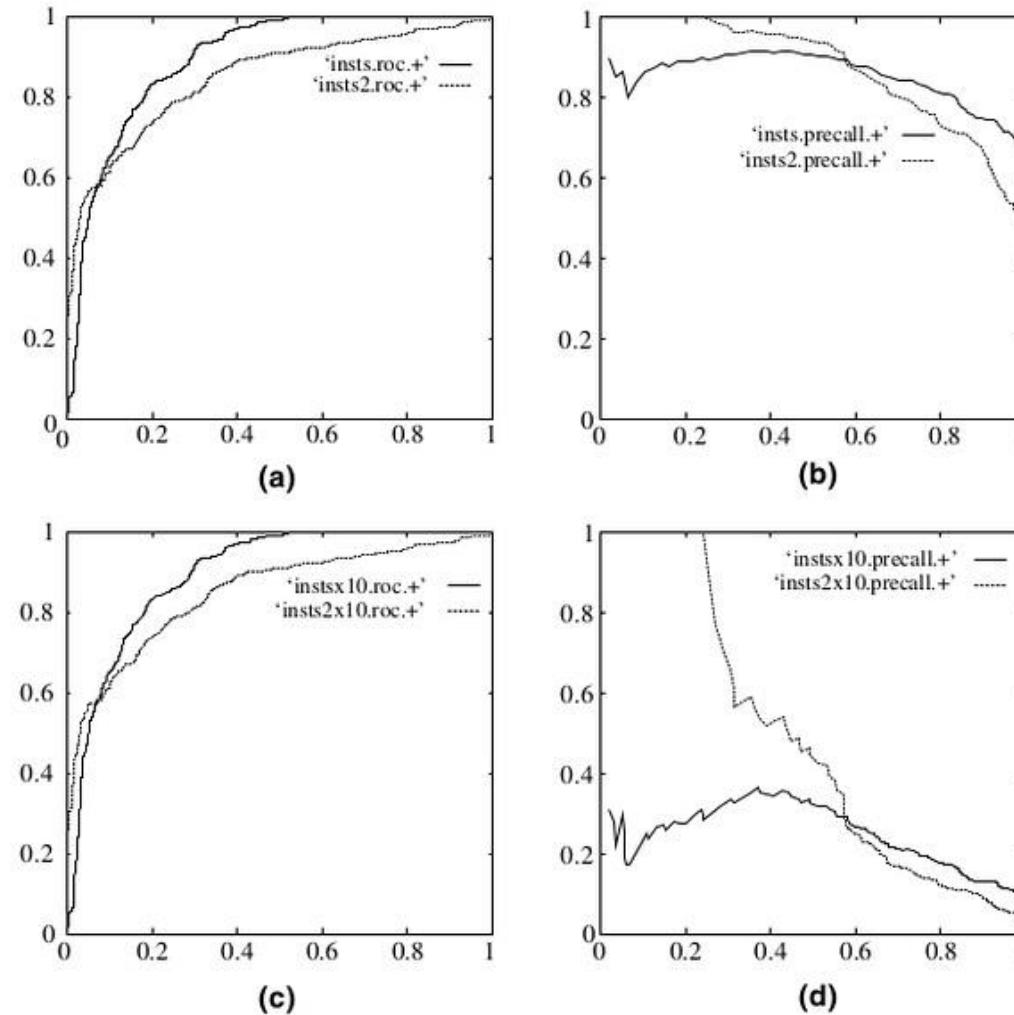
- area under ROC curve



## 2、ROC与AUC

### ➤ ROC曲线的特性

当测试集中的正负样本的分布变化的时候，  
ROC曲线能保持不变





**Subsection:**

## 7. 2 线性模型

7. 2. 1 线性回归

7. 2. 2 分类与Logistic回归

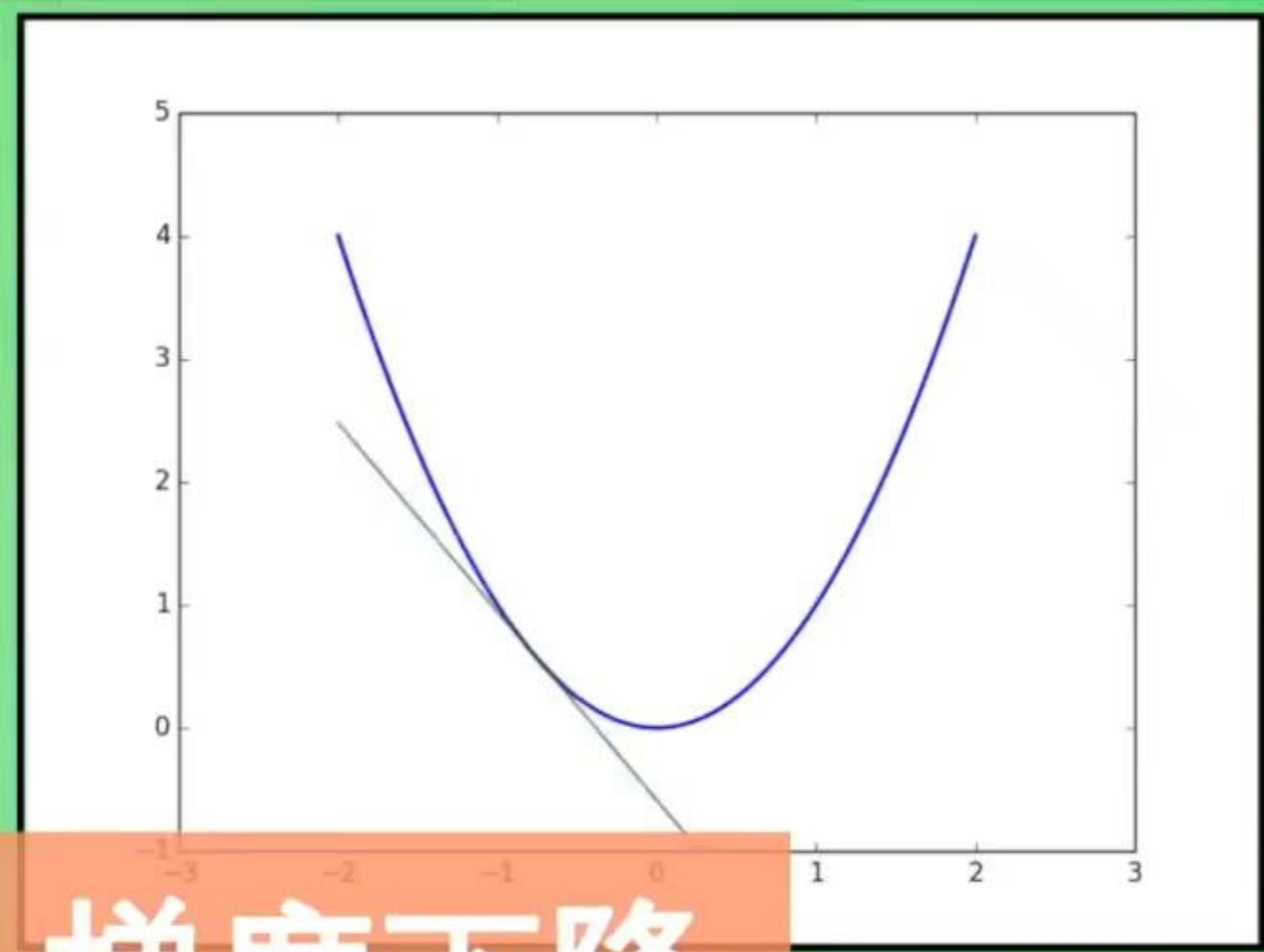
7. 2. 3 多分类学习

# Machine Learning

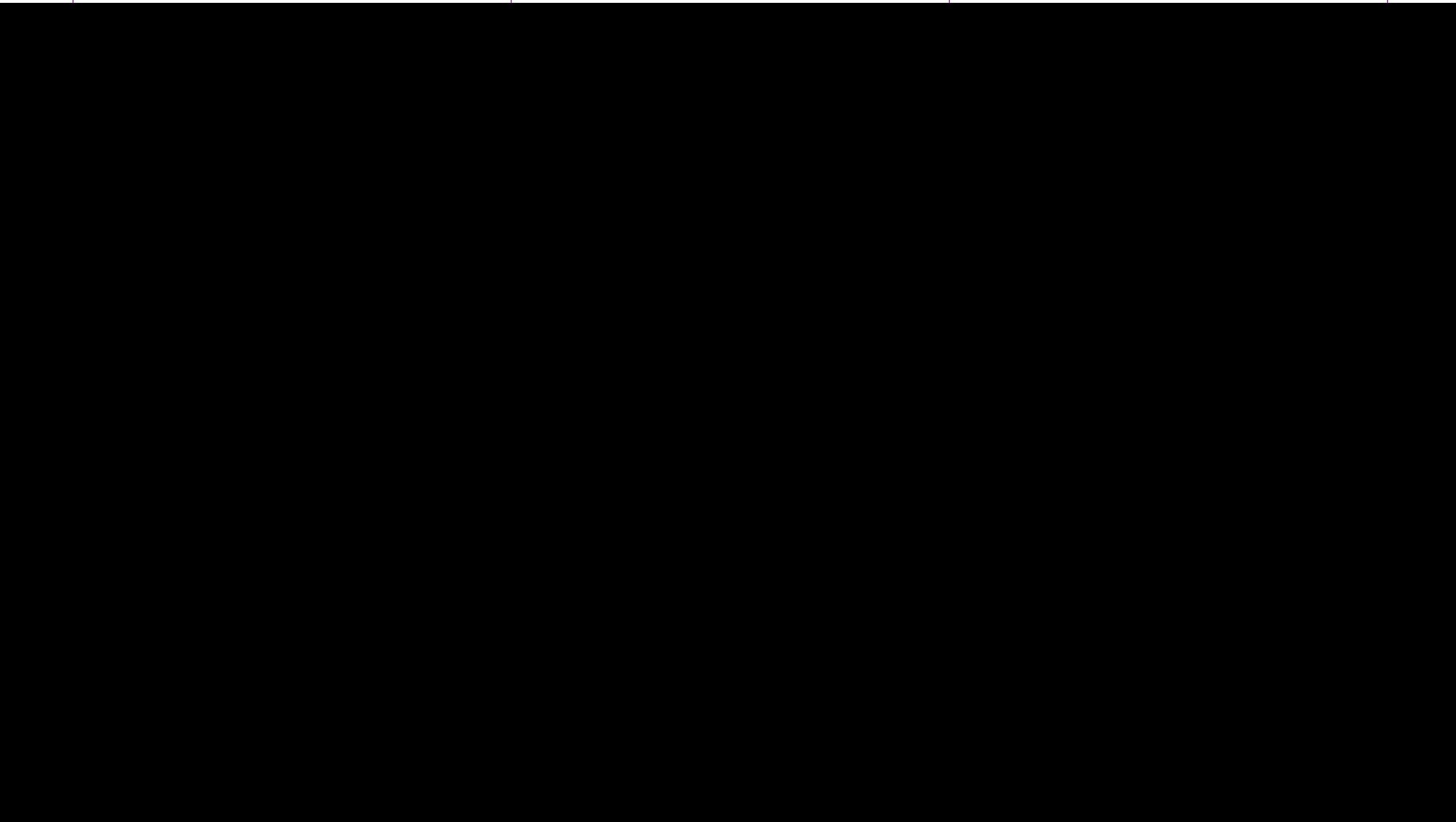


EVEREAST

# 神经网络：梯度下降



Gradient Descent in Neural Nets





# 线性回归基本形式

- 给定由 $d$ 个属性描述的示例 $\mathbf{x} = (x_1; x_2; \dots; x_d)$ , 其中 $x_i$ 是 $\mathbf{x}$ 在第 $i$ 个属性上的取值, 线性模型(linear model)试图学得一个通过属性的线性组合来进行预测的函数, 即

$$f(\mathbf{x}) = w_1 x_1 + w_2 x_2 + \dots + w_d x_d + b$$

一般用向量形式写成:

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

- 线性模型有很好的可解释性(comprehensibility)



## 7.2.1 线性回归

➤ 给定数据集  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ ,

其中  $x_i = (x_{i1}; x_{i2}; \dots; x_{id})$ ,  $y_i \in R$

线性回归 (linear regression) 试图学得一个线性模型以尽可能准确的预测实值输出标记。

➤ 线性模型举例：房屋的价格

Living area (feet <sup>2</sup> )	#bedrooms	Price (1000\$s)
2104	3	400
1600	3	330
2400	3	369
1416	2	232
3000	4	540
:	:	:



## 7.2.1 线性回归

- 线性模型 (linear model) 预测函数是属性的线性组合

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$



$$h(x) = \sum_{i=0}^n \theta_i x_i = \theta^T x$$

- 定义代价函数，最小化 $h(x)$ 和 $y$ 的差别

$$J(\theta) = \frac{1}{2} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2.$$

求解 $\theta$ 使代价函数 $J(\theta)$ 最小化的过程，称为线性回归模型的最小二乘  
(least square method) 参数估计 (parameter estimation)



## 7.2.1 线性回归

- 代价函数最小化

$$J(\theta) = \frac{1}{2} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2.$$

- 梯度下降 (gradient descent) 算法

$$\begin{aligned}\theta_j &:= \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta). \\ &= \theta_j + \alpha (y^{(i)} - h_\theta(x^{(i)})) x_j^{(i)}.\end{aligned}$$

- 使用上述公式同时处理所有的未知  $\theta_j, j = 0, 1, 2, \dots, n$
- $\alpha$  称为学习率 (learning rate)



## 7.2.1 线性回归

- 梯度下降 (gradient descent) 算法
  - 批量梯度下降 (Batch gradient descent)

Repeat until convergence {

$$\theta_j := \theta_j + \alpha \sum_{i=1}^m (y^{(i)} - h_\theta(x^{(i)})) x_j^{(i)} \quad (\text{for every } j).$$

}

- 随机/增量梯度下降 (Stochastic/Incremental gradient descent)

Loop {

for i=1 to m, {

$$\theta_j := \theta_j + \alpha (y^{(i)} - h_\theta(x^{(i)})) x_j^{(i)} \quad (\text{for every } j).$$

}

}



## 7.2.1 线性回归

- 最小二乘回归的概率解释

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)},$$

- $\epsilon^{(i)}$  是误差项·或者随机噪声， 满足**独立同分布IID** “ $\epsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$ .”

$$p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\epsilon^{(i)})^2}{2\sigma^2}\right).$$

- 则y的概率分布为

$$p(y^{(i)}|x^{(i)}; \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right).$$

- 似然函数

$$L(\theta) = L(\theta; X, \vec{y}) = p(\vec{y}|X; \theta).$$



## 7.2.2 线性回归

➤ 最小二乘回归的概率解释

➤ 似然函数

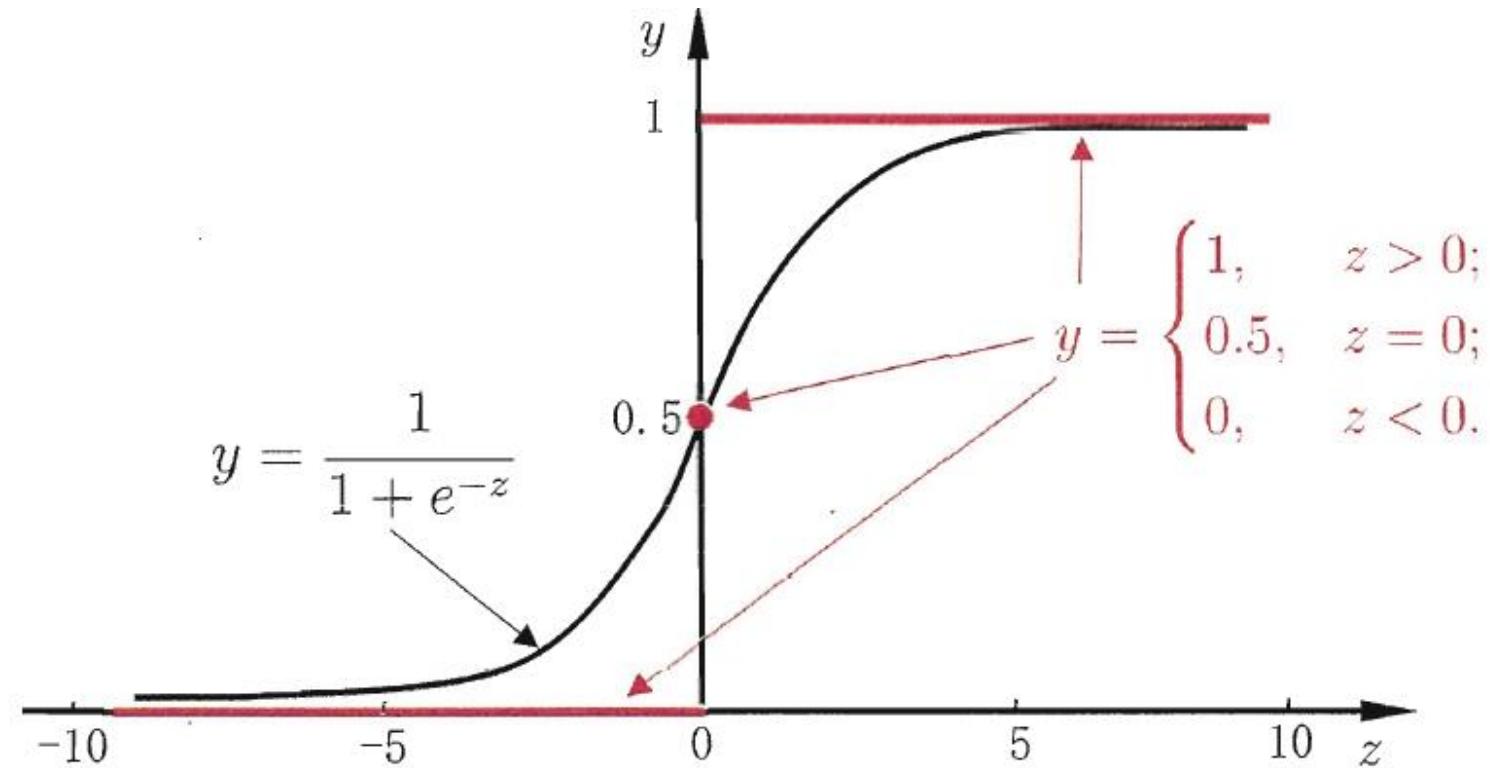
$$L(\theta) = L(\theta; X, \vec{y}) = p(\vec{y}|X; \theta).$$

$$\begin{aligned} L(\theta) &= \prod_{i=1}^m p(y^{(i)} | x^{(i)}; \theta) \\ &= \prod_{i=1}^m \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right). \end{aligned}$$

➤ 极大似然估计：选择 $\theta$ 使概率尽可能大

## 7.2.2 分类与Logistic回归

- 对于分类任务  $y \in \{0, 1\}$





## 7.2.2 分类与Logistic回归

- 对于分类任务  $y \in \{0, 1\}$
- Logistic/Sigmoid函数

$$h_{\theta}(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}},$$

$$g(z) = \frac{1}{1 + e^{-z}}$$

- Logistic函数性质

$$g'(z) = g(z)(1 - g(z)).$$



## 7.2.2 分类与Logistic回归

➤ 假设

$$P(y = 1 \mid x; \theta) = h_{\theta}(x)$$

$$P(y = 0 \mid x; \theta) = 1 - h_{\theta}(x)$$

$$h_{\theta}(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}},$$

➤ 得到y的概率分布

$$p(y \mid x; \theta) = (h_{\theta}(x))^y (1 - h_{\theta}(x))^{1-y}$$

➤ 通过极大似然估计，计算 $\theta$

$$\theta_j := \theta_j + \alpha (y^{(i)} - h_{\theta}(x^{(i)})) x_j^{(i)}$$



## 7.2.3 多分类学习

- 拆解法：将多分类任务拆为若干个二分类任务进行求解
- 拆分策略
  - 一对一 (one vs. one, OvO)
  - 一对其余 (one vs. rest, OvR)
  - 多对多 (many vs. many, MvM)
- 给定数据集

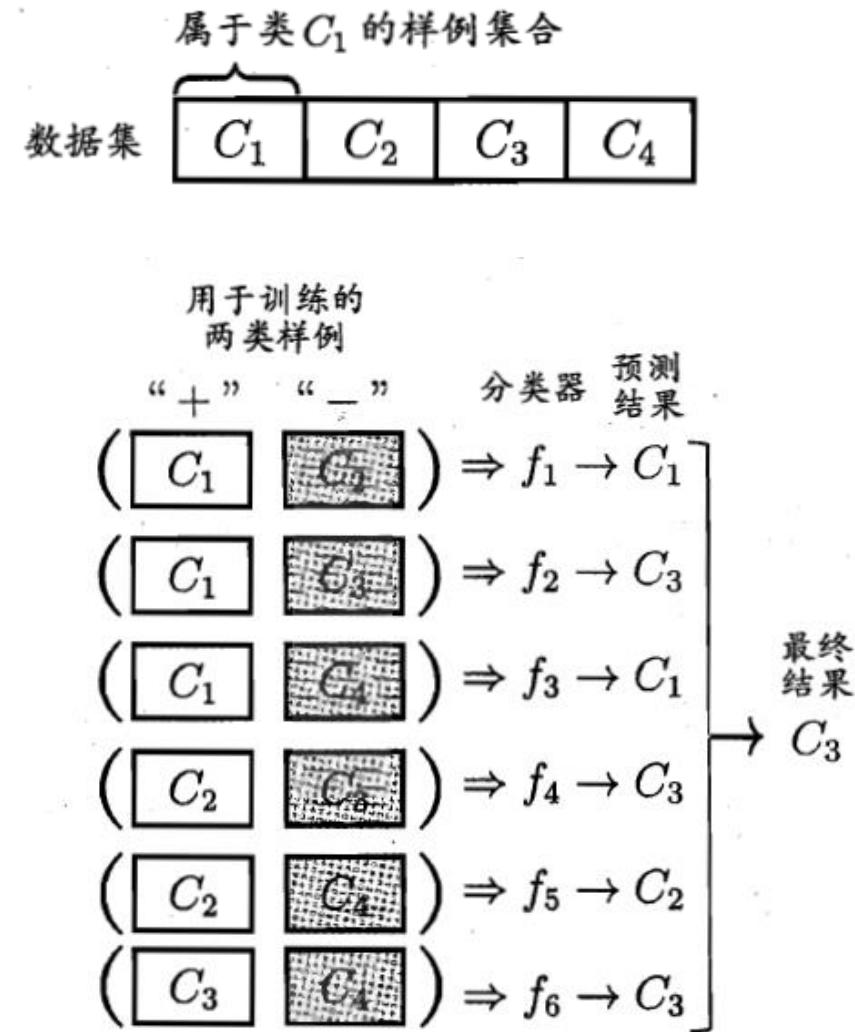
$$D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$$

$$y_i \in \{C_1, C_2, \dots, C_N\}$$

## 7.2.3 多分类学习

### ➤ 1. OvO

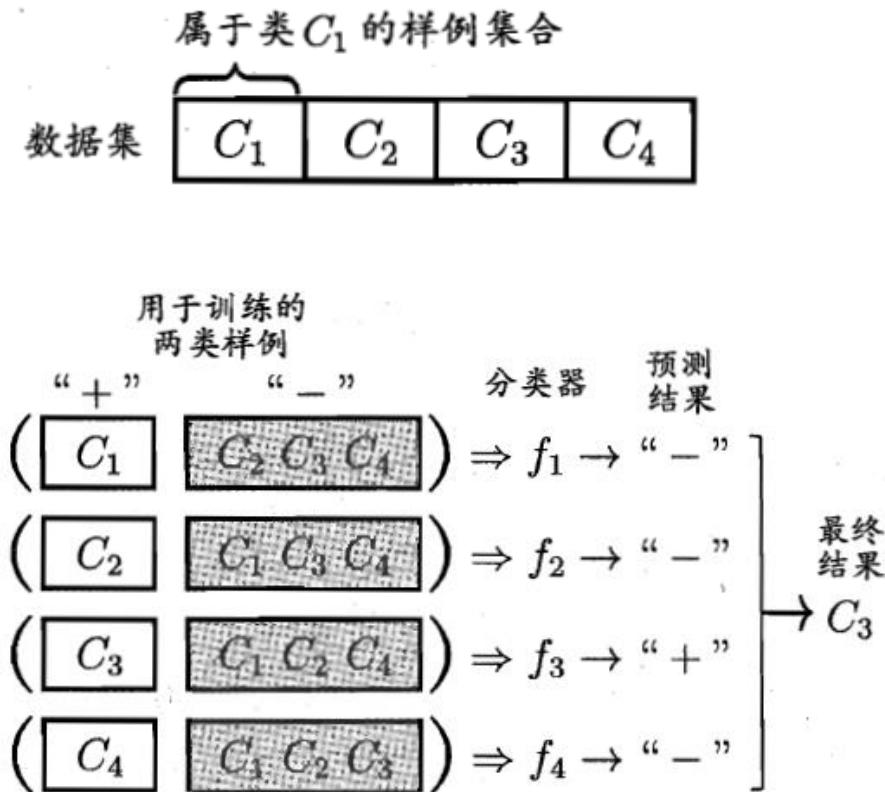
- 将N个类别两两配对，产生  $N(N-1)/2$  个分类器
- 测试阶段，将测试样本同时提交给所有分类器，得到  $N(N-1)/2$  个分类结果，把预测最多的类别作为最终分类结果



## 7.2.3 多分类学习

### ➤ 2. OvR

- 将一个类的样例作为正例，所有其他类的样例作为反例，训练N个分类器
- 测试时，若仅有一个分类器预测为正类，则对应的类别标记作为最终的分类结果
- 若有多个分类器预测为正类，则通常考虑各分类器的置信度，选择置信度最大的类别标记作为分类结果



## 7.2.3 多分类学习

### ➤ 3. MvM

- 每次将若干类作为正类、若干类作为反类利用纠错输出码构造正反类
- 编码：对N个类别做M次划分，共产生M个训练集，训练出M个分类器
- 解码：M个分类器分布对测试样本进行预测，预测标记组成一个编码，将预测编码与每个类别各自的编码进行比较，选择距离最小的类别作为最终预测结果

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	海明 距离	欧氏 距离
$C_1 \rightarrow$	-1	+1	-1	+1	+1	3	$2\sqrt{3}$
$C_2 \rightarrow$	+1	-1	-1	+1	-1	4	4
$C_3 \rightarrow$	-1	+1	+1	-1	+1	1	2
$C_4 \rightarrow$	-1	-1	+1	+1	-1	2	$2\sqrt{2}$
测试 示例	-1	-1	+1	-1	+1		

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	海明 距离	欧氏 距离
$C_1 \rightarrow$	-1	-1	+1	+1	-1	+1	+1	4	4
$C_2 \rightarrow$	-1				+1	-1		2	2
$C_3 \rightarrow$	+1	+1	-1	-1	-1	+1	-1	5	$2\sqrt{5}$
$C_4 \rightarrow$	-1	+1		+1	-1		+1	3	$\sqrt{10}$
测试 示例	-1	+1	+1	-1	+1	-1	+1		



**Subsection:**

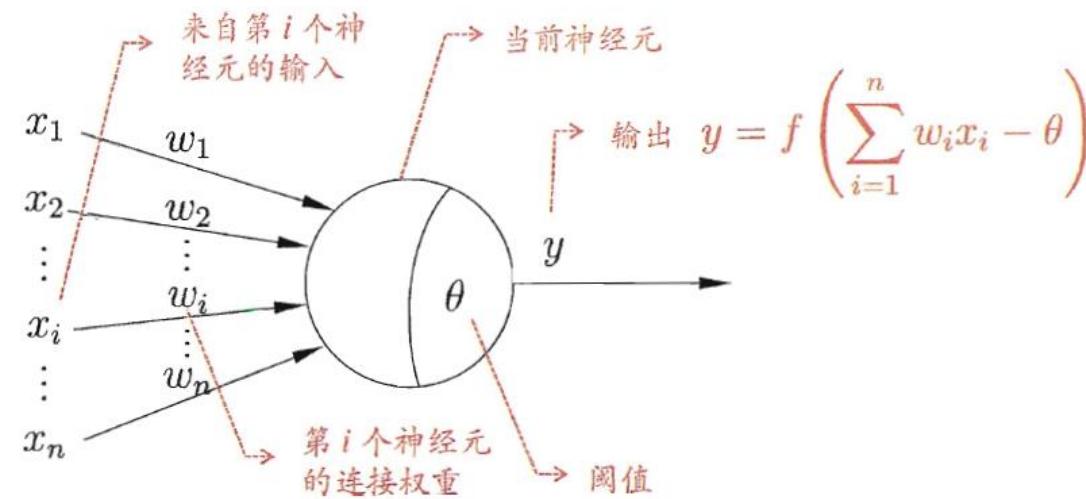
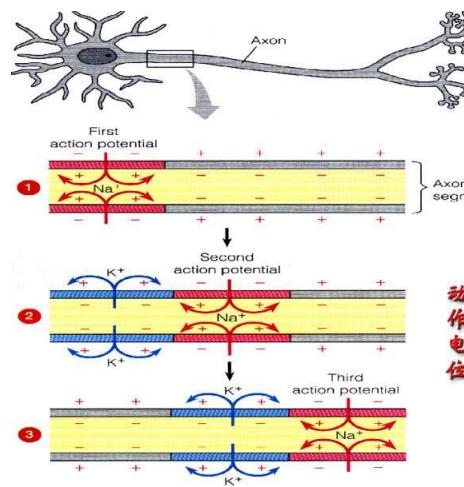
## 7.3 神经网络

- 7.3.1 神经元模型
- 7.3.2 感知机与多层网络
- 7.3.3 误差逆传播算法
- 7.3.4 深度学习\*



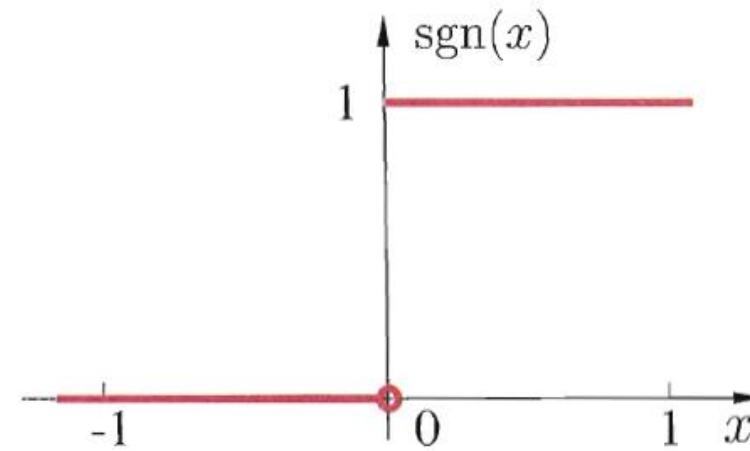
## 7.3.1 神经元模型

- 神经网络（neural networks）：是由具有适应性的简单单元组成的广泛并行互连的网络，它的组织能够模拟生物神经系统对真实世界物体所作出的交互反应
- 神经元（neuron）模型：简单单元
- 神经网络学习：根据训练数据调整神经元之间的连接权重 $\omega$ 以及功能神经元的阈值 $\theta$



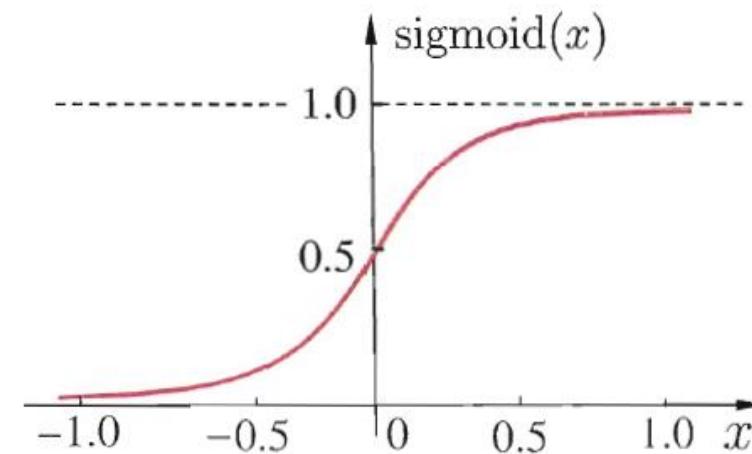
## 7.3.1 神经元模型

- 激活函数 (activation function)
  - 理想激活函数：阶跃函数 (1: 神经元兴奋, 0: 神经元抑制)
  - Sigmoid函数：将在较大范围内变化的输入值挤压到 (0,1) 输出值范围内



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

(a) 阶跃函数



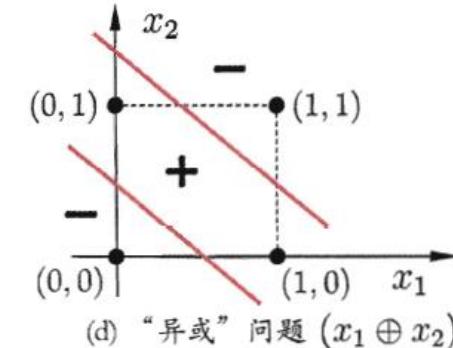
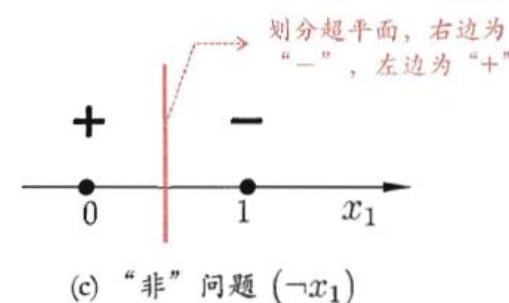
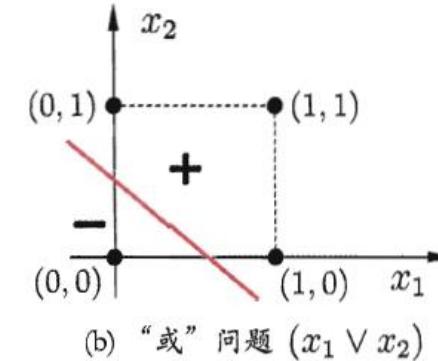
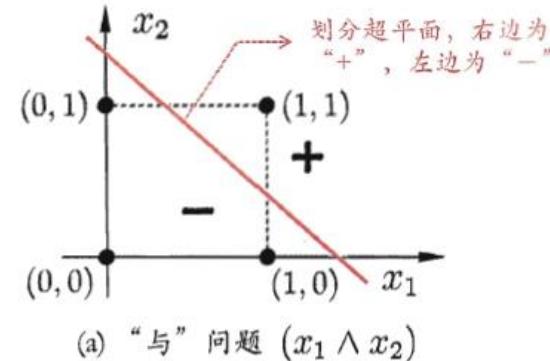
$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

(b) Sigmoid 函数

## 7.3.2 感知机与多层网络

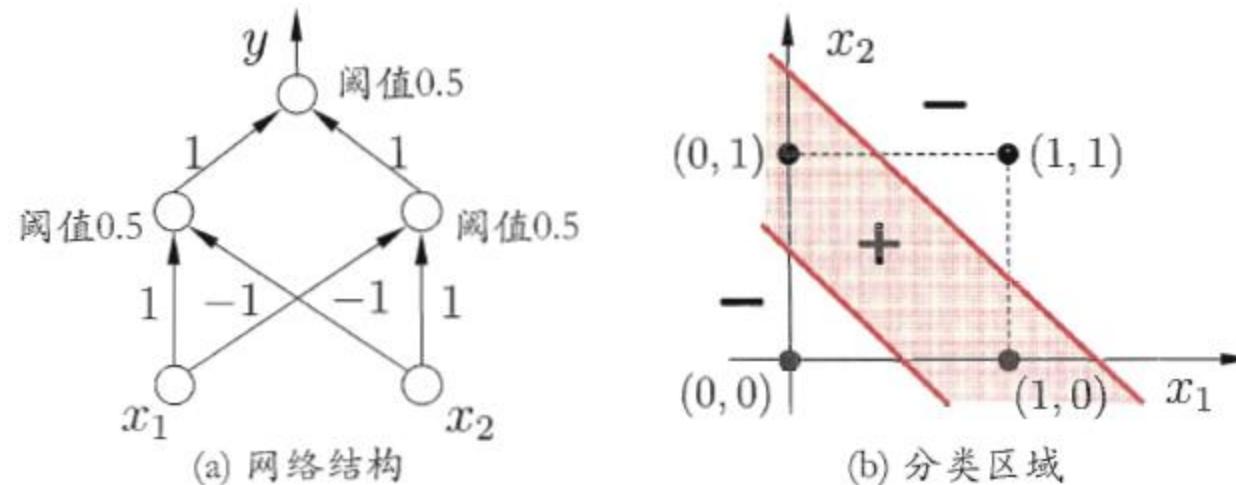
### ➤ 线性可分 (linearly separable) 问题

- 若两类模式是线性可分的，即存在一个线性超平面能将它们分开，则感知机学习过程一定会收敛 (converge)，否则感知机学习过程将发生振荡 (fluctuation)



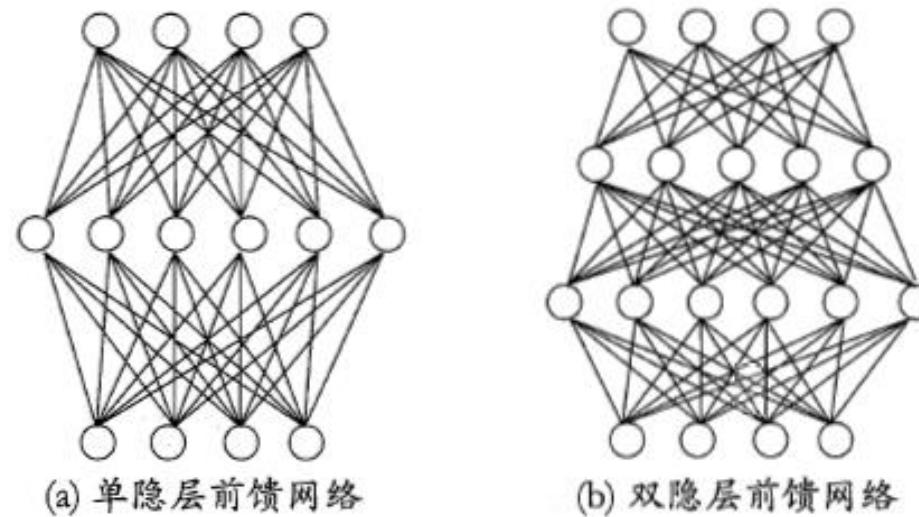
## 7.3.2 感知机与多层网络

- 非线性可分问题
  - 使用多层功能神经元
  - 输入层与输出层之间的神经元，称为隐层或隐含层（hidden layer）
  - 隐含层和输出层神经元都是拥有激活函数的功能神经元



## 7.3.2 感知机与多层网络

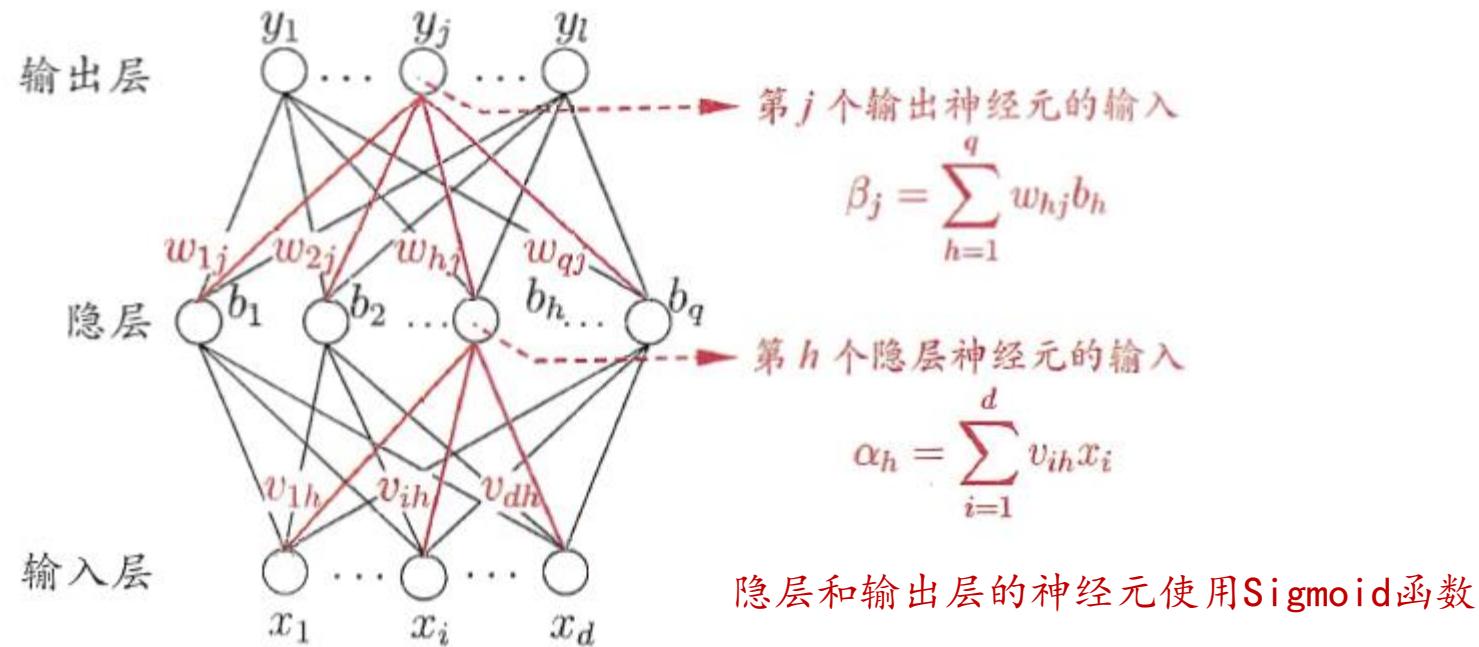
- 多层前馈神经网络 (multi-layer feedforward neural network)
  - 输入层：接受外界输入
  - 隐层与输出层：包含功能神经元，进行函数处理
  - 同层神经元不互连，每层神经元与下一层神经元全互连，不存在跨层连接



### 7.3.3 误差逆传播算法

- 利用误差逆传播 (error BackPropagation, BP) 算法实现多层网络训练
- 训练集

$$D = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_m, \mathbf{y}_m)\}, \mathbf{x}_i \in \mathbb{R}^d, \mathbf{y}_i \in \mathbb{R}^l,$$







### 7.3.3 误差逆传播算法

- BP算法是一个迭代学习算法，在迭代每一轮采用广义的感知机学习规则对参数进行更新
- 对于训练例 $(x_k, y_k)$ ，假定神经网络输出为  $\hat{y}_k = (\hat{y}_1^k, \hat{y}_2^k, \dots, \hat{y}_l^k)$  即

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

- 神经网络均方差为
  - 参数更新公式
- $$E_k = \frac{1}{2} \sum_{j=1}^l (\hat{y}_j^k - y_j^k)^2$$
- $$\beta_j = \sum_{h=1}^q w_{hj} b_h$$
- $$\alpha_h = \sum_{i=1}^d v_{ih} x_i$$
- $$v \leftarrow v + \Delta v$$

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

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

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

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

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

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



### 7.3.3 误差逆传播算法

---

**输入:** 训练集  $D = \{(x_k, y_k)\}_{k=1}^m$ ;  
学习率  $\eta$ .

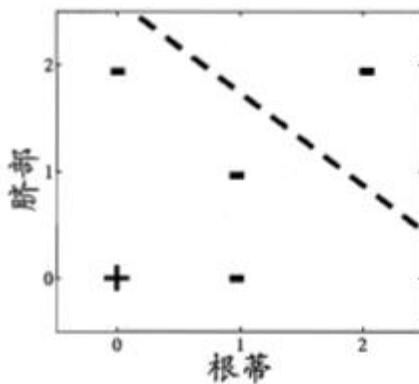
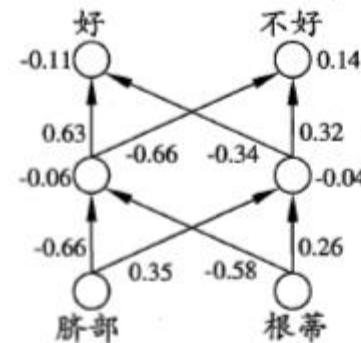
**过程:**

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

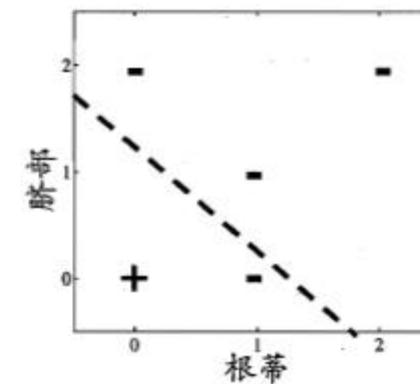
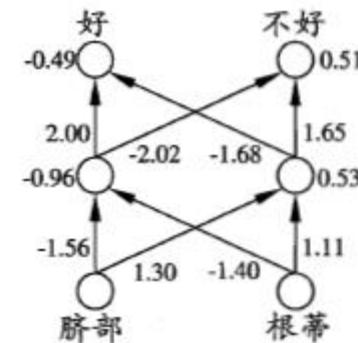
**输出:** 连接权与阈值确定的多层前馈神经网络

---

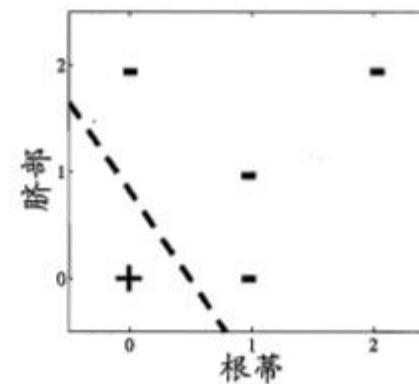
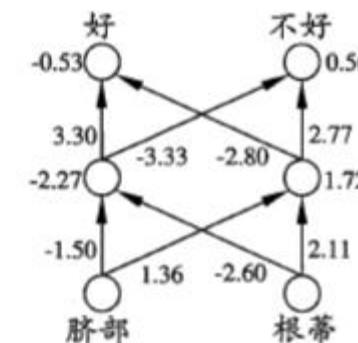
### 7.3.3 误差逆传播算法



(a) 第 25 轮



(b) 第 50 轮



(c) 第 100 轮



### 7.3.3 误差逆传播算法

➤ 标准BP算法

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

➤ 累计 (accumulate error backpropagation) 误差逆传播算法

$$E = \frac{1}{m} \sum_{k=1}^m E_k$$

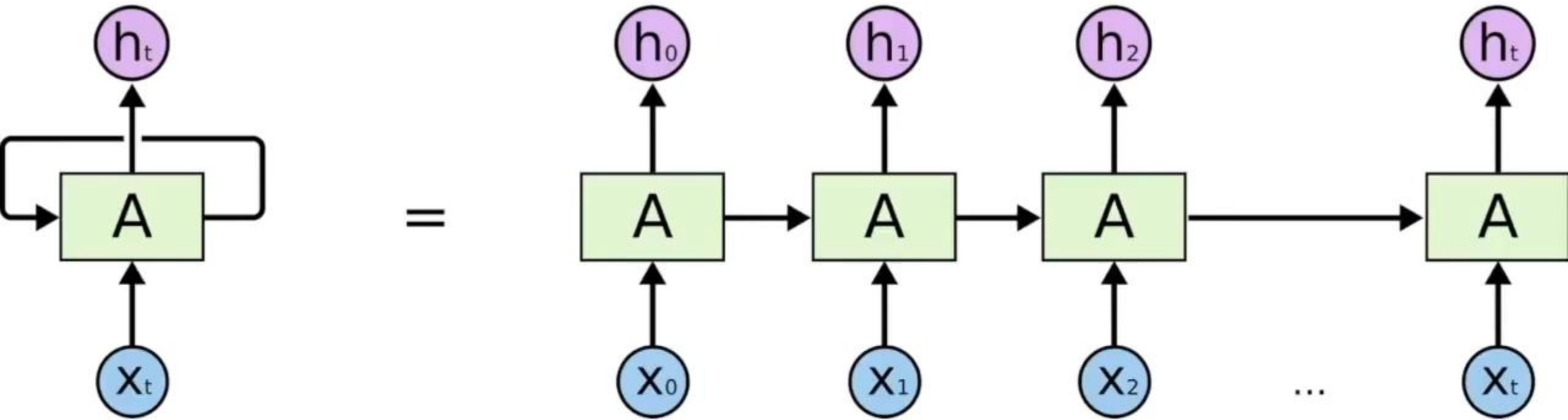


## 7.3.4 深度学习\*

- 典型深度学习 (deep learning)是很深层的神经网络，具有多个隐含层。
- 无监督逐层训练 (unsupervised layer-wise training)
  - 是多隐层网络训练的有效手段
  - 预训练 (pre-training)：每次训练一层隐节点，训练时将上一层隐节点的输出作为输入，本层隐节点的输出作为下一层隐节点的输入
  - 微调 (fine-tuning)：预训练全部接受后，对整个网络进行微调训练
- 权共享 (weight sharing)
  - 举让多组神经元使用相同的连接权
  - 例：卷积神经网络 (convolutional neural network, CNN)

# 什么是卷积神经网络？

What is the Convolutional Neural Network?

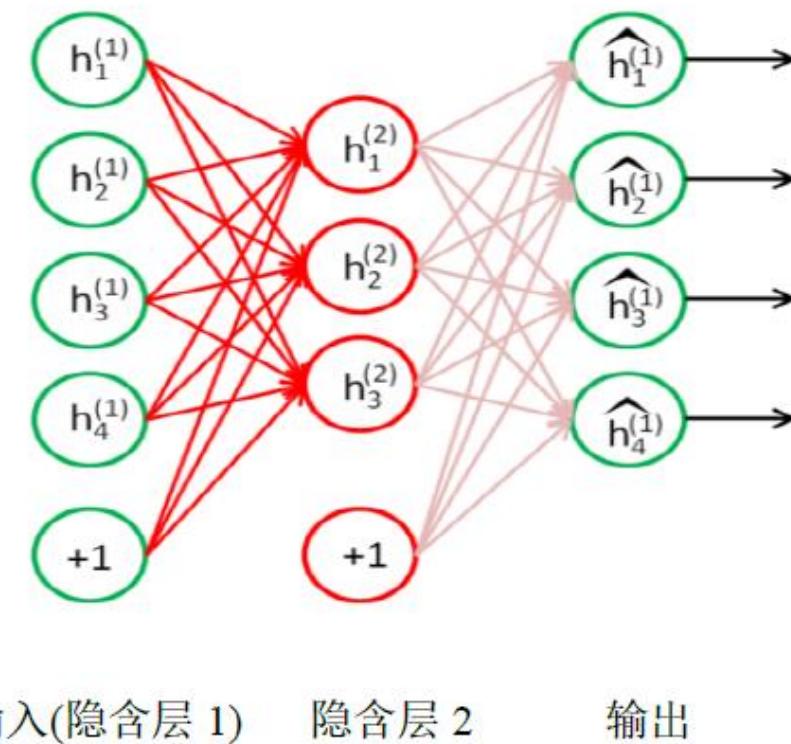
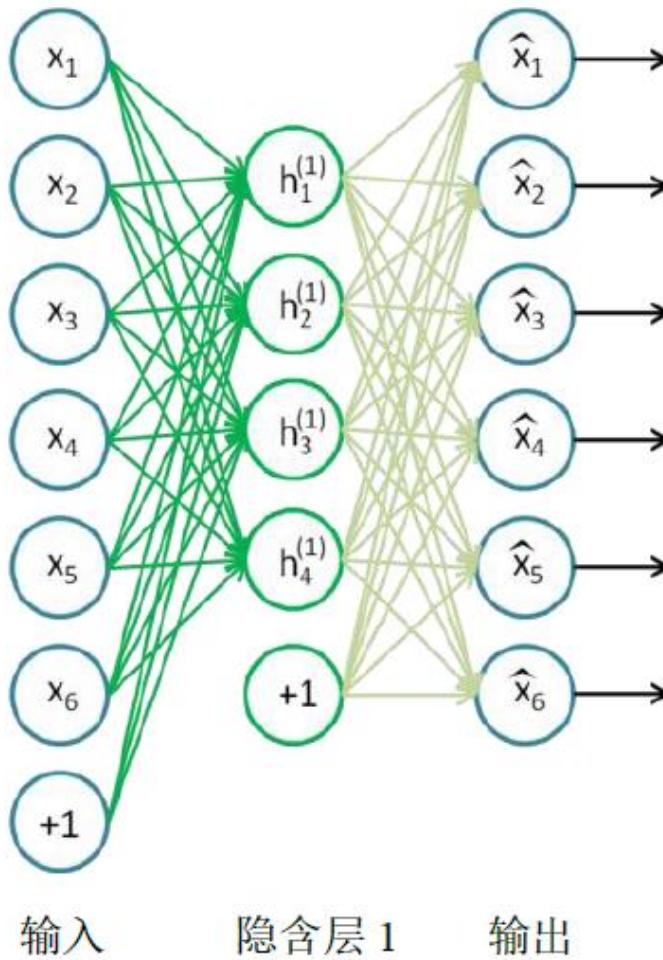


# 什么是循环神经网络RNN

What is Recurrent Neural Networks (RNN)?

## 7.3.4 深度学习\*

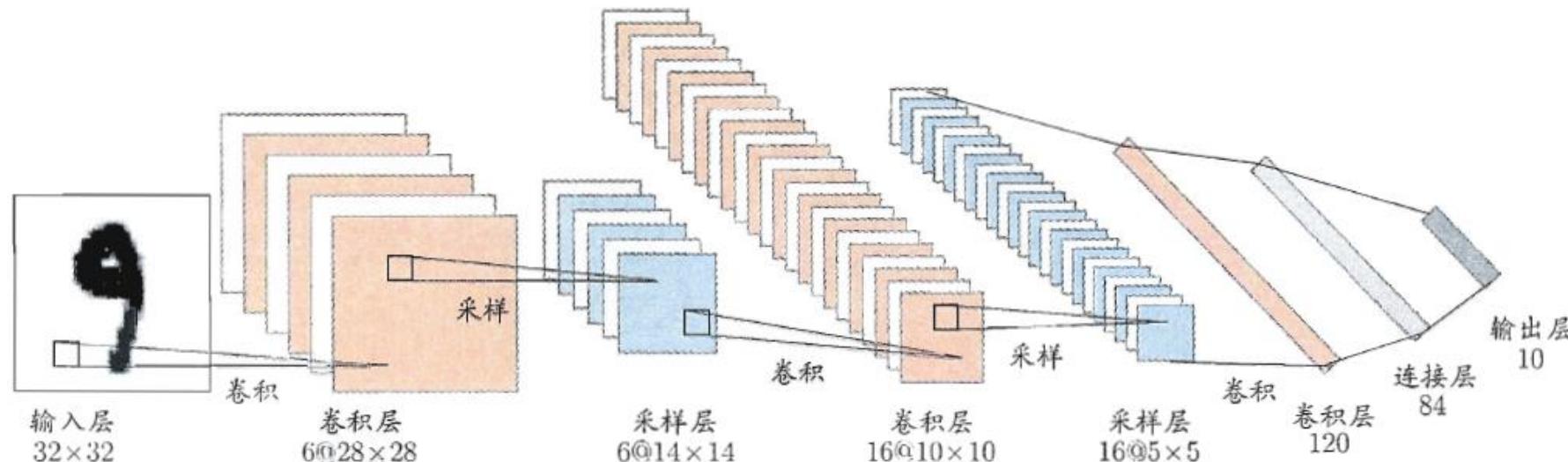
无监督逐层训练 (unsupervised layer-wise training)



## 7.3.4 深度学习\*

### ➤ 卷积神经网络

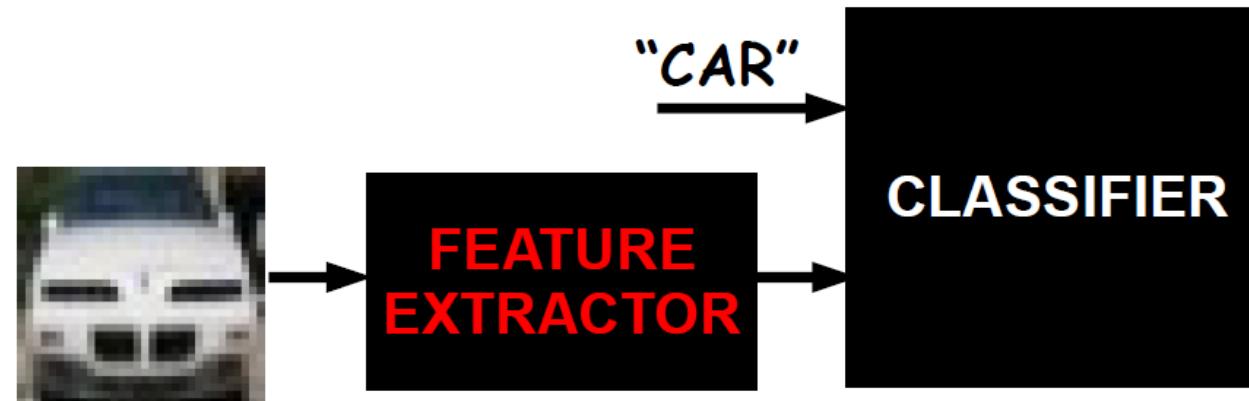
- 复合多个卷积层和采样层对输入信号进行加工，在连接层实现与输出目标之间的映射
- 卷积层：包含多个特征映射，每个特征映射是一个由多个神经元构成的平面，通过一种卷积滤波器提取输入的一种特征
- 采样层（汇合层）：基于局部相关性原理进行亚采样，从而在减少数据量的同时保留有用信息
- 每一组神经元（每个平面）使用相同的连接权，大幅减少训练的参数数目



## 7.3.4 深度学习\*

### ➤ 特征学习

- 将多隐层堆叠、每层对上层输出处理的机制，看做对输入信号的逐层加工
- 把初始的、与输出目标之间不太密切的输入表示，转化成与输出目标联系更紧密的表示
- 通过多层处理，将初始低层特征表示，转化为高层特征表示，用简单模型完成复杂的分类学习任务





**Subsection:**

## 7.4 支持向量机



找到一个线性的边界，因此需要增加第三个维度。我们创建一个z维度，定义 $z=x^2+y^2$ （圆的公式）。

间。从这个角度看，数据可以线性分为两组。因为 $z$ 是 $x$ 和 $y$ 的平方和，所以所有的 $z$ 值都是正的。

和 $x$ 轴平行，在 $z$ 某一位置的平面。选择一个距离两类数据最远的。

择的边界，就是一个圆圈，将两类数据区分开。

## 调整参数

### 核函数(KERNEL)

学习线性SVM的超平面就是通过线性代数转化问题。这是核函数扮演的角色。多项式和径向基用于更高维度。这被称为核函数技巧。

### 正则化(REGULARIZATION)

对于较大值的此参数，最好选择一个较小间距的超平面，如果这个超平面可以更好的区分训练集合点。相反地，对于非常小值的此参数，需要使用更大的间距，即使此超平面误区分更多的点。

### 系数(GAMMA)

系数定义了单个训练集合的影响程度。小的系数值，距离远的点也会用于计算。而大的系数值，更多使用距离近的点。

### 间距(MARGIN)

间距指的是到最近点的分界线。一个好的分界距离两类数据更远，且可以把一类的点区分开，而不需要穿过另外一类。

Introduction to Deep Learning Basics: Machine Learning

# 深度学习基础介绍: 机器学习

美国犹他州立大学在读计算机博士，从事机器学习，深度学习，以及计算机视觉方向的研究。

美国国家科学基金（National Science Foundation）年轻学者奖学金获得者。

在美国任教大学本科课程6年以上经验，曾在Intel & Micro Flash Technologies, TCL 硅谷北美研究院实习。

目前在佳能公司美国分公司硅谷创新中心图像计算研究所实习。

发表国际论文1篇并已经申报正在审批的图像处理方面美国以及中国共3项专利。

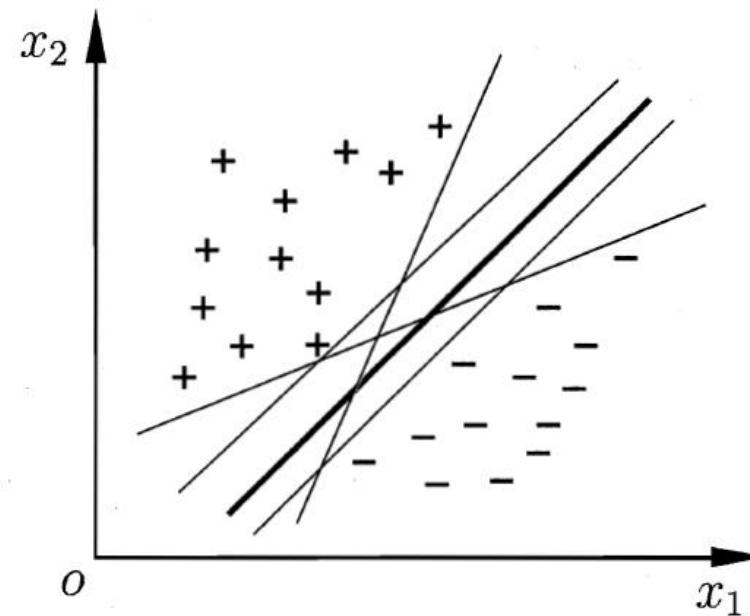


彭亮



## 7.4 支持向量机

- 间隔与支持向量 (support vector)
  - 对于给定训练样本集  $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$ ,  $y_i \in \{-1, +1\}$ ,
  - 基于训练集D在样本空间中找到一个划分超平面, 将不同类别的样本分开



## 7.4 支持向量机

- 间隔与支持向量 (support vector)

- 通过线性方程描述划分超平面:

- $\omega$ 是d维法向量，决定了超平面的方向

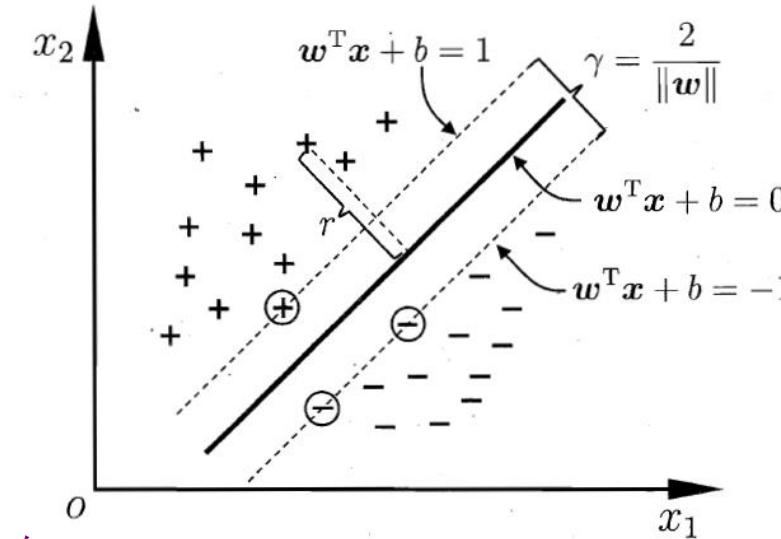
- $b$ 为位移项，决定了超平面与原点间的距离

- 样本空间中任意点 $x$ 到超平面的距离:

- 令:

- 距离超平面最近的训练样本使公式中的等号成立，称为支持向量

- 两个异类支持向量到超平面的距离之和称为间隔 (margin)





## 7.4 支持向量机

- 间隔与支持向量 (support vector)
- 欲找到具有最大间隔的划分超平面, 即寻找参数 $\omega, b$ , 使得 $\gamma$ 最大:

$$\begin{aligned} \max_{\boldsymbol{w}, b} \quad & \frac{2}{\|\boldsymbol{w}\|} \\ \text{s.t. } \quad & y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1, \quad i = 1, 2, \dots, m. \end{aligned}$$

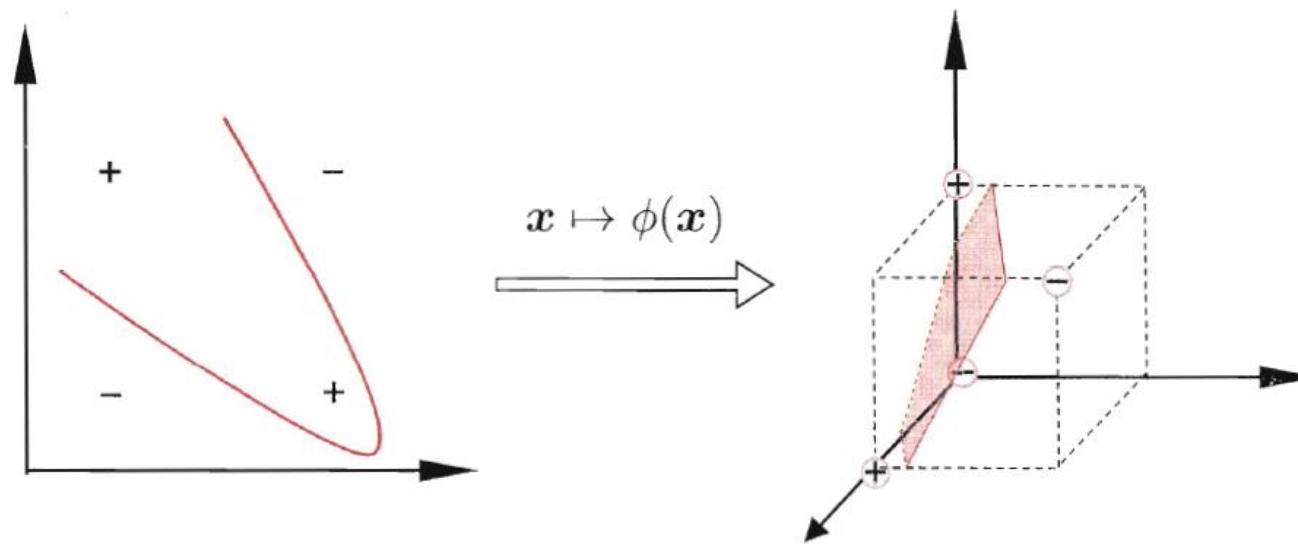
- 支持向量机的基本型:

$$\begin{aligned} \min_{\boldsymbol{w}, b} \quad & \frac{1}{2} \|\boldsymbol{w}\|^2 \\ \text{s.t. } \quad & y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1, \quad i = 1, 2, \dots, m. \end{aligned}$$

## 7.4 支持向量机

### ➤ 核函数 (kernel function)

- 原始样本空间不存在能正确划分两类样本的超平面
- 将样本从原始空间映射到一个更高维的特征空间，使得样本在这个特征空间内线性可分
- 如果原始空间是有限维，即属性数有限，一定存在一个高维特征空间使样本可分





## 7.4 支持向量机

### ➤ 核函数 (kernel function)

➤  $x$ 映射后的特征向量为 $\phi(x)$ , 特征空间中的划分超平面为:

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b ,$$

➤ 支持向量机: 
$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$

$$\text{s.t. } y_i(\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geq 1, \quad i = 1, 2, \dots, m.$$

➤ 核函数:  $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$

名称	表达式	参数
线性核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$	
多项式核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j)^d$	$d \geq 1$ 为多项式的次数
高斯核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\ \mathbf{x}_i - \mathbf{x}_j\ ^2}{2\sigma^2}\right)$	$\sigma > 0$ 为高斯核的带宽(width)
拉普拉斯核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\ \mathbf{x}_i - \mathbf{x}_j\ }{\sigma}\right)$	$\sigma > 0$
Sigmoid 核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta \mathbf{x}_i^T \mathbf{x}_j + \theta)$	$\tanh$ 为双曲正切函数, $\beta > 0, \theta < 0$



**Subsection:**

## 7.4 决策树

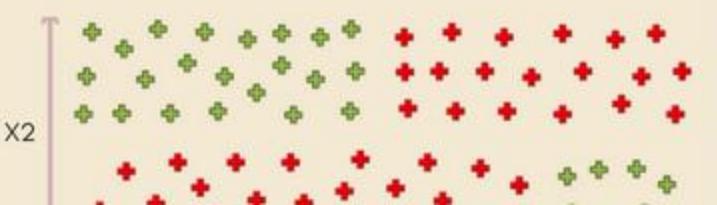
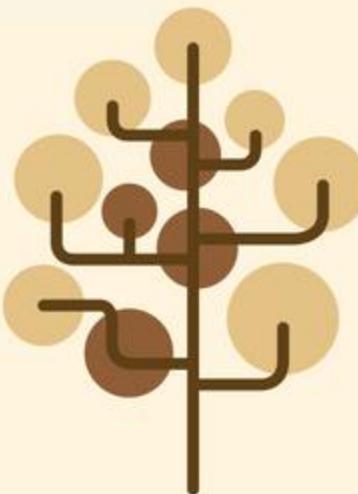
# 决策树

一篇关于分类决策树的直观介绍

## 1 决策树是什么？

它是一种监督学习算法，主要用于分类问题，适用于可分类的、连续的输入和输出变量。

决策树是这样的一种树，这棵树的每个分支节点表示多个可以选择的选项，并且每个叶节点表示最终所做的决策。



这里我们可以举一个栗子：在二维散点图上有许多的点，那么，现在这个决策树如何工作呢？



# Decision Tree

---

决策树

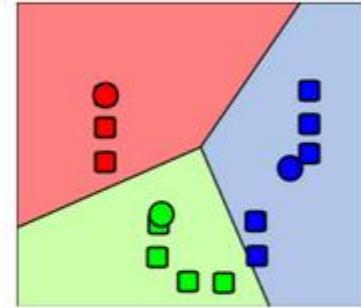
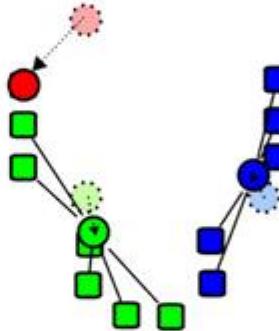
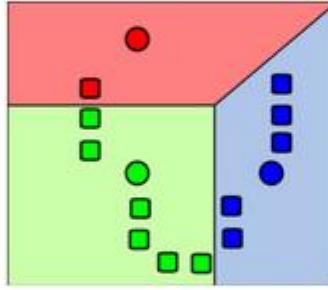
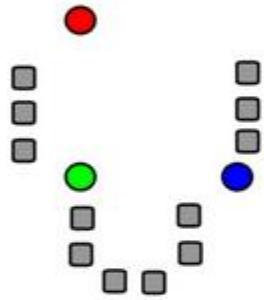




**Subsection:**

## 7.5 聚类

## 4) K-均值聚类如何工作



1. 在数据域中随机生成k个初始“均值”  
(本例中k=3)。

2. 通过关联每个观测值到最近的均值，创建k个簇。

3. 每个簇的形心变成新的均值。

4. 重复步骤2和步骤3，直到收敛。

K-均值聚类的目标是使  
总体群内方差最小，  
或平方误差函数：

目标函数  $J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2$

距离函数

簇的个数

簇内点个数

簇内第*i*个

第*j*个簇的形心

# Python数据 可视化

K-means clustering.

Starting with 4 random points in one cluster.





## 7.5 聚类

- 对于包含m个无标记的样本集D  $\{x^{(1)}, \dots, x^{(m)}\}$
- 聚类算法将样本划分为k个不相交的簇  $\{C_l \mid l = 1, 2, \dots, k\}$

$$C_{l'} \cap_{l' \neq l} C_l = \emptyset \quad D = \bigcup_{l=1}^k C_l$$

- K-means聚类

1. Initialize **cluster centroids**  $\mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^n$  randomly.
2. Repeat until convergence: {

For every  $i$ , set

$$c^{(i)} := \arg \min_j \|x^{(i)} - \mu_j\|^2.$$

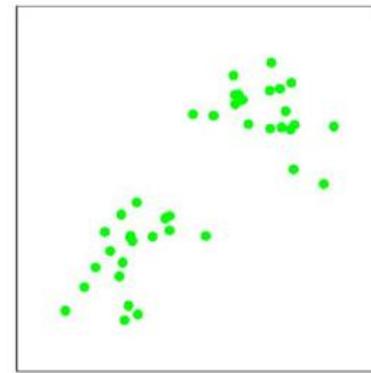
For each  $j$ , set

$$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

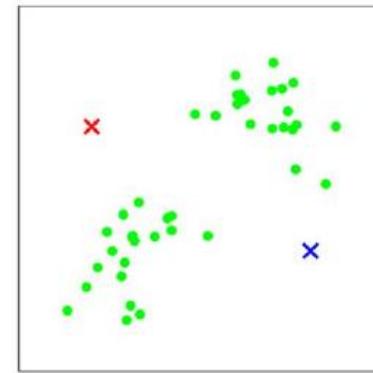
}

## 7.5 聚类

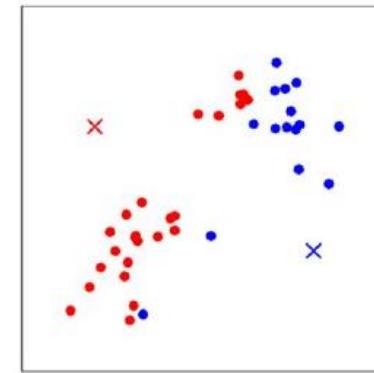
### ➤ K-means聚类



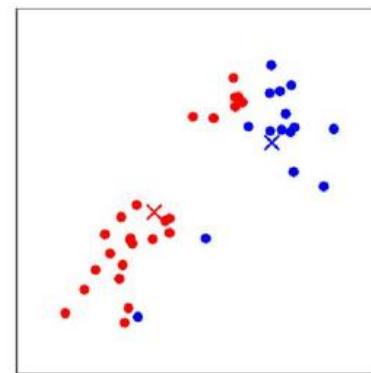
(a)



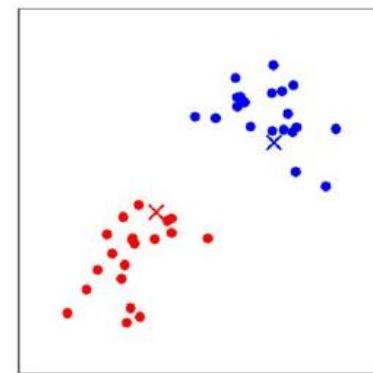
(b)



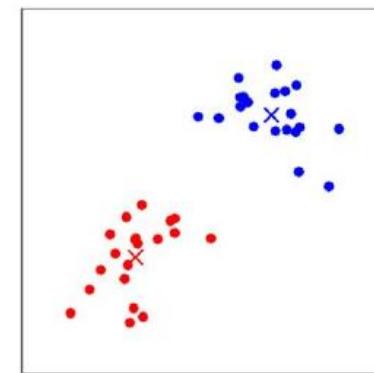
(c)



(d)



(e)



(f)

Q&A

THANKS!

