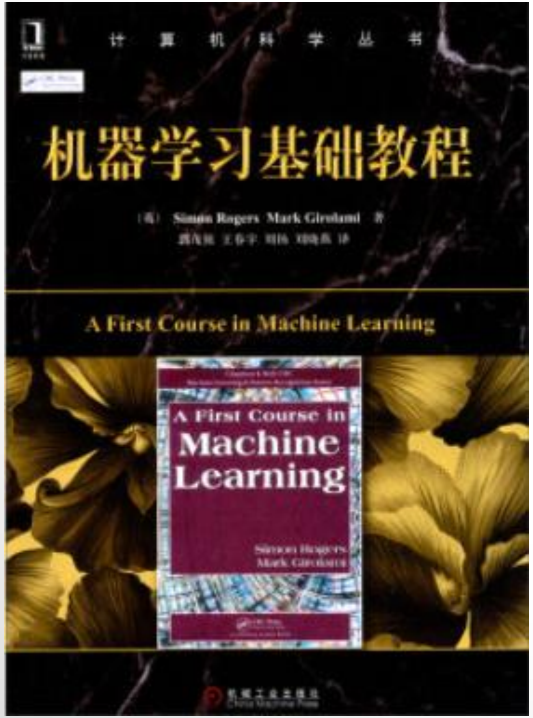
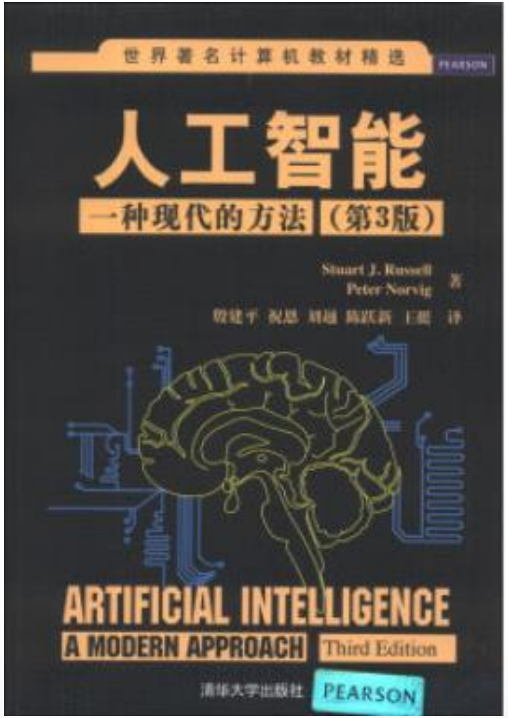
智能信息处理

主讲：张 磊
办公室：主教1030
E-mail: leizhang@cqu.edu.cn
Website: http://www.leizhang.tk

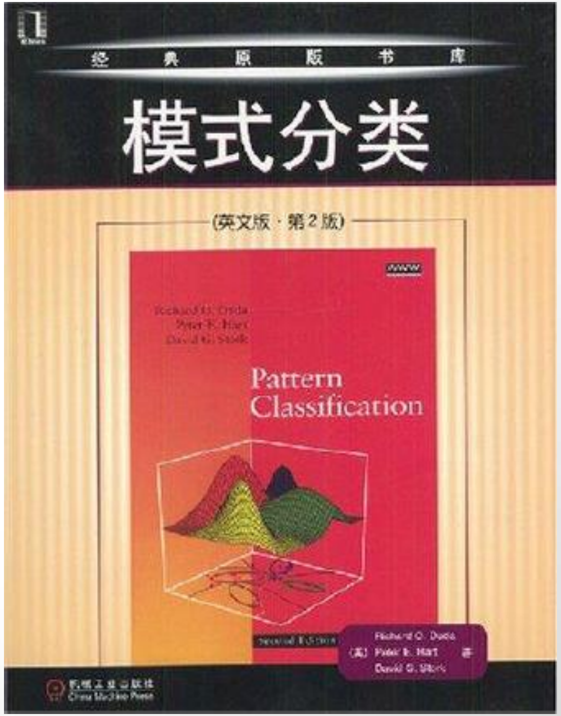
教材与参考书目



计 算 机 科 学 丛 书
机器学习基础教程
Simon Rogers Mark Girolami 著
A First Course in Machine Learning
A First Course in
Machine
Learning



世界著名计算机教材精选
PEARSON
人工智能
一种现代的方法（第3版）
Stuart J. Russell
Peter Norvig 著
ARTIFICIAL INTELLIGENCE
A MODERN APPROACH
Third Edition
清华大学出版社
PEARSON



经 典 原 版 书 库
模式分类
(英文版 · 第2版)
Pattern
Classification

参考书目

Simon Rogers, Mark Girolami编，《机器学习基础教程》，机械工业出版社

Stuart J. Russell, Peter Norvig编《人工智能—一种现代的方法》第3版，清华大学出版社，

Richard O. Duda, Peter E. Hart, David G. Stork编《模式分类》第2版，机械工业出版社，中信出版社

Tom M. Mitchell编，《机器学习》，机械工业出版社

课时安排

学时：24学时（12次）

考试：理论笔试

学分：1.5

成绩（百分制）：70%考试+30%出勤

人工智能(Artificial Intelligence, AI) 1956：

人工智能是最新兴的科学与工程领域。智能信息处理为实现人工智能的必要手段，其目标和任务是试图理解和创造智能实体。

人工智能的大量粉丝：潜在的爱因斯坦们和爱迪生们，被大多数领域的科学家们评为“最想参与的研究领域”

什么是人工智能(Artificial Intelligence, AI)：

- 电影和小说中描绘的人工智能（主宰的可怕未来）塑造了大众对人工智能的想象，但这些都是虚构的。在现实生活中，人工智能基本上是在改善人类健康、安全和提升生产力等好的方面。

- 多数研究型大学和科技公司（苹果、谷歌、facebook、IBM、微软、百度）也投入巨资，并将人工智能视为未来发展的关键。

- 好莱坞也将人工智能技术搬上荧幕（反乌托邦人工智能幻想故事）

人工智能定义：

一直以来，人工智能缺乏一个精确的、被普遍接受的定义。目前，一个有用的定义：人工智能是致力于让机器变得智能的活动，而智能就是使实体在其环境中有远见地、适当地实现功能性的能力。

什么是人工智能(Artificial Intelligence, AI)？

◆ 像人一样思考

机器要有“脑”，而且会自己思考、决策、求解、学习等。初级的人工智能，需要人类去“教”会机器如何思考，也就是利用计算机模型来研究机器的智能（智力），使机器的自动感知、推理、行为，从计算上成为可能（智能计算）。

利用计算机完成人更加擅长的事情，称为计算智能。

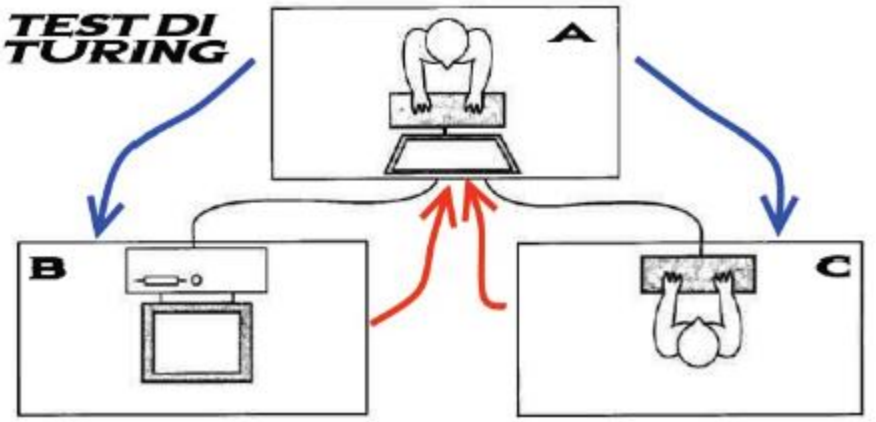
第一章：绪论

什么是人工智能(Artificial Intelligence, AI)？

◆ 像人一样行动（行为）

图灵测试：由Alan Turing1950提出，为人工智能提供了令人满意的可操作的定义。如果一位人类询问者提出一些问题后，无法判断对方的回答是来自人还是计算机，那么这台计算机通过测试（成功“骗”过人）。

60年以后，该测试仍然适合，因此值得称赞



什么是人工智能(Artificial Intelligence, AI)？

◆ 像人一样行动（行为）

为了能“骗”过人类，还需要通过大量的计算机模型和编程工作。计算机需要具备以下能力：

知识表示与存储：用于计算机去“记忆”

机器学习：对新知识的认知和学习

自动推理：思考并回答问题和推出新结论

自然语言处理：“听、说”，用于外语交流

计算机视觉：“看”世界（感知）

机器嗅觉、触觉：“嗅”和“摸”（感知）

机器人学：操纵和移动对象

⟹这些领域涉及人工智能(AI)的大部分内容

什么是人工智能(Artificial Intelligence, AI)？

◆ 像人一样行动（行为）

综上所述，要真正通过图灵测试，计算机必须具备理解语言、学习、记忆、推理、决策等能力。从而产生许多学科，如机器感知（计算机视觉、自然语言处理），机器学习（模式识别、增强学习），记忆（知识表示），决策（规划、数据挖掘），这些都属于人工智能的研究范畴。

什么是人工智能(Artificial Intelligence, AI)？

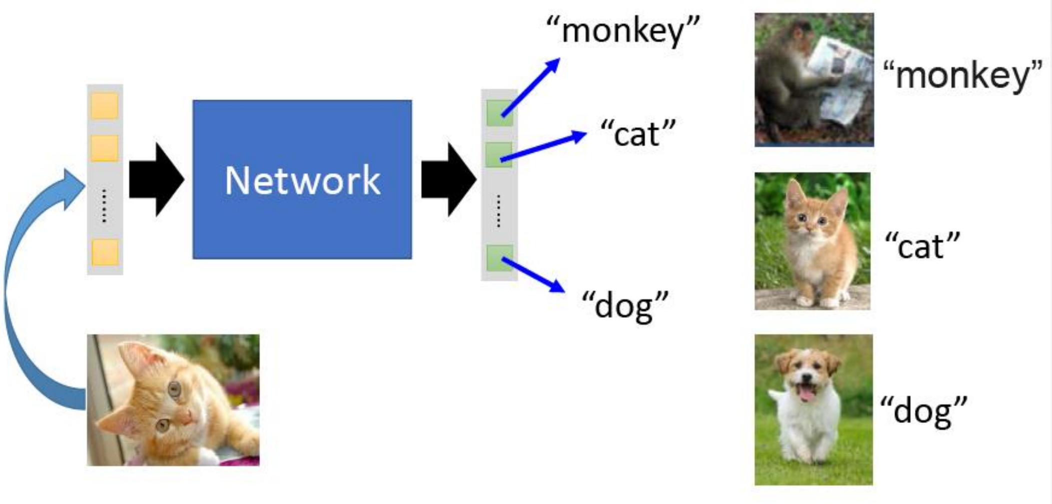
大家认为飞机/飞行器属于人工智能吗？？

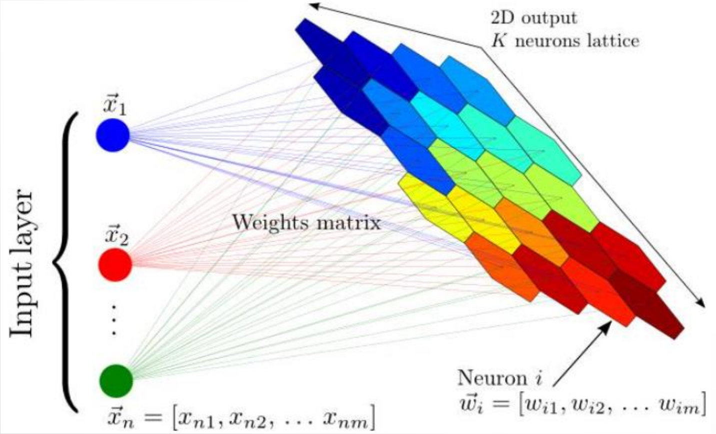
航空领域的研究人员不会把研究目标定义为制造“能完全像鸽子一样飞行的机器，使得它们可以骗过其他真鸽子”。

如何实现人工智能(AI)?

- 如何让机器像人一样“思考”？

首先，弄清楚人是如何思考，才能教会机器。这是关于逻辑学、心理学、神经科学、脑科学的复杂问题，目前尚无精确的理论。虽然人工智能通过计算机模型实现初步的思考、推理和行动，但与人脑的工作原理之间的联系尚无解释。



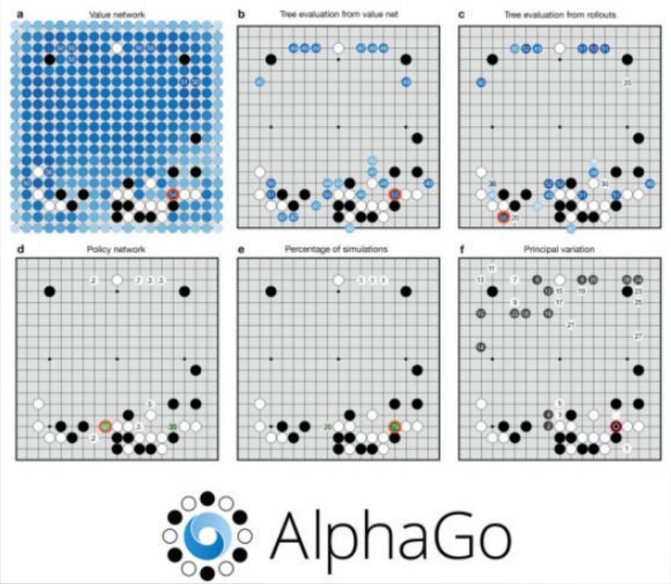


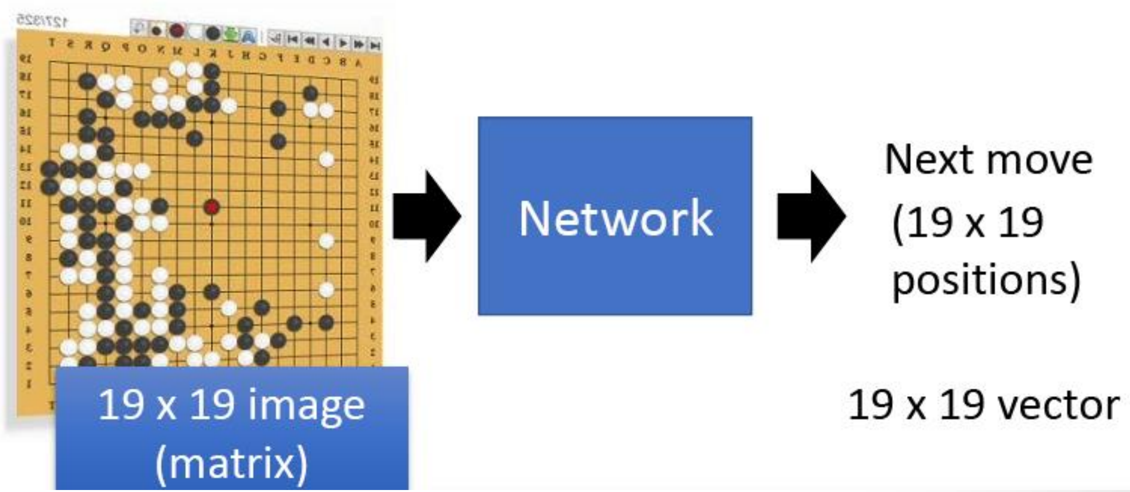
如何实现人工智能(AI)？

- 如何让机器像人一样“思考”？

目前，深度神经网络是较为热的研究方向，其能够实现高逼真的人类行为，甚至表现出超过人类的行为，也被誉为最类似人脑的人工智能技术，在自然语言处理、计算机视觉和围棋上的效果最为突出。

第一章：绪论





如何实现人工智能(AI)？

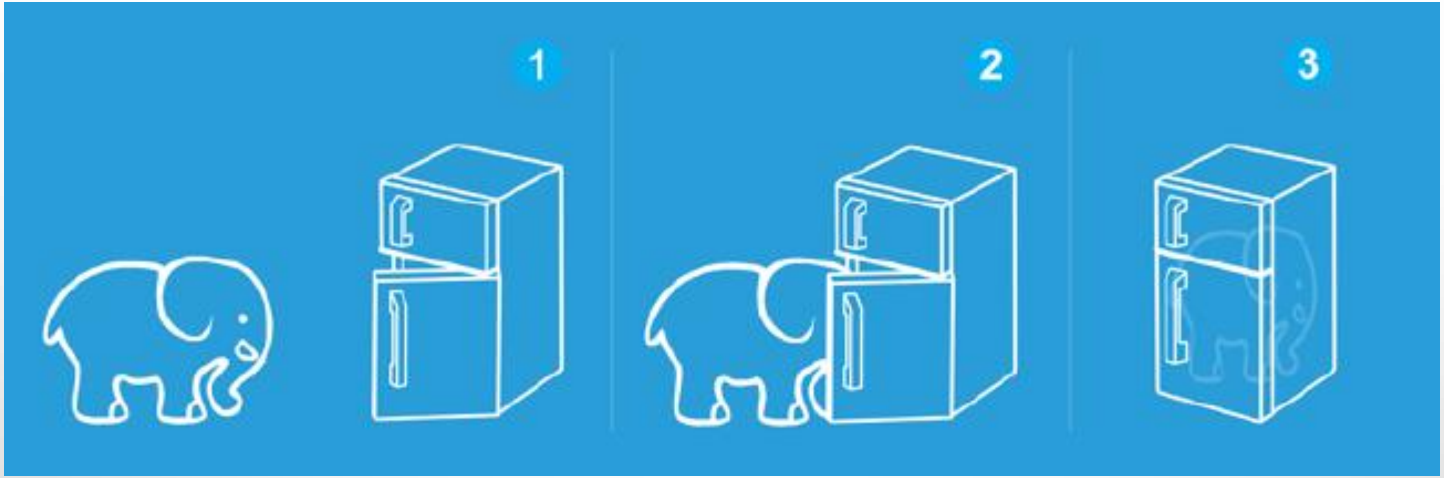
◆ 如何让机器像人一样“思考”？

最近，类脑智能的研究逐渐展开，脑科学的研究人员开始和人工智能领域的计算机和数学研究相结合。以构造符合人脑思维的人工智能新理论和新方法。

如何实现人工智能(AI)?

◆ 如何让机器像人一样“思考”？

深度学习的简单框架：大象和冰箱的故事。

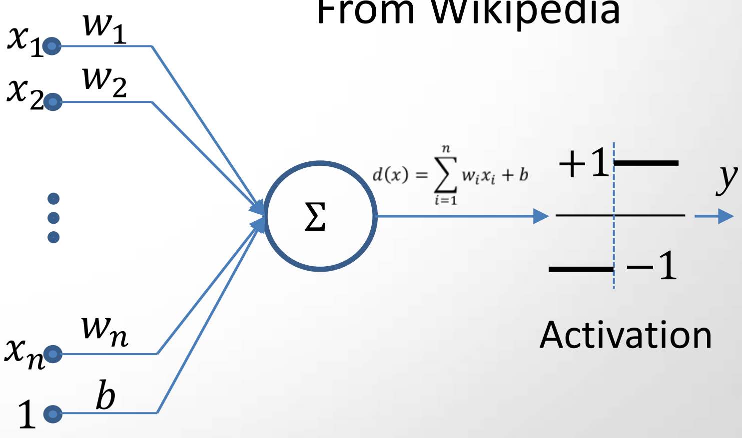


人工智能(AI)起源(60年前，1950):

Rosenblatt 提出“感知器”概念，并在刚刚萌芽的AI界产生了不小的争论。

关于“认知”，由于“感知器”的提出，纽约时报称之为“可使电子计算机能够散步、说话、看、写、自我复制、产生意识的胚胎。”

From Wikipedia

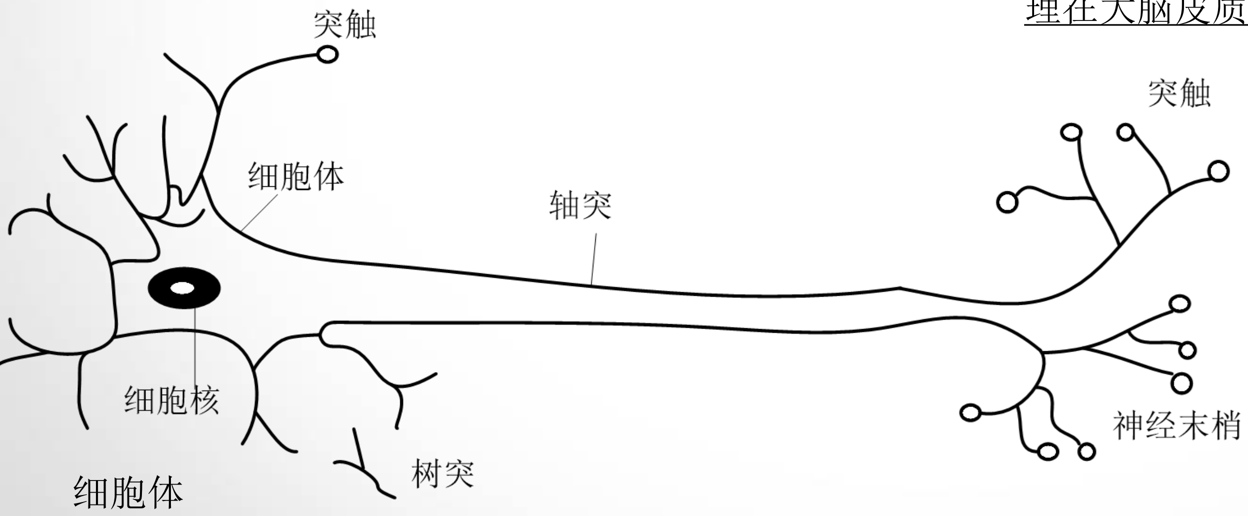


第一章：绪论

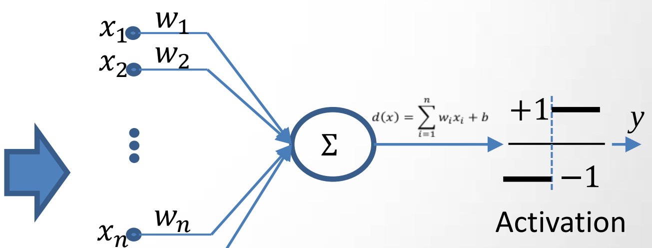
每个神经元由一个细胞体组成，其包含一个细胞核。从细胞体分支扩展出许多为树突的神经纤维和一根长的轴突。一个神经元与10到10万个其他神经元相连，连接处称为突触。信号通过复杂的化学反应从神经元传播到神经元。研究发现，大多数的信息处理在大脑皮质，即大脑外层进行。

人工智能(AI)起源(60年前，1950)：

生物神经元（神经细胞）结构：



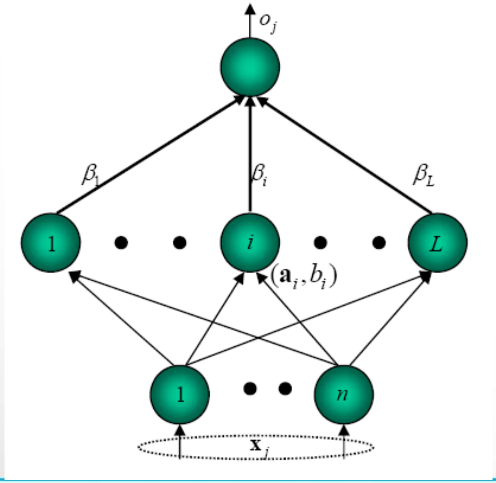
生物神经元结构

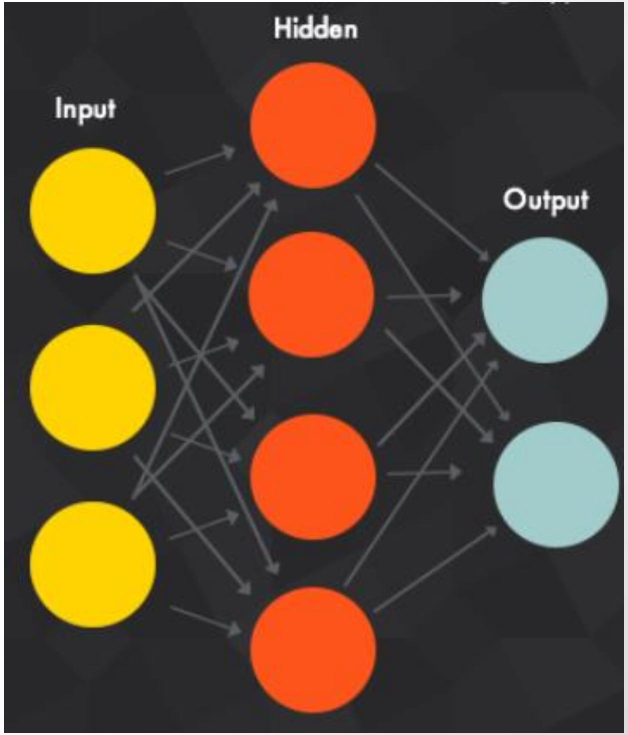


人工神经元结构

人工智能(AI)冬天(1970s)：

人工智能冬天也是神经网络的冬天，Minsky说“两层的神经网络并不能解决简单的XOR问题，导致AI winter的来临”

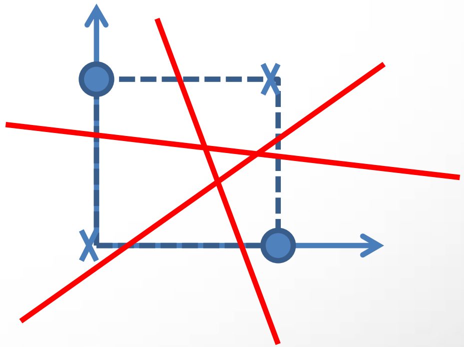




人工智能(AI)冬天(1970s)：

人工智能冬天也是神经网络的冬天，Minsky说“两层的神经网络并不能解决简单的XOR问题，导致AI winter的来临”

XOR(异或)问题：
0,0： 0
0,1： 1
1,0： 1
1,1： 0



无法找到一条直线将+1和-1分开

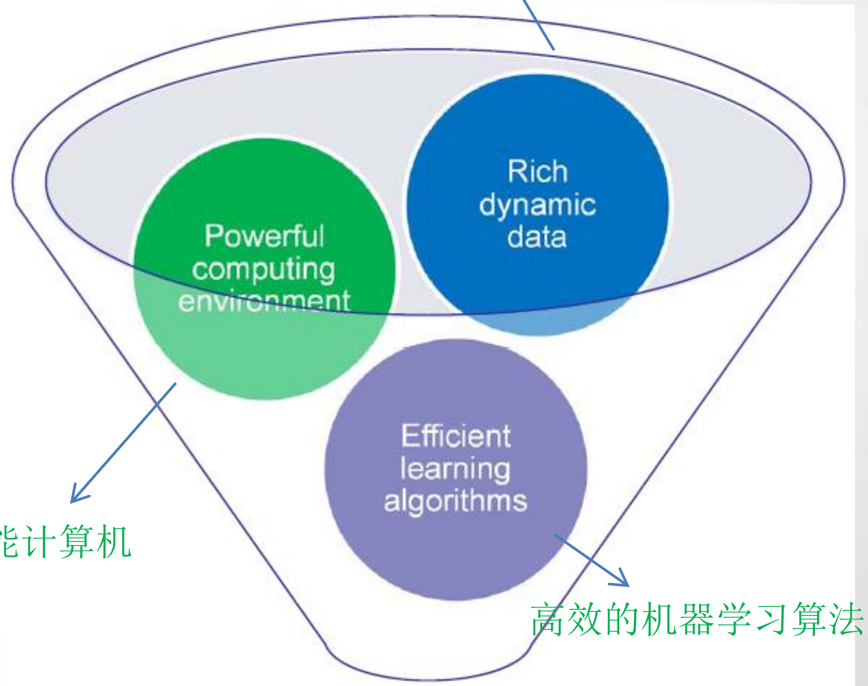
第一章：绪论

人工智能(AI)基础：

<u>数学</u>、<u>计算科学</u>、控制、神经科学、心理学、语言学、哲学

过去60年，人工智能发展重点在算法和模型上；

而今天，人工智能发展处于巅峰期，取决于**三点**：

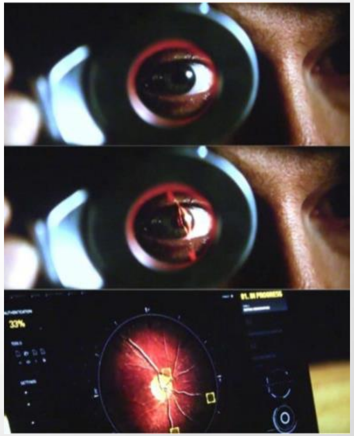


人工智能(AI)最新发展：

机器视、听、触、嗅、感觉及思维方式的模拟：指纹识别、人脸识别、虹膜识别、掌纹识别、图像/视频识别、语音识别、机器翻译、自然语言处理、搜索/检索、博弈（围棋、象棋）、无人驾驶等

人脸识别（谍影重重5）

步态分析与识别（碟中谍5）



视网膜识别（碟中谍5）

第一章：绪论

利用智能信息处理技术从大规模人脸图像数据中获取特征、知识表达、识别模型等

人工智能(AI)最新发展：

自然语言处理(NLP)：计算机科学和人工智能领域的重要方向之一。其研究人与计算机如何使用自然语言进行通信的理论和方法。主要包括两个方面：自然语言理解、自然语言生成。

难点：自然语言文本和对话存在不同的歧义性或多义性。

存在的技术问题：1. 局限于分析一个孤立的句子，上下文关系和谈话场景对该句子的影响缺乏系统研究。2. 人类理解一个句子不仅仅是凭语法，还运用大量有关知识，比如生活常识和专门技能，而这些知识无法存储在计算机中。

人工智能(AI)最新发展：

图像视频理解：人工智能和计算机视觉领域的重要方向。其研究如何利用计算机对图像和视频的内容、场景、语义的理解、标注、分类和识别。目前在视频监控、图像检索、图-文转换等领域应用较广。其实现过程分成4个层：

数据层：图像数据的获取（传感器）、压缩和传输（通信）；

描述层：特征提取，图像信号(像素)的形式化；

认知层：图像理解，包括机器学习和推理；

应用层：对特定任务（分类、识别、检测），设计模式识别和机器学习算法。

图像和文本之间的转换是一项具有趣味性和挑战性的研究

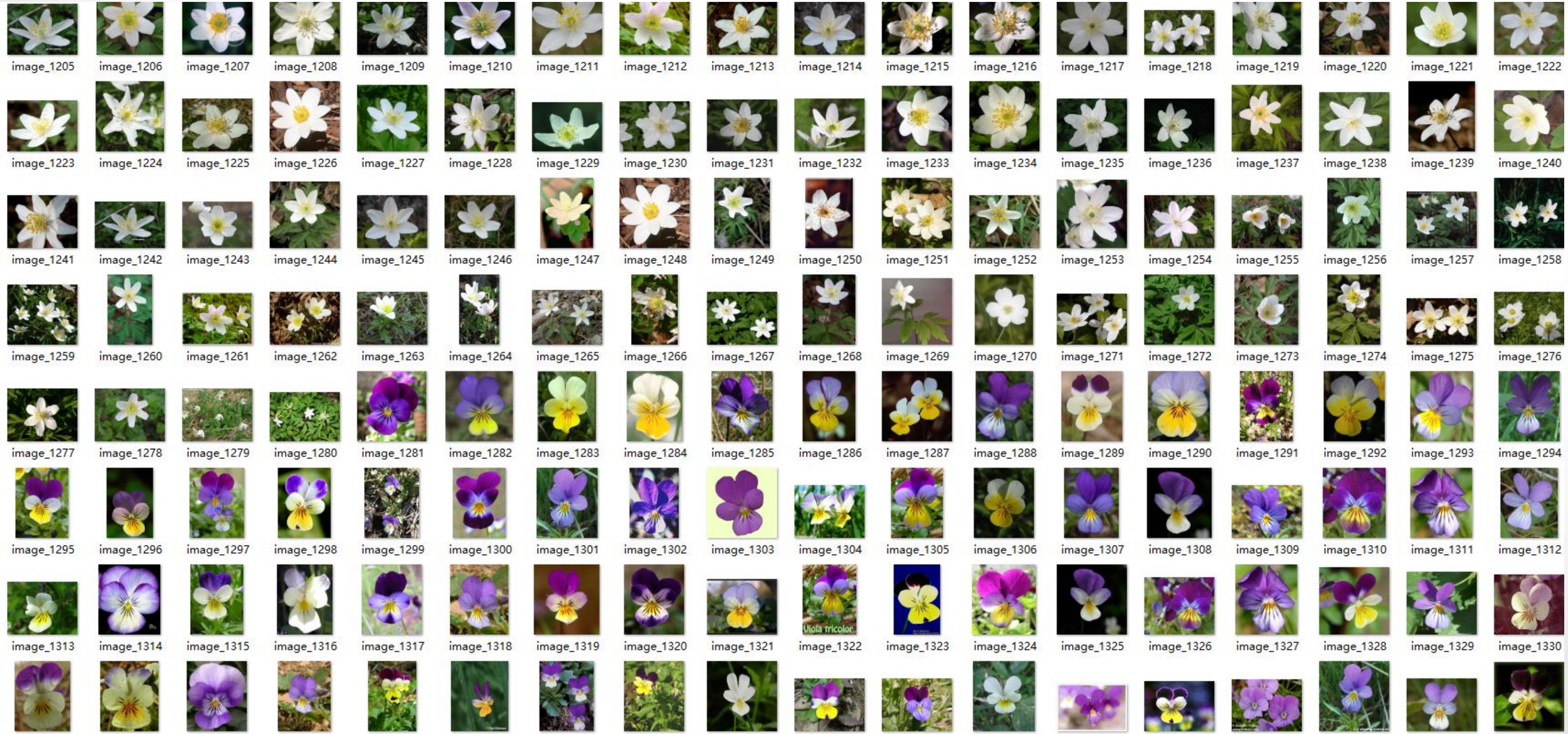


image_1205 image_1206 image_1207 image_1208 image_1209 image_1210 image_1211 image_1212 image_1213 image_1214 image_1215 image_1216 image_1217 image_1218 image_1219 image_1220 image_1221 image_1222
image_1223 image_1224 image_1225 image_1226 image_1227 image_1228 image_1229 image_1230 image_1231 image_1232 image_1233 image_1234 image_1235 image_1236 image_1237 image_1238 image_1239 image_1240
image_1241 image_1242 image_1243 image_1244 image_1245 image_1246 image_1247 image_1248 image_1249 image_1250 image_1251 image_1252 image_1253 image_1254 image_1255 image_1256 image_1257 image_1258
image_1259 image_1260 image_1261 image_1262 image_1263 image_1264 image_1265 image_1266 image_1267 image_1268 image_1269 image_1270 image_1271 image_1272 image_1273 image_1274 image_1275 image_1276
image_1277 image_1278 image_1279 image_1280 image_1281 image_1282 image_1283 image_1284 image_1285 image_1286 image_1287 image_1288 image_1289 image_1290 image_1291 image_1292 image_1293 image_1294
image_1295 image_1296 image_1297 image_1298 image_1299 image_1300 image_1301 image_1302 image_1303 image_1304 image_1305 image_1306 image_1307 image_1308 image_1309 image_1310 image_1311 image_1312
Viola tricolor
image_1313 image_1314 image_1315 image_1316 image_1317 image_1318 image_1319 image_1320 image_1321 image_1322 image_1323 image_1324 image_1325 image_1326 image_1327 image_1328 image_1329 image_1330

?

PSO

?

人工智能(AI)最新发展：

博弈：

IBM公司的深蓝(deep blue)第一个在象棋比赛中击败世界冠军(加里. 卡斯帕罗夫, 1997)。

深蓝是一台超级计算机，而非智能体（Agent）；

Google公司DeepMind团队开发的AlphaGo在围棋比赛中采用最新的深度学习技术战胜世界冠军(李世石, 2016)。

AlphaGo是一个人工智能程序，是当前人工智能的标志性事件。

人工智能(AI)争议：

Von Neumann： 计算机不会有智能；

Alan Turing: 计算机是能达到人的智力水平的；

McCarthy: 人工智能的主要问题是难解的；

Minsky: 人工智能是最难的科学之一，是思维的社会无统一的知识表示和理论基础

反对派核心观点：计算机只能解决形式化的问题，而客观世界是非形式化的，变化无穷的。

智能信息处理 课程：

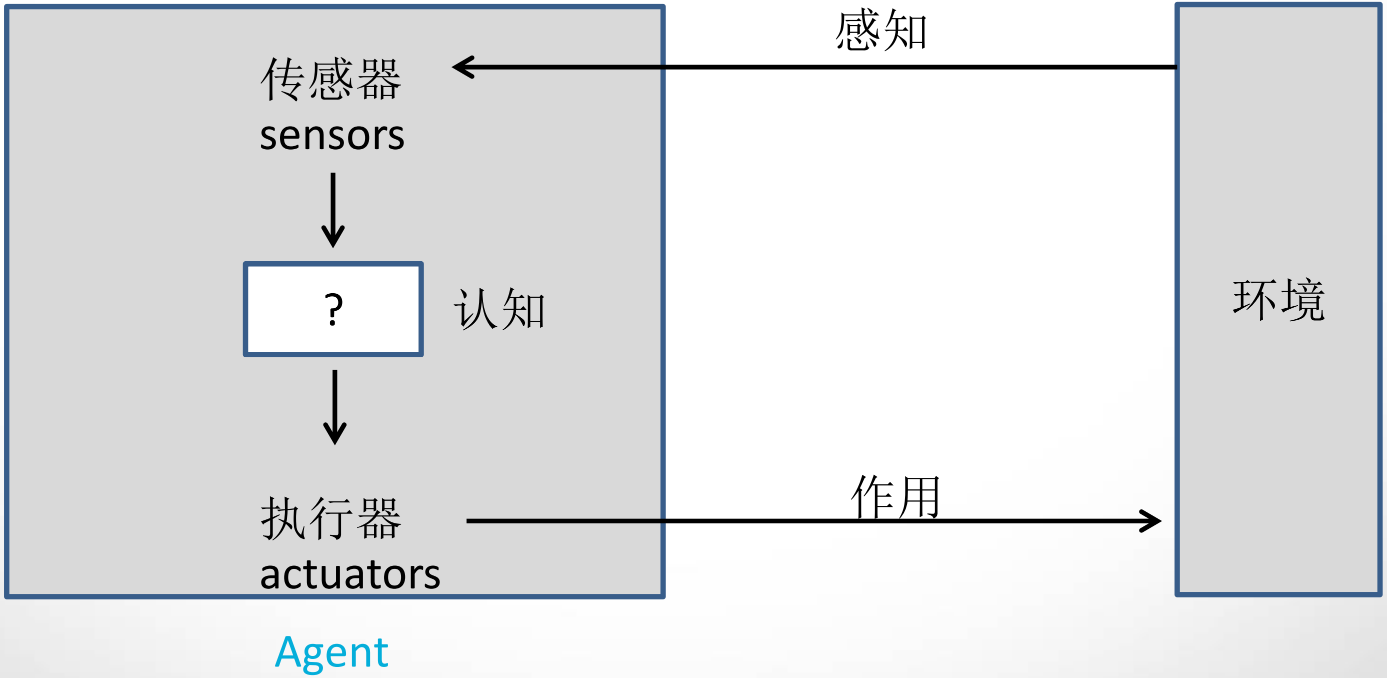
人工智能涉及多个领域，本课程主要介绍智能信息处理的新方法和新理论，主要包括智能体Agent、机器学习、演化进化算法、神经网络、深度学习、人脸识别、计算机视觉应用。

第二章： 智能体AGENT

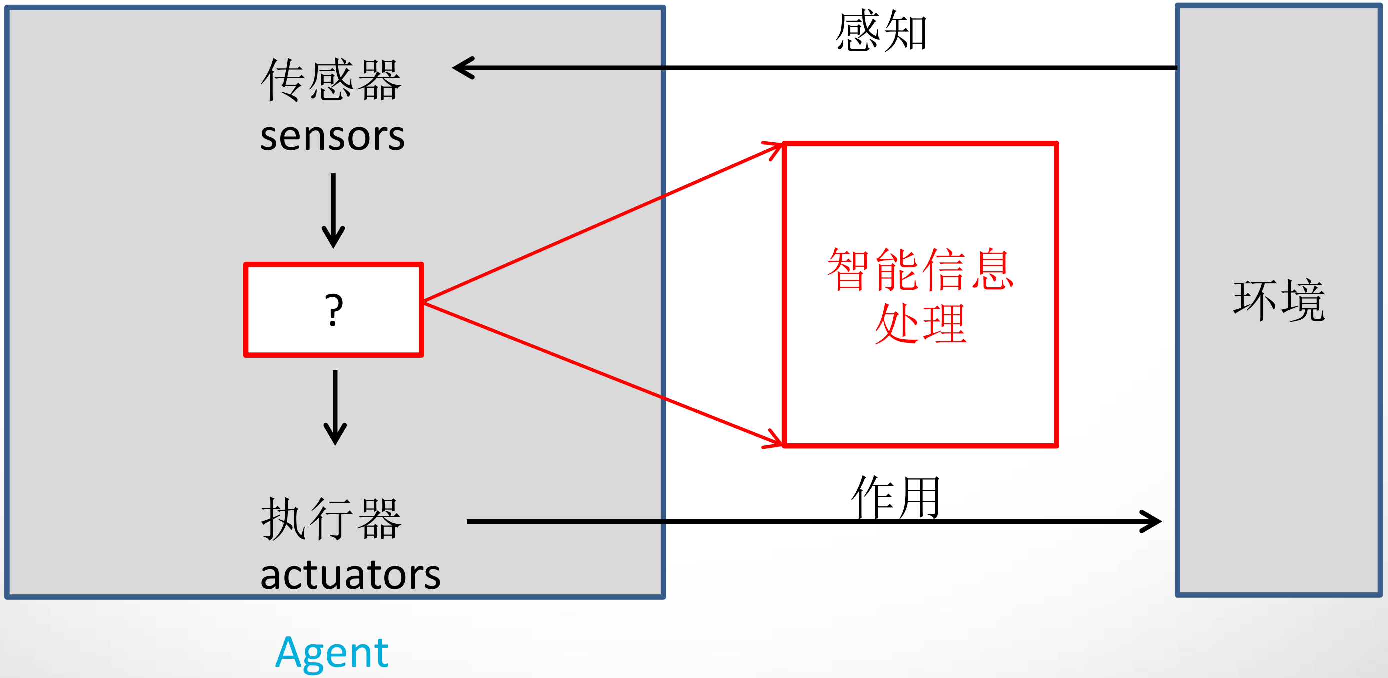
Agent：

- 在人工智能领域，Agent有多种翻译，如“智能体”、“智能代理”，“主体”等。它可以看做是一个自动执行的实体，通过传感器感知环境，通过执行器作用于环境。
- 多Agent系统即是研究多个智能Agent的协调工作，实现问题求解。
- 构建Agent的任务，就是设计Agent程序，实现从感知到动作的映射。

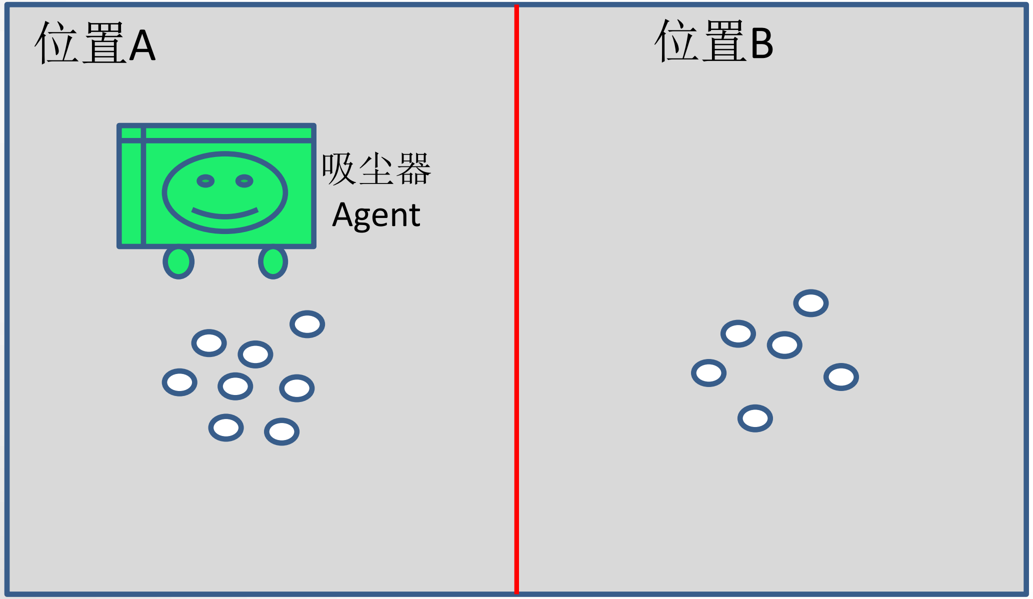
智能体通过传感器和执行器与环境进行交互的过程：



智能体通过传感器和执行器与环境进行交互的过程：



智能体通过传感器和执行器与环境进行交互的过程：



吸尘器Agent系统

感知	Action
A, clean	right
A, dirty	Suck
B, clean	Left
B, dirty	Suck

简单的Agent函数表格

如何定义Agent函数表格才是最好的？是什么决定了一个Agent是好还是坏？智能还是愚笨？

- 好的行为：理性概念

理性Agent：根据Agent行动的后果，当把Agent置于一个环境中后，它针对收到的感知信息生成一个行动序列，该行动序列会对环境产生一系列的状态变化，如果该系列是渴望的，那么该Agent性能良好。

注意：性能度量是根据环境状态变化进行评估，而不是Agent的状态。

设计Agent要考虑的因素

- 理性Agent

对于吸尘器，我们通过其1小时内的吸尘总量来评估Agent的性能，那么对于理性Agent来说，一边吸尘，一边把灰尘撒到地上，再吸尘，最终可以保证吸尘量的最大化，但这样好吗？

更合适的性能度量，应当奖励保持干净地面的Agent，根据环境的状态来设计性能度量，而不是根据Agent表现出的行为。

设计Agent要考虑的因素

- 理性Agent应具备的四点：
- 定义成功标准的性能度量。 假设性能度量在每个时间步对每块清洁的方格奖励1分。
- Agent对环境的先验知识。环境的“地形”作为先验是已知的。
- Agent可以完成的行动。左、右、吸尘。
- Agent的感知序列。Agent可以正确感知位置及所在方格是否有灰尘。

设计Agent要考虑的因素

- 非理性Agent：

同样的Agent在不同环境下会变的非理性。一旦所有灰尘全部清洁，该吸尘器会毫无必要的来回移动，如果性能度量包含对左右移动惩罚1分，该Agent的性能评价将变的十分糟糕。

一个好的Agent需能够确保所有地方清洁干净后，不在有任何行动。如果再次弄脏，应该不定期检查，重新清洁。如果环境“地形”未知，可能还会去探查其他区域。

设计Agent要考虑的因素

- 理性与全知(完美)Agent：

理性是使期望的性能最大化，而完美是使实际的性能最大化。理性的选择只依赖于截止当时为止的感知序列，还要确保没有因漫不经心而让Agent进行愚蠢的活动。

根据信息不全的感知序列，采取行动是不理性的，利用全面的感知信息，有助于期望性能最大化。比如智能体过马路。

设计Agent要考虑的因素

- Agent的自主性：

我们定义理性的Agent不仅要收集信息，还应当从感知信息中学习。如果系统设计时，只依赖设计人员的先验知识，Agent缺乏自主性。比如一个吸尘器Agent能够预见灰尘出现的时间和地点，显然会更加“智能”。

Agent应当具备“进化”能力。只有与“学习”相结合，才能设计出适应于不同环境的理性Agent。

“学习”是人工智能必备的能力。

如何设计Agent?

■ Agent的任务环境定义：

通过理性Agent，我们知道：性能度量(**P**erformance)、环境(**E**nvironment)、执行器(**A**ctuators)、传感器(**S**ensors)。把这四个因素归结在一起，就是任务环境，也称为PEAS描述。这是设计Agent的第一步。

考虑自动驾驶出租车Agent:

出租车任务环境的PEAS描述，如下：

出租车任务环境的PEAS描述

Agent类型	Performance 性能度量	Environment 环境	Actuators 执行器	Sensors 传感器
自动驾驶出租车	安全、快速、舒适、符合交通法规、费用低	马路、其他车辆、行人、顾客、道路施工、障碍物	油门、刹车、发动机、转向控制、显示器、语音合成器与顾客交流	摄像头、红外设备、声呐、加速计、速度表、GPS导航、麦克风

医疗诊断系统、卫星图像分析系统等Agent

如何设计Agent?

■ Agent任务环境的性质：

✓ 完全可观察与部分可观察；

取决于传感器，如果传感器能获取环境的完整状态，那么任务环境是完全可观察的；如果能够检测与行动决策相关的信息，那么是有效、完全可观察的。

如果因噪声、或传感器丢失状态数据，将导致任务环境为部分可观察的。如果没有传感器，则是无法观察的。

✓ 单Agent和多Agent

多Agents是以某种竞争或合作的形式存在。比如，国际象棋是双Agent环境，Agent A和B是以竞争的形式，比如A试图最大化自己的性能度量，而最小化对手的性能度量。两个出租车司机，则是以合作的形式避免碰撞，从而实现各自的性能度量最大化。当然，也存在部分竞争，比如两个Agent，但只有一个停车位。

如何设计Agent？

- Agent任务环境的性质：
- ✓ 确定的与随机的：环境的下一个状态完全取决于当前状态和Agent执行的动作，则是确定的。否则，是随机的，比如突如其来的飞机舱门。
- ✓ 静态的与动态的：环境在Agent计算时会发生变化，比如路况是动态的。
- ✓ 已知的与未知的：比如一个Agent到了一个陌生的城市环境，自然不熟悉路况，到了一个陌生国家，对交通法规也是未知的。

Agent的结构？

- Agent内部工作方式：

AI的任务是设计Agent程序，实现感知信息映射到行动的Agent函数 (认知过程)。程序的运行环境，即具有某个物理传感器和执行器的计算装置，也称为体系结构（载体或平台）。

Agent=体系结构+程序

体系结构可能是一台计算机，或者具有车载计算机、摄像头和其他传感器的自动驾驶汽车。

一般而言，体系架构为**程序**提供来自传感器的感知信息，并运行程序，并把程序计算结果传送给执行器。

Agent程序

■ Agent程序的几种：

- 简单反射Agent：基于当前的感知信息，选择行动，不关注历史信息。比如简单的吸尘器，只建立在当前位置和是否包含灰尘。触发Agent程序的规则，称为“条件--行为规则”。
- 基于模型的反射Agent：根据感知历史，维持智能体内部状态。
- 基于目标的Agent：需要提供目标信息。比如出租车可以选择任何方向（左转、右转、直行），但如果提供目标信息后，可以做出正确的选择。
- 基于效用的Agent：仅靠目标信息，不能够做出最正确的选择，比如要考虑更安全、更快速、更可靠、更便宜的路线。

第三章： 智能信息处理中的数学基础

人工智能、智能信息处理、机器学习、模式识别的数学基础主要包括以下几个部分：

- 线性代数
- 矩阵论
- 优化理论
- 概率论
- 信息论
- 计算复杂度理论。

学好这门课，这些数学基础是必不可少的。

向量

在线性代数中，标量(scalar)是一个实数，而向量(vector)是指n个实数组成的有序数组，称为n维向量。如果没有特别说明，一个n维向量一般表示为一个列向量，即大小为nx1的列向量。向量的表示形式一般采用粗体、小写。如**a**. 如果写成*a*,则表示一个标量。

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \in \Re^n$$ （n维欧几里得空间）

常见向量

全1向量：指所有元素为1的向量，通常用$\mathbf{1}_n$表示,*n*表示向量的**维度**。$\mathbf{1}_K = [1, \cdots, 1]^{\top}_{K\times 1}$ 表示*K*维的全1向量。

向量的模和范数

向量**a**的模表示为‖**a**‖,计算方法为

$$\|\mathbf{a}\| = \sqrt{\sum_{i=1}^{n} a_i^2}$$

在线性代数中，范数(norm)是一个表示“长度”概念的函数，为向量空间内所有向量赋予非零的正长度或大小。对于一个*n*维的向量**x**，其常见的范数有

L_1 范数：

$$|\mathbf{x}|_1 = \sum_{i=1}^{n} |x_i|.$$

L_2 范数：

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} = \sqrt{\mathbf{x}^T\mathbf{x}}.$$

欧几里得范数，向量的长度，如果向量**x**满足‖**x**‖ = 1, 则称向量是**归一化**的

向量的内积

两个具有相同维数n的向量**x**和**y**的内积记为$\mathbf{x}^{\mathrm{T}}\mathbf{y}$,是一个**标量**。

$$\mathbf{x}^{\mathrm{T}}\mathbf{y} = \sum_{i=1}^{n} x_i y_i$$

内积也称作标量积或点积，在泛函里面表示为**<x,y>**

向量的夹角

两个n维向量的夹角定义

$$\cos\boldsymbol{\theta} = \frac{\mathbf{x}^{\mathrm{T}}\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$$

涵义： 向量内积是两个向量共线性的度量，说明两个向量之间的相似性。
若**x**和**y**正交，则$\mathbf{x}^{\mathrm{T}}\mathbf{y} = 0$。由$\cos\boldsymbol{\theta} \le \mathbf{1}$，$\mathbf{x}^{\mathrm{T}}\mathbf{y} \le \|\mathbf{x}\|\|\mathbf{y}\|$ (柯西-施瓦茨不等式)

向量的外积

两个具有相同维数n的向量$\mathbf{x}$和$\mathbf{y}$的外积记为$\mathbf{xy}$,是一个矩阵。

$$\mathbf{xy}=\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}(y_1\ y_2 \cdots y_3)$$

$$=\begin{pmatrix} x_1y_1 & \cdots & x_1y_n \\ \vdots & \ddots & \vdots \\ x_ny_1 & \cdots & x_ny_n \end{pmatrix}$$

向量外积也称作<u>矩阵积或二元积</u>

矩阵

一个大小为mxn的矩阵(matrix)是一个由m行n列元素排列成的矩形阵列。矩阵里的元素可以是数字、符号或数学式。在我们的课程里，矩阵一般默认为数字矩阵。

一个n维向量，可以看作是一个nx1的矩阵。

一个mxn的矩阵**M**，表达为$\mathbf{M} \in \Re^{m\times n}$ (欧几里得空间)

矩阵的数学符号通常有粗体、大写表示。

矩阵的基本运算

如果 A 和 B 都为 $m \times n$ 的矩阵，则 A 和 B 的加减为

$$(A+B)_{ij} = A_{ij} + B_{ij},$$

$$(A-B)_{ij} = A_{ij} - B_{ij}.$$

A 和 B 的点乘 $A \odot B \in \mathbb{R}^{m \times n}$ 为

$$(A \odot B)_{ij} = A_{ij} B_{ij}.$$

一个标量 c 与矩阵 A 乘积为

$$(cA)_{ij} = cA_{ij}.$$

若 A 是 $m \times p$ 矩阵和 B 是 $p \times n$ 矩阵，则乘积 AB 是一个 $m \times n$ 的矩阵

$$(\mathbf{AB})_{ij} = \sum_{k=1}^{p} A_{ik} B_{kj}$$

矩阵的基本运算

$m \times n$ 矩阵 A 的**转置**（Transposition）是一个 $n \times m$ 的矩阵，记为 $A^\top$，$A^\top$ 第 i 行第 j 列的元素是原矩阵 A 第 j 行第 i 列的元素，

$$(A^\top)_{ij} = A_{ji}.$$

如果A是对称矩阵，则$A=A^T$

矩阵的**向量化**是将矩阵表示为一个列向量。这里，vec 是向量化算子。设 $A = [a_{ij}]_{m \times n}$，则

$$\mathrm{vec}(A) = [a_{11}, a_{21}, \cdots, a_{m1}, a_{12}, a_{22}, \cdots, a_{m2}, \cdots, a_{1n}, a_{2n}, \cdots, a_{mn}]^\top.$$

矩阵的基本运算

矩阵的迹

矩阵**A**的迹，数学表示为Tr(**A**)或Trace(**A**),即对角线元素的和。

$$\mathrm{Tr}(AA^{T})=\mathrm{Tr}(A^{T}A)=\|A\|_F^2=\sum_{i,j}A_{i,j}^2$$

矩阵的逆 A^{-1}

矩阵的转置A^{T}

矩阵的基本运算

矩阵与向量的乘积

$$\begin{bmatrix} m_{11} & \cdots & m_{1d} \\ \vdots & \ddots & \vdots \\ m_{n1} & \cdots & m_{nd} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

可以写另一种表达方式

$$\mathbf{Mx} = \mathbf{y}$$

其中，

$$y_i = \sum_{j=1}^{d} m_{ij} x_j \text{ (线性组合)}$$

常见矩阵

对称矩阵指其转置等于自己的矩阵，即满足 $A = A^\top$。

对角矩阵（Diagonal Matrix）是一个主对角线之外的元素皆为 0 的矩阵。对角线上的元素可以为 0 或其他值。一个 $n \times n$ 的对角矩阵矩阵 A 满足：

$$A_{ij} = 0 \text{ if } i \neq j \qquad \forall i, j \in \{1, \cdots, n\}$$

对角矩阵 A 也可以记为 $\mathrm{diag}(\mathrm{a})$，$\mathrm{a}$ 为一个 n 维向量，并满足

$$A_{ii} = a_i.$$

常见矩阵

$n \times n$ 的对角矩阵矩阵 $A = \mathrm{diag}(\mathbf{a})$ 和 n 维向量 $\mathbf{b}$ 的乘积为一个 n 维向量

$$A\mathbf{b} = \mathrm{diag}(\mathbf{a})\mathbf{b} = \mathbf{a} \odot \mathbf{b},$$

其中 $\odot$ 表示点乘，即 $(\mathbf{a} \odot \mathbf{b})_i = a_i b_i$。

单位矩阵是一种特殊的的对角矩阵，其主对角线元素为 1，其余元素为 0。n 阶单位矩阵 I_n，是一个 $n \times n$ 的方形矩阵。可以记为 $I_n = \mathrm{diag}(1, 1, ..., 1)$。一个矩阵和单位矩阵的乘积等于其本身。

导数

对于定义域和值域都是实数域的函数y=f(x),若f(x)在点x_0的某个邻域Δx内，极限 $f'(x_0)=\lim_{\Delta x\to 0}\frac{f(x_0+\Delta x)-f(x_0)}{\Delta x}$

存在，则称函数y=f(x)在点x_0处可导，其导数为$f'(x_0)$

函数 $f(x)$ 的**导数** $f'(x)$ 也可记作 $\nabla_x f(x)$，$\frac{\partial f(x)}{\partial x}$ 或 $\frac{\partial}{\partial x}f(x)$。

向量和矩阵的导数

- 对于一个p维的向量$\mathbf{x} \in \Re^p$,函数 $y = f(\mathbf{x}) = f(x_1, x_2, \cdots, x_p)$是一个标量,则y关于**x**的导数为

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_p} \end{bmatrix} \in \Re^p$$

→ 相当于梯度，是由p个偏导数组成的矢量

- 如果$\mathbf{y} = f(\mathbf{x}) = f(x_1, x_2, \cdots, x_p) \in \Re^q$是一个向量，

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_q(\mathbf{x})}{\partial x_1} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_1(\mathbf{x})}{\partial x_p} & \cdots & \frac{\partial f_q(\mathbf{x})}{\partial x_p} \end{bmatrix} \in \Re^{p\times q}$$

导数法则

加（减）法则

$\mathbf{y} = f(\mathbf{x})$,$\mathbf{z} = g(\mathbf{x})$ 则

$$\frac{\partial(\mathbf{y}+\mathbf{z})}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}} + \frac{\partial \mathbf{z}}{\partial \mathbf{x}}$$

乘法法则

（1）若 $\mathbf{x} \in \mathbb{R}^p$，$\mathbf{y} = f(\mathbf{x}) \in \mathbb{R}^q$，$\mathbf{z} = g(\mathbf{x}) \in \mathbb{R}^q$，则

$$\frac{\partial \mathbf{y}^\top \mathbf{z}}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}}\mathbf{z} + \frac{\partial \mathbf{z}}{\partial \mathbf{x}}\mathbf{y}$$

导数法则

（2）若 $\mathbf{x} \in \mathbb{R}^p$，$\mathbf{y} = f(\mathbf{x}) \in \mathbb{R}^s$，$\mathbf{z} = g(\mathbf{x}) \in \mathbb{R}^t$，$A \in \mathbb{R}^{s\times t}$ 和 $\mathbf{x}$ 无关，则

$$\frac{\partial \mathbf{y}^\top A\mathbf{z}}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}}A\mathbf{z} + \frac{\partial \mathbf{z}}{\partial \mathbf{x}}A^\top \mathbf{y}$$

（3）若 $\mathbf{x} \in \mathbb{R}^p$，$y = f(\mathbf{x}) \in \mathbb{R}$，$\mathbf{z} = g(\mathbf{x}) \in \mathbb{R}^p$，则

$$\frac{\partial y\mathbf{z}}{\partial \mathbf{x}} = y\frac{\partial \mathbf{z}}{\partial \mathbf{x}} + \frac{\partial y}{\partial \mathbf{x}}\mathbf{z}^\top$$

导数法则

链式法则

（1）若 $\mathbf{x} \in \mathbb{R}^p$，$\mathbf{y} = g(\mathbf{x}) \in \mathbb{R}^s$，$\mathbf{z} = f(\mathbf{y}) \in \mathbb{R}^t$，则

$$\frac{\partial \mathbf{z}}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{z}}{\partial \mathbf{y}}$$

（2）若 $X \in \mathbb{R}^{p \times q}$ 为矩阵，$Y = g(X) \in \mathbb{R}^{s \times t}$，$z = f(Y) \in \mathbb{R}$，

$$\frac{\partial z}{\partial X_{ij}} = \mathbf{tr}\left((\frac{\partial z}{\partial Y})^\top \frac{\partial Y}{\partial X_{ij}} \right)$$

（3）若 $X \in \mathbb{R}^{p \times q}$ 为矩阵，$\mathbf{y} = g(X) \in \mathbb{R}^s$，$z = f(\mathbf{y}) \in \mathbb{R}$，则

$$\frac{\partial z}{\partial X_{ij}} = (\frac{\partial z}{\partial \mathbf{y}})^\top \frac{\partial \mathbf{y}}{\partial X_{ij}}$$

导数法则

如果矩阵$\mathbf{M} \in \Re^{n\times d}$的每一个元素都是关于标量参数$\theta$的函数，那么矩阵$\mathbf{M}$对参数$\theta$的导数为

$$\frac{\partial \mathbf{M}}{\partial \theta} = \begin{pmatrix} \frac{\partial m_{11}}{\partial \theta} & \cdots & \frac{\partial m_{1d}}{\partial \theta} \\ \vdots & \ddots & \vdots \\ \frac{\partial m_{n1}}{\partial \theta} & \cdots & \frac{\partial m_{nd}}{\partial \theta} \end{pmatrix}$$

导数法则

按位运算的向量函数及其导数:

定义$\mathbf{x} = [x_1, \cdots, x_d]^{\mathrm{T}} \in \Re^d$, $\mathbf{z} = [z_1, \cdots, z_d]^{\mathrm{T}} \in \Re^d$

$$\mathbf{z} = f(\mathbf{x})$$

其中$f(\mathbf{x})$是按位运算的，即$z_i = f(x_i)$

那么$f(\mathbf{x})$对$\mathbf{x}$的导数为

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial f(x_1)}{\partial x_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial f(x_d)}{\partial x_d} \end{pmatrix}$$

第四章： 线性建模

线性建模

在智能信息处理中，一个重要且普遍的问题是推断属性变量（自变量）与响应变量（因变量）之间的函数关系，使得给定任何一个属性集合，可以通过该函数关系，预测其响应。

高数中的自变量x与因变量y，有以下函数关系

$$y=h(x)$$

线性建模的目的：

当给定一组（**x,y**）集合时，如何估计函数**h(.)**的表达式？

线性建模

例1：建立一个能够执行疾病诊断的模型h(.)。

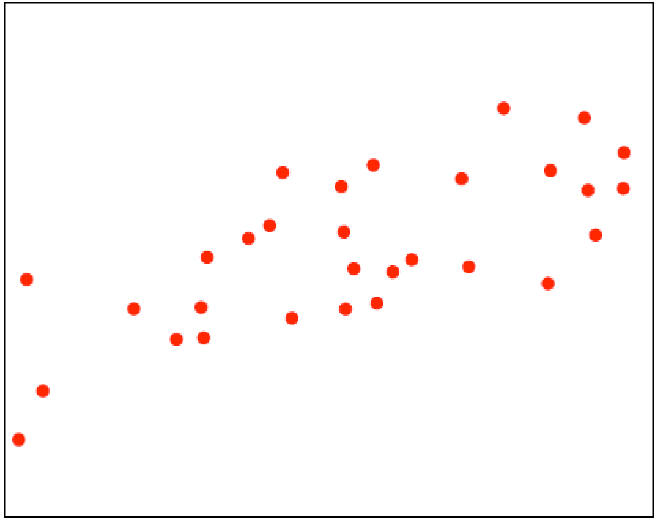
已知条件：已知疾病状态（健康、患病）的多个患者，以及对这些患者测量的属性（血压、心率、体重等）

例2：通过某用户在电影网页上的点击情况，预测该用户的喜好。

已知条件：已知被点击的电影种类（喜剧、动作、恐怖等），以及该用户对每个种类的点击次数。

模型假设

在建立回归模型时，模型假设很重要。



x	y
0.86	2.49
0.09	0.83
-0.85	-0.25
0.87	3.10
-0.44	0.87
-0.43	0.02
-1.10	-0.12
0.40	1.81
-0.96	-0.83
0.17	0.43

线性建模

线性建模是机器学习中最直接的学习问题：学习 属性与响应之间的线性关系。

例：建立苹果的重量预测模型：单属性问题（一维问题）

已知条件：已知N个苹果的重量$t_1, t_2, \cdots, t_N$，以及N个苹果的水平宽度$x_1, x_2, \cdots, x_N$ 。

任务：估计苹果的重量预测线性模型$y = h(x) = \omega_0 + \omega_1 x$

什么是好的模型？

为了选择某种方式下最好的ω_0和ω_1值，使得该直线能够尽可能与所有数据点接近的直线。

度量“好”的方法是采用“平方差”即预测的苹果重量y与实际重量t之间的平方差，定义为

$$(t_n - h(x_n;\omega_0,\omega_1))^2$$

该式子表示第n个苹果的预测误差，该值越小，说明模型$h(\cdot)$在x_n处的值越接近t_n。

这个度量有个专业称呼“平方损失函数”(squared loss function)，因此，第n个苹果（样本）的损失可以定义为

$$L(t_n, h(x_n;\omega_0,\omega_1)) = (t_n - h(x_n;\omega_0,\omega_1))^2$$

什么是好的模型？

对于全部的**N**个苹果，我们希望所有苹果的损失最小，因此，有平均损失，

$$L=\frac{1}{N}\sum_{n=1}^{N}L(t_n,h(x_n;\omega_0,\omega_1))=\frac{1}{N}\sum_{n=1}^{N}(t_n-h(x_n;\omega_0,\omega_1))^2$$

我们建模的目的是求得最佳的ω_0,ω_1,使得平均损失L达到最小。

上面的问题可以写成专业化的机器学习模型：

$$\min_{\omega_0,\omega_1} L=\frac{1}{N}\sum_{n=1}^{N}(t_n-h(x_n;\omega_0,\omega_1))^2$$

或者$\{\omega_0,\omega_1\}=\underset{\omega_0,\omega_1}{\arg\min}\,L=\frac{1}{N}\sum_{n=1}^{N}(t_n-h(x_n;\omega_0,\omega_1))^2$

求解过程

将损失函数展开

$$L = \frac{1}{N}\sum_{n=1}^{N}(t_n - h(x_n; \omega_0, \omega_1))^2$$

$$=$$

$$\vdots$$

$$= \frac{1}{N}\sum_{n=1}^{N}(\omega_1^2 x_n^2 + 2\omega_1 x_n(\omega_0 - t_n) + \omega_0^2 - 2\omega_0 t_n + t_n^2)$$

计算损失函数L对ω_0和ω_1的偏导数，并令导数为0，可得出损失最小值时的两个估计参数。

预测过程

目前，根据求解已获得了ω_0和ω_1的最佳估计值$\widehat{\omega}_0$和 $\widehat{\omega}_1$，从而可以写出函数表达式

$$t = \widehat{\omega}_0 + \widehat{\omega}_1 x = -0.8 + 0.19x$$

此时，对于一个新的样本（苹果），经过测量，发现其宽度为**12cm**，那该苹果的重量应该是多少？

$$t = -0.8 + 0.19 \times 12 = 1.48kg$$

模型的矩阵求解

继续考虑刚才的模型

$$t_n = h(x_n) = \omega_0 + \omega_1 x_n$$

可以写成

$$t_n = h(\mathbf{x}_n) = (\omega_0, \omega_1)\begin{pmatrix} 1 \\ x_n \end{pmatrix} = \mathbf{w}^{\mathrm{T}}\mathbf{x}_n$$

现将N个样本的属性向量$\mathbf{x}_n$合并成一个矩阵$\mathbf{X}$

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_N^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ & \vdots \\ 1 & x_N \end{pmatrix}$$

也将N个样本的目标t_n也合并成一个向量

$$\mathbf{t} = (t_1, t_2, \cdots, t_N)^{\mathrm{T}}$$

模型的矩阵求解

原损失函数

$$L=\frac{1}{N}\sum_{n=1}^{N}(t_n-h(x_n;\omega_0,\omega_1))^2$$

可以完全等价地转为向量和矩阵的函数，即

$$\begin{aligned}L&=\frac{1}{N}(\mathbf{t}-\mathbf{X}\mathbf{w})^{\mathrm{T}}(\mathbf{t}-\mathbf{X}\mathbf{w})\\&=\frac{1}{N}\left(\mathbf{t}^{\mathrm{T}}-(\mathbf{X}\mathbf{w})^{\mathrm{T}}\right)(\mathbf{t}-\mathbf{X}\mathbf{w})\\&=\frac{1}{N}\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w}-\frac{2}{N}\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{t}+\frac{1}{N}\mathbf{t}^{\mathrm{T}}\mathbf{t}\end{aligned}$$

此时，可以通过矩阵向量求导，可以直接获得**w**,而无需对ω_0,ω_1分别求偏导。

模型的矩阵求解

$$L = \frac{1}{N}\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w} - \frac{2}{N}\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{t} + \frac{1}{N}\mathbf{t}^{\mathrm{T}}\mathbf{t}$$

此时，可以通过矩阵向量求导，可以直接获得**w**,而无需对ω_0, ω_1分别求偏导。

因为$\frac{\partial L}{\partial \mathbf{w}} = \begin{bmatrix} \frac{\partial L}{\partial \omega_0} \\ \frac{\partial L}{\partial \omega_1} \end{bmatrix} = \frac{2}{N}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w} - \frac{2}{N}\mathbf{X}^{\mathrm{T}}\mathbf{t} = 0$

可以推出

$$\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w} = \mathbf{X}^{\mathrm{T}}\mathbf{t}$$

那么$\mathbf{w} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}$（完毕）

模型的矩阵求解

现在已经获得w的解析表达式

$$\mathbf{w} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t} = \begin{bmatrix} \omega_0 \\ \omega_1 \end{bmatrix}$$

例：给出3个样本点组成的集合

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 1 & 3 \\ 1 & 5 \end{bmatrix}, t = \begin{bmatrix} 4.8 \\ 11.3 \\ 17.2 \end{bmatrix}$$

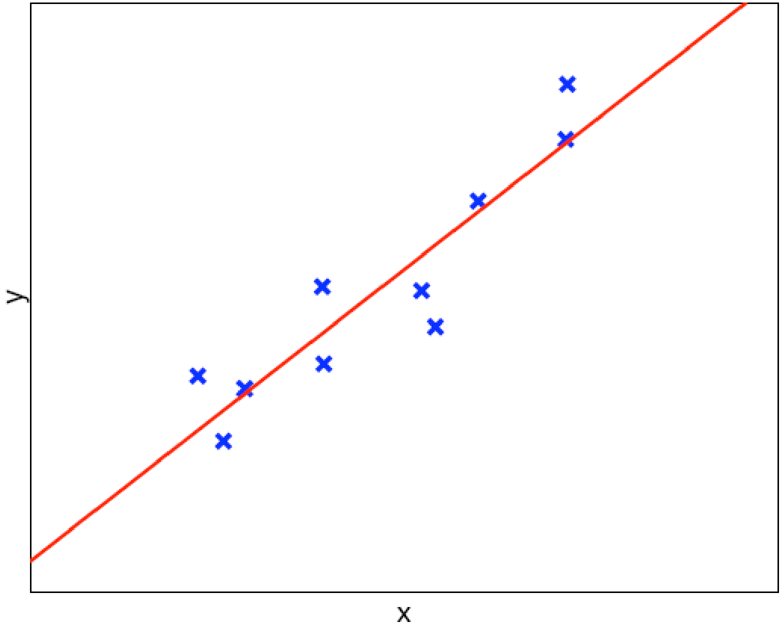
如何建立线性模型的表达式？

预测：给定一个属性$\mathbf{x}_{new} = [1\ x_{new}]^{\mathrm{T}}$的新向量, 其预测公式：

$$t_{new} = \omega_0 + \omega_1 x_{new} = [\omega_0\ \ \omega_1][1\ x_{new}]^{\mathrm{T}} = \mathbf{w}^{\mathrm{T}}\mathbf{x}_{new}$$

Example: Data and best linear hypothesis

$$y = 1.60x + 1.05$$



线性建模-多维问题

多维问题是机器学习、实际应用中的普遍问题，即有多个 属性的情况。

例：建立一个能够执行疾病诊断的线性模型：多属性问题（多维）

已知条件：已知疾病状态（健康、患病）的N个患者，以及每个患者的血压值x_1、心率x_2 、体重x_3等，通过多个方面的因素，来预测患者的身体状况（响应）。

线性建模-多维问题

如果有d个属性，此时的属性向量$\mathbf{x}_n$，可以表示为

$$\mathbf{x}_n = \begin{pmatrix} 1 \\ x_{1,n} \\ \vdots \\ x_{d,n} \end{pmatrix}$$

那么多维下的预测可以写成

$$t_n = \omega_0 + \omega_1 x_{1,n} + \omega_2 x_{2,n} + \omega_3 x_{3,n} + \cdots + \omega_d x_{d,n}$$

$$= \begin{bmatrix} \omega_0 \\ \omega_1 \\ \omega_2 \\ \vdots \\ \omega_d \end{bmatrix}^{\mathrm{T}} \begin{pmatrix} 1 \\ x_{1,n} \\ \vdots \\ x_{d,n} \end{pmatrix} = \mathbf{w}^{\mathrm{T}}\mathbf{x}_n$$

注意：在上式中，ω_0也叫作“偏置项”。样本$\mathbf{x}_n$和系数向量**w**是d+1维。

线性建模-多维问题：模型的矩阵求解

现将N个样本的属性向量$\mathbf{x}_n=\left[x_{1,n},x_{2,n},\cdots,x_{d,n}\right]^{\mathrm{T}}$合并成一个矩阵**X**

$$\mathbf{X}=\begin{pmatrix}\mathbf{x}_1^{\mathrm{T}}\\ \mathbf{x}_2^{\mathrm{T}}\\ \vdots\\ \mathbf{x}_N^{\mathrm{T}}\end{pmatrix}=\begin{pmatrix}x_{1,1},x_{2,1},\cdots,x_{d,1}\\ x_{1,2},x_{2,2},\cdots,x_{d,2}\\ \vdots\\ x_{1,N},x_{2,N},\cdots,x_{d,N}\end{pmatrix}\in\Re^{N\times d}$$

也将N个样本的目标t_n也合并成一个向量

$$\mathbf{t}=(t_1,t_2,\cdots,t_N)^{\mathrm{T}}$$

对于{健康、患病}两种响应，通常可以用**+1**和**-1**表示目标**t.**

线性建模-多维问题：模型的矩阵求解

损失函数

$$L = \frac{1}{N}(\mathbf{Xw}-\mathbf{t})^{\mathrm{T}}(\mathbf{Xw}-\mathbf{t}) = \frac{1}{N}\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{Xw} - \frac{2}{N}\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{t} + \frac{1}{N}\mathbf{t}^{\mathrm{T}}\mathbf{t}$$

通过计算$\frac{\partial L}{\partial \mathbf{w}} = \begin{bmatrix} \frac{\partial L}{\partial \omega_1} \\ \frac{\partial L}{\partial \omega_2} \\ \vdots \\ \frac{\partial L}{\partial \omega_d} \end{bmatrix} = \frac{2}{N}\mathbf{X}^{\mathrm{T}}\mathbf{Xw} - \frac{2}{N}\mathbf{X}^{\mathrm{T}}\mathbf{t} = 0$

可以推出

$$\mathbf{X}^{\mathrm{T}}\mathbf{Xw} = \mathbf{X}^{\mathrm{T}}\mathbf{t}$$

那么$\mathbf{w} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}$（完毕）

线性建模-多维问题：模型的矩阵求解

举例：设有4个患者，测量血压值、心率 、体重三个属性。

现将4个样本的属性向量$\mathbf{x}_n=\left[x_{1,n},x_{2,n},x_{3,n}\right]^{\mathrm{T}}$合并成一个矩阵**X**

$$\mathbf{X}=\begin{pmatrix}\mathbf{x}_1^{\mathrm{T}}\\ \mathbf{x}_2^{\mathrm{T}}\\ \mathbf{x}_3^{\mathrm{T}}\\ \mathbf{x}_4^{\mathrm{T}}\end{pmatrix}=\begin{pmatrix}x_{1,1},x_{2,1},x_{3,1}\\ x_{1,2},x_{2,2},x_{3,2}\\ x_{1,3},x_{2,3},x_{3,3}\\ x_{1,4},x_{2,4},x_{3,4}\end{pmatrix}=\begin{pmatrix}80,70,50\\ 150,100,70\\ 100,80,60\\ 60,120,40\end{pmatrix}\in\Re^{4\times3}$$

也将N个样本的目标t_n也合并成一个向量

$$\mathbf{t}=(t_1,t_2,t_3,t_4)^{\mathrm{T}}=(1,-1,1,-\mathbf{1})^{\mathrm{T}}$$

对于{健康、患病}两种响应，分别用+1和-1表示.

线性建模-多维问题：模型的矩阵求解

一般，X中的数值很大，需要对属性值进行归一化处理，以便于求解和计算。归一化方法，通常除以属性最大值实现。

$$\mathbf{X}=\begin{pmatrix}80,70,50\\150,100,70\\100,80,60\\60,120,40\end{pmatrix}\rightarrow\begin{pmatrix}\frac{80}{150},\frac{70}{120},\frac{50}{70}\\\frac{150}{150},\frac{100}{120},\frac{70}{70}\\\frac{100}{150},\frac{80}{120},\frac{60}{70}\\\frac{60}{150},\frac{120}{120},\frac{40}{70}\end{pmatrix}\rightarrow\begin{pmatrix}\frac{8}{15},\frac{7}{12},\frac{5}{7}\\1,\frac{5}{6},1\\\frac{2}{3},\frac{2}{3},\frac{6}{7}\\\frac{2}{5},1,\frac{4}{7}\end{pmatrix}$$

利用解析形式$\mathbf{w}=\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}$，很容易计算出模型的参数$\mathbf{w}$

普通最小二乘线性回归总结：

- 模型的最优解，可以通过最小化误差的平方和（损失函数）进行计算。

$$L=\frac{1}{N}\sum_{n=1}^{N}(t_n-h(x_n;\omega_0,\omega_1))^2$$

- 最小二乘法回归算法存在闭解（解析解、精确解）

$\mathbf{w}=(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}$，其中**X**是样本数据矩阵，**t**是目标值矢量。

存在的问题：当**X**不是列满秩时，$\mathbf{X}^{\mathrm{T}}\mathbf{X}$是奇异矩阵，那么计算$(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$会产生较大误差，变成了一个不适定问题，该模型缺乏稳定性（对噪声的敏感性很强）。

模型的性能评价

已经介绍了普通最小二乘法的原理，对于由N个样本，计算出的最小二乘模型参数w，如何说明该模型是好的呢？

- 一般地，在机器学习中，为了验证模型的好坏，还需要一组新的样本（测试样本集）进行测试，根据测试准确率，判定模型的好坏。注：测试样本集是独立于N个样本的（也称训练样本）。
- 交叉验证（Leave-one-out cross validation, LOO）：K折交叉验证，也是目前机器学习和人工智能领域具有代表性的验证模型泛化能力的训练方法。

模型的性能评价

K折交叉验证（Leave-one-out cross validation, LOO）（留一法）：

将数据集分成相同数量的K份（K折），每份数据分别轮流作为测试集，其余的K-1份作为训练集。执行K次测试的平均精度，作为最后的模型精度。

特别注意的是：当K=N时，N折交叉验证，变成了“留一法”，即每个样本分别轮流作为测试样本（测试集中只有一个样本），剩余的N-1个样本作为训练集。

为什么要提出模型性能的评价方式？？

模型的性能评价

K折交叉验证（Leave-one-out cross validation, LOO）（留一法）：

将数据集分成相同数量的K份（K折），每份数据分别轮流作为测试集，其余的K-1份作为训练集。执行K次测试的平均精度，作为最后的模型精度。

特别注意的是：当K=N时，N折交叉验证，变成了“留一法”，即每个样本分别轮流作为测试样本（测试集中只有一个样本），剩余的N-1个样本作为训练集。

为什么要提出模型性能的评价方式？？

广义线性回归

给定一组样本：$\langle \mathbf{x}_i, y_i \rangle_{i=1\ldots m}$ ，我们要拟合下面的假设

$$h_{\mathbf{w}}(\mathbf{x}) = \sum_{k=0}^{K-1} w_k \phi_k(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x})$$

其中，ϕ_k 为基函数，可以是多阶的函数，比如平方、立方等。

为了求最佳的预测系数w, 我们需要最小化下列损失函数

$$\sum_{i=1}^{m} (y_i - h_{\mathbf{w}}(\mathbf{x}_i))^2$$

那么最优解w可以写为

$$\mathbf{w} = (\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{y}$$

广义线性回归

- 由前面的分析 可知，$h_w(x)$对于参数w来说，是线性模型。**但并不意味着**，$h_w(x)$是关于输入x的线性函数，比如$h_w(x) = wx^2$。(多项式回归)
- 因此，广义的线性模型可以写成

$$h_{\mathbf{w}}(\mathbf{x}) = \sum_{k=0}^{K-1} w_k \phi_k(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x})$$

其中，ϕ_k 为基函数。

- 线性假设模型还可以重新写成

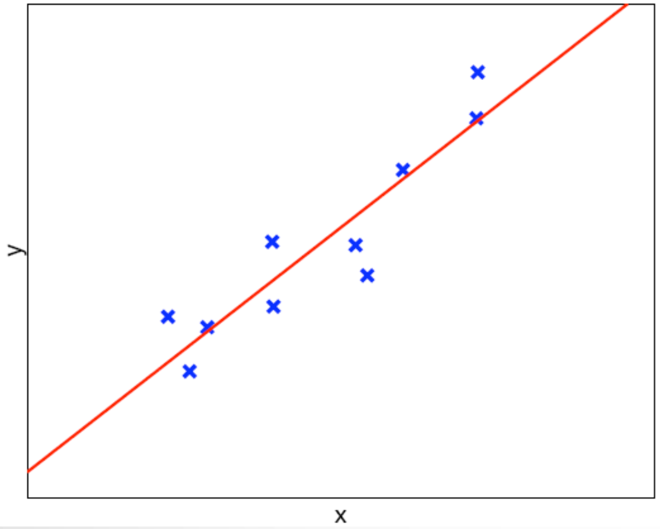
$$h_{\mathbf{w}}(\mathbf{x}) = \mathbf{\Phi w}$$

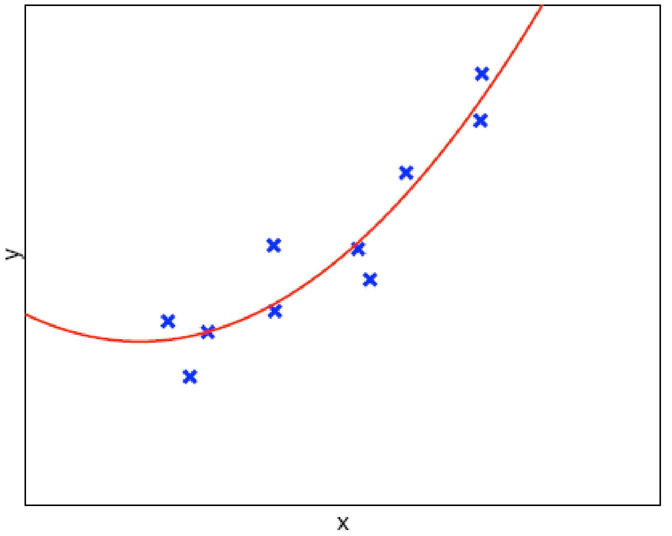
关于w的线性模型

其中，$\mathbf{\Phi}$是由 $\phi(\mathbf{x}_j)$构成的矩阵。

广义线性回归

采用不同基函数（不同阶次）的回归模型效果

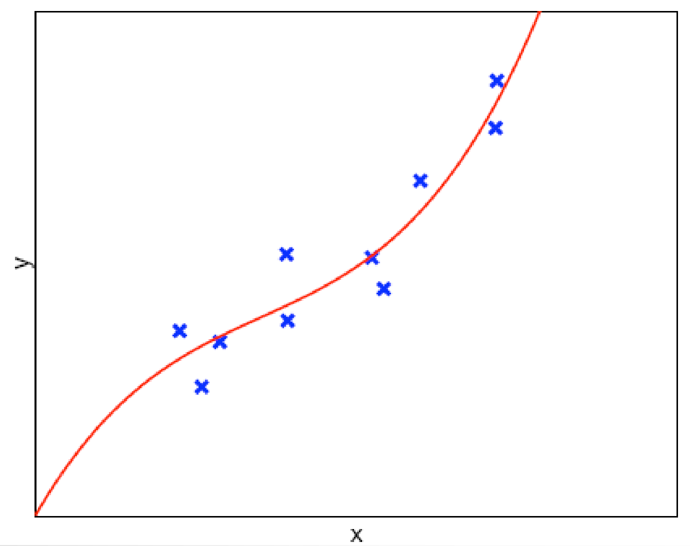


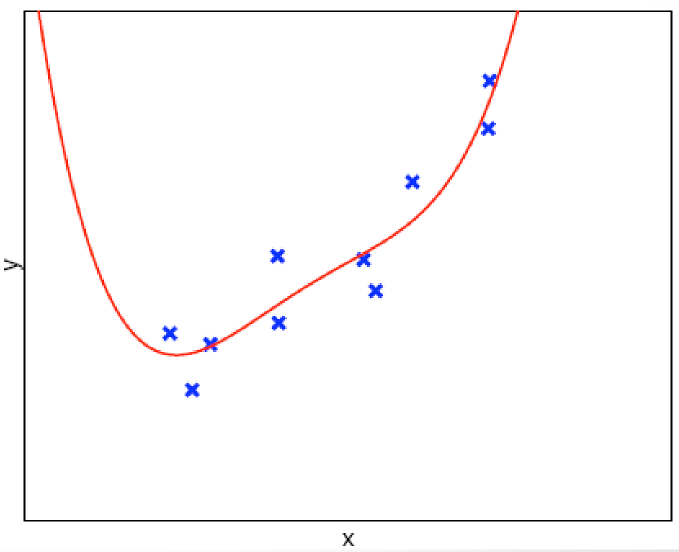


1阶回归 （is it better to fit the data?） 2阶回归

广义线性回归

采用不同基函数（不同阶次）的回归模型效果

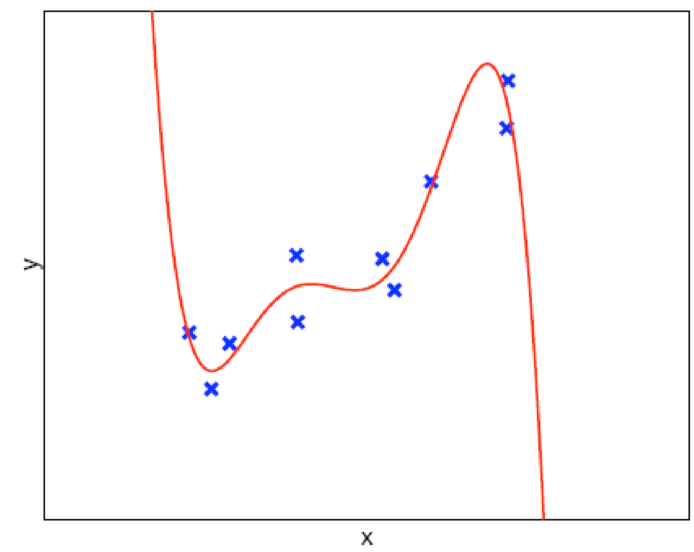


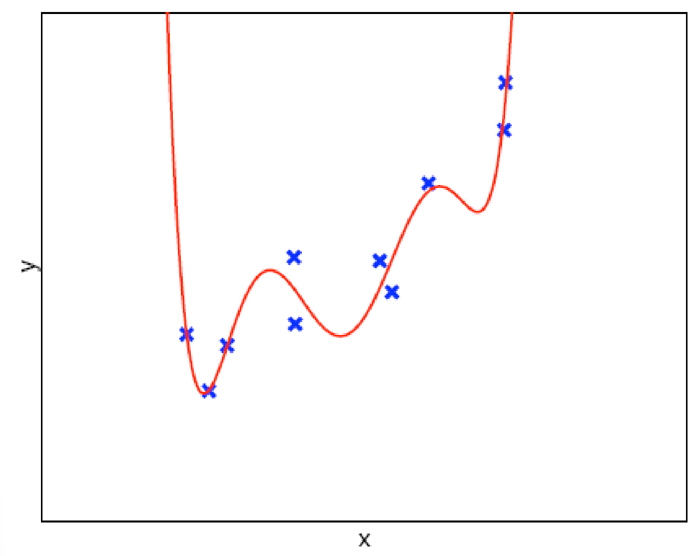


3阶回归 (is it better to fit the data?)　　4阶回归

广义线性回归

采用不同基函数（不同阶次）的回归模型效果

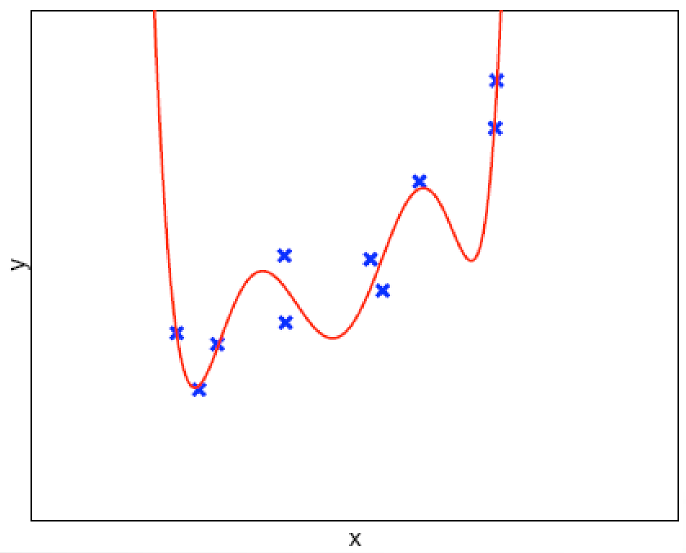


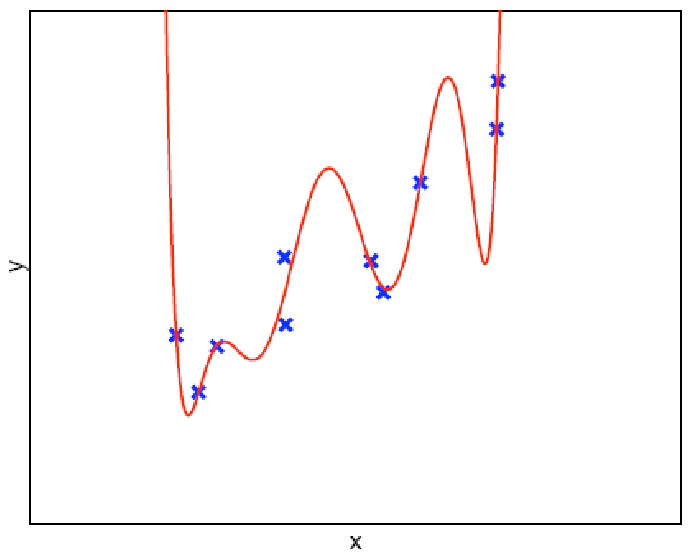


5阶回归 (is it better to fit the data?)　　6阶回归

广义线性回归

采用不同基函数（不同阶次）的回归模型效果

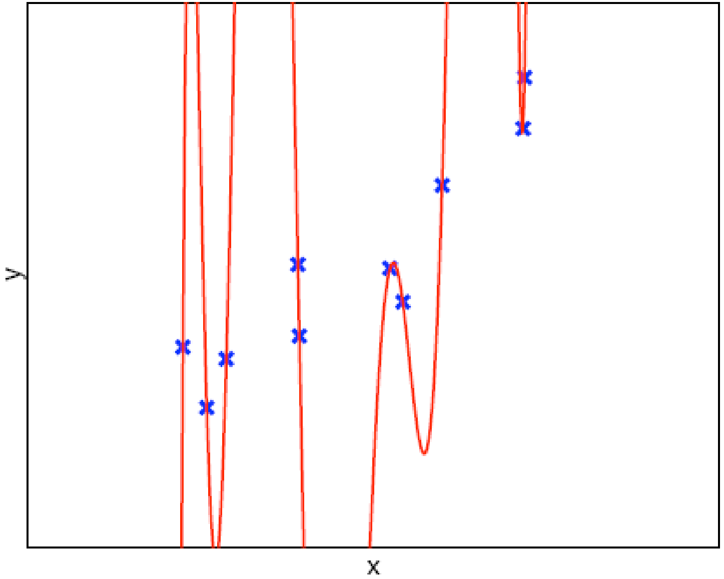




7阶回归 (is it better to fit the data?) **8阶回归**

广义线性回归

采用不同基函数（不同阶次）的回归模型效果

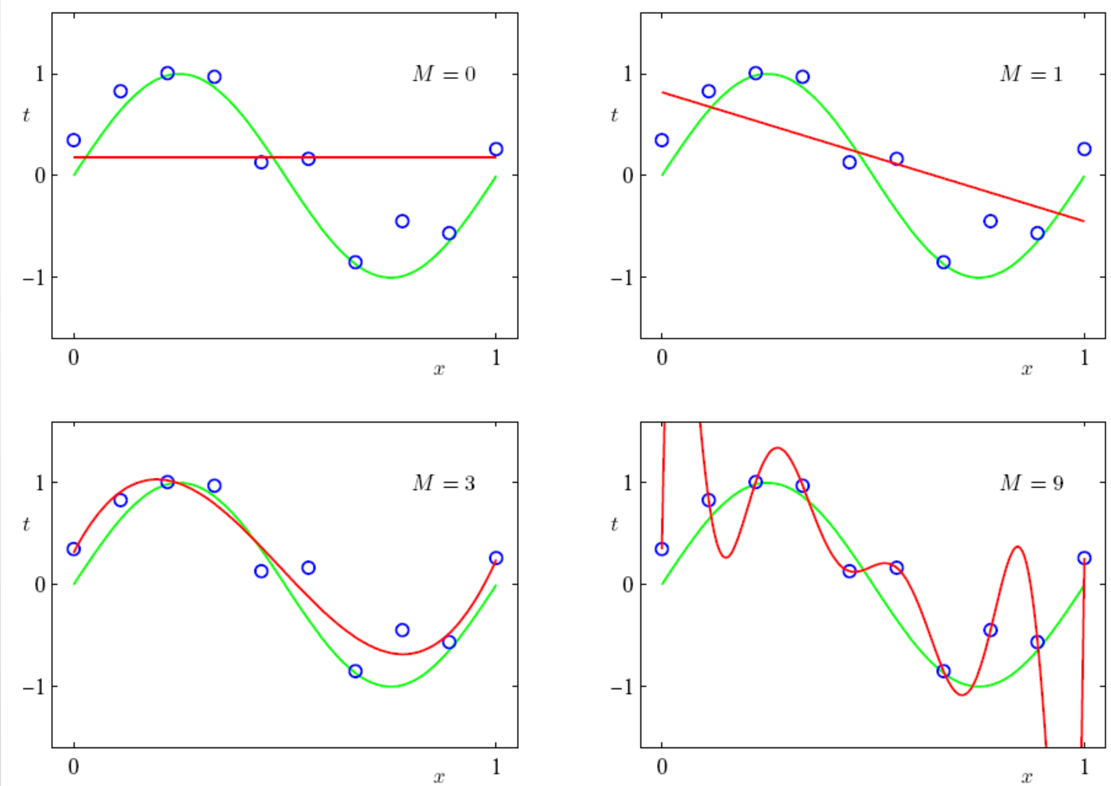


9阶回归 (is it better to fit the data?)

过拟合（overfitting）

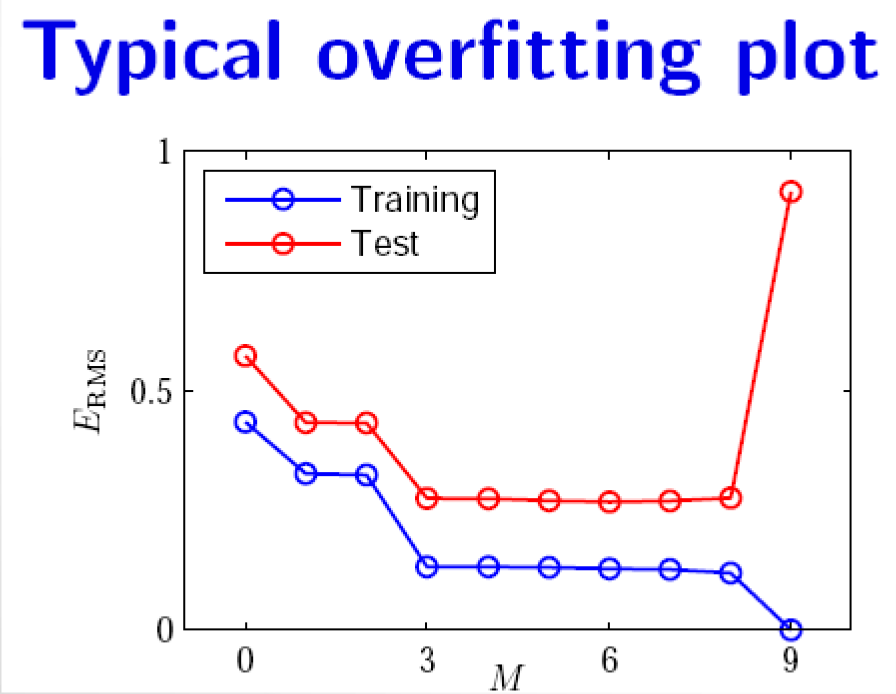
- 一般来说，过拟合是所有机器学习算法中的一个非常重要的问题；
- 通常，我们能够找到一种完美的假设，准确无误的预测训练数据。但是，对于新数据却不能够很好的预测（泛化能力很差）。
- 通过刚才的例子看到，如果参数越来越多，模型倾向于“记忆”训练数据点，而不是“推理”某种规律。

过拟合（overfitting）与欠拟合（underfitting）



- M为模型的阶数，反映模型的自由度和复杂度。
- **过拟合**是指，训练样本的误差非常低，而新测试样本的预测误差非常大。
- **欠拟合**是指，训练样本的预测误差非常高（学习不够）。
- 通常，**过拟合**是更为常见的问题，因此，在建模过程中，我们需要经验和理论分析，来避免这个问题。

过拟合（overfitting）与欠拟合（underfitting）



- 为从图中可以看出，M=0,1,2时，属于**欠拟合**（训练误差较大）
- M=3,4,5,6,7,8时，属于**拟合情况良好**；但根据模型的几个准则（参数越少、模型越简单越好），显然M=3是最佳参数。
- M=9时，属于**过拟合**。（训练误差非常小，接近0，但测试误差非常大，）

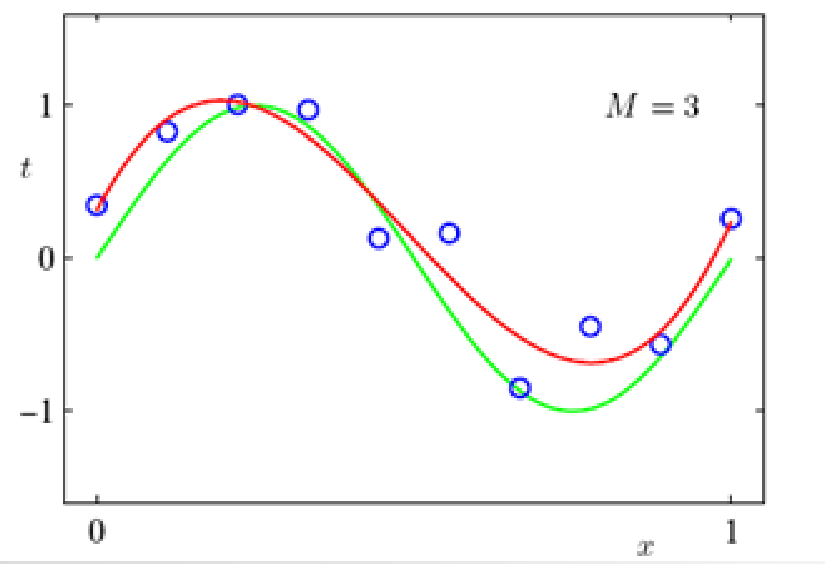
在设计模型寻求某种假设时，如何防止过拟合（overfitting）与欠拟合（underfitting）？

非常有效的交叉验证策略：

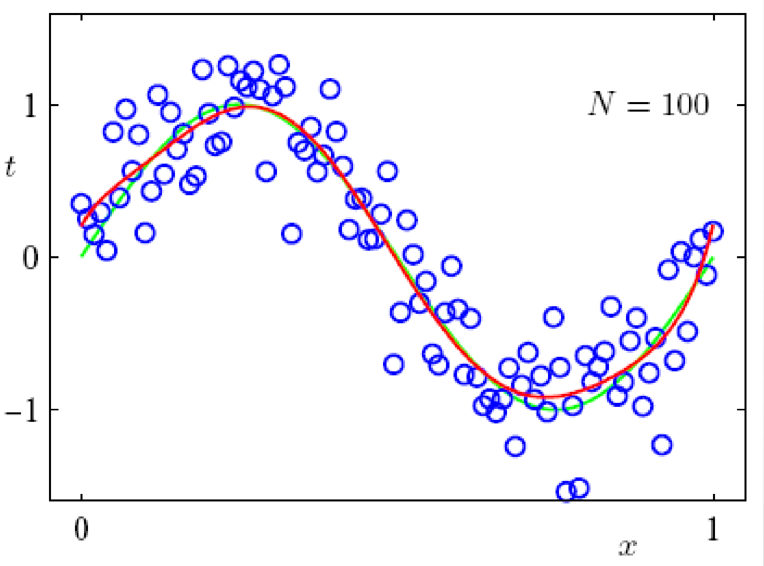
- 将整个数据集分成3个独立的部分：训练集、验证集、测试集
- ✓ 训练集是用于训练某种假设；
- ✓ 验证集用于检验假设的可靠性，并进一步修正模型（即*M*）；
- ✓ 测试集用于检验最终模型假设的性能。

一般地，采用随机分割的形式，执行以上程序N次，每次测试集的准确率的平均值为最终的模型的性能评价指标。

最佳的假设h(x)依赖于训练样本数量和模型复杂度



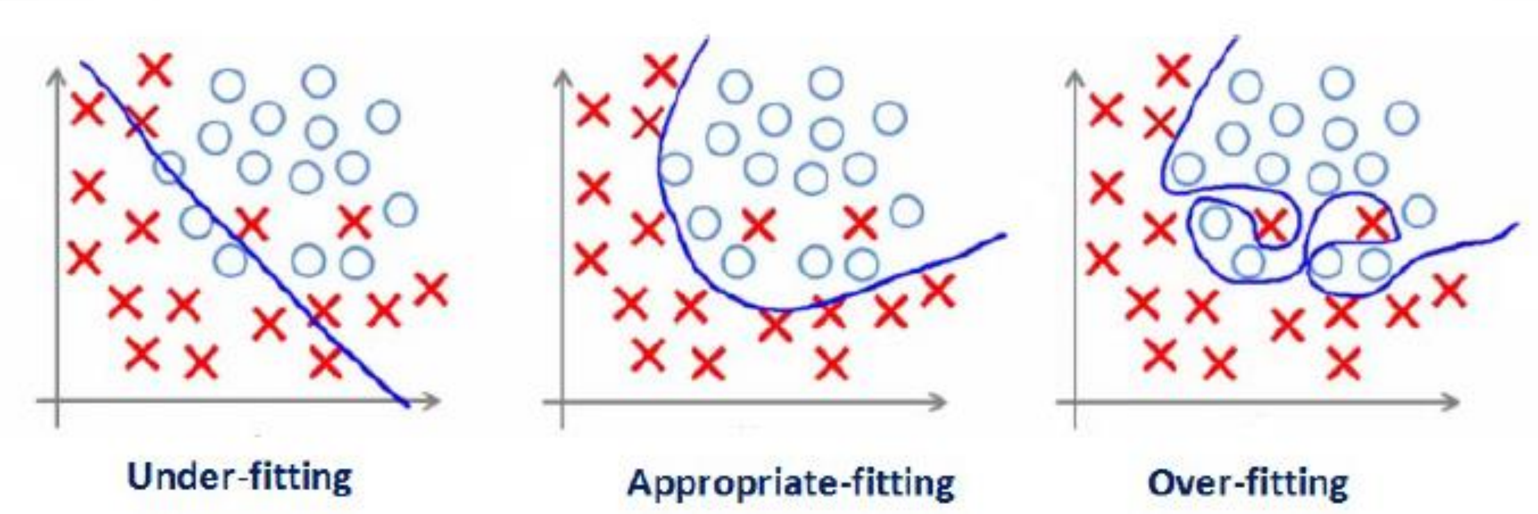
10个训练样本的回归曲线 （M=3）
红色表示真实曲线，绿色为假设



100个训练样本的回归曲线 (M=3)
假设与真实更加接近

最佳的假设h(x)依赖于训练样本数量和模型复杂度

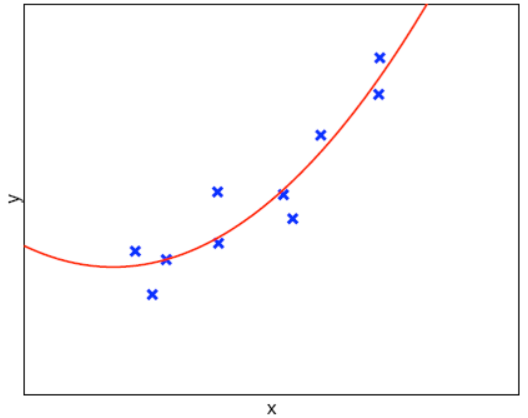
分类问题：

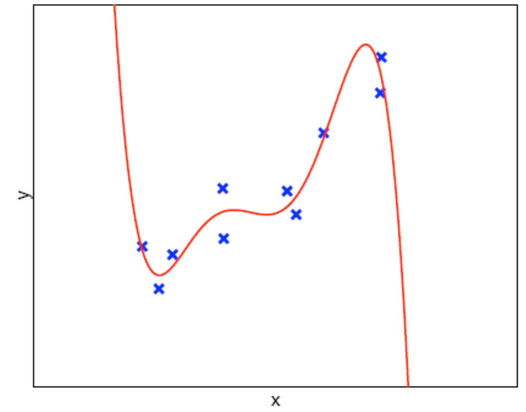


第五章： 正则化

正则化是机器学习也是智能信息处理中的重要一环，其目的是为了降低**模型复杂度**，增强模型的**抗噪声能力**，防止模型对训练数据集的**过拟合**（过学习）。换句话说，正则化是用于对模型参数的一种约束。

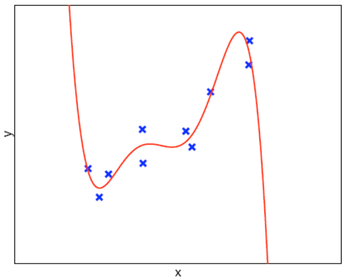
举例：一个2阶和5阶的拟合函数





$$h_\theta(x)=\theta_0+\theta_1 x+\theta_2 x^2$$

$$h_\theta(x)=\theta_0+\theta_1 x+\theta_2 x^2+\theta_3 x^3+\theta_4 x^4+\theta_5 x^5$$



原线性模型的最小化表达式为

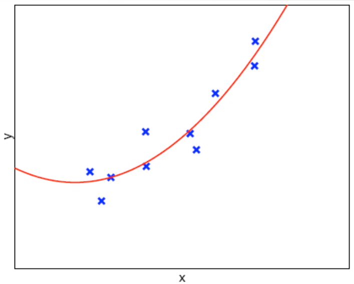
$$\min_{\theta} \frac{1}{N}\sum_{i=1}^{N}\left(h_\theta(x_i)-t_i\right)^2 \quad \Rightarrow \quad \boxed{h_\theta(x)=\theta_0+\theta_1 x+\theta_2 x^2+\theta_3 x^3+\theta_4 x^4+\theta_5 x^5}$$

为了获得一个较好的模型，我们希望模型复杂度越小越好（阶数越小），也就是希望得到 $\theta_3,\theta_4,\theta_5 \approx 0$ (参数变少，得到简单的 模型假设)

那么模型应当改进如下

$$\min_{\theta} \frac{1}{N}\sum_{i=1}^{N}\left(h_\theta(x_i)-t_i\right)^2+10000\cdot\theta_3^2+10000\cdot\theta_4^2+10000\cdot\theta_5^2$$

$$\boxed{h_\theta(x)=\theta_0+\theta_1 x+\theta_2 x^2}$$



正则化是机器学习也是智能信息处理中的重要一环，其目的是为了降低模型复杂度，增强模型的抗噪声能力，防止模型对训练数据集的过拟合（过学习）。换句话说，正则化是用于对模型参数的一种约束。

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \left(h_{\theta}\left(x_i\right) - t_i\right)^2 + \boxed{\lambda \cdot \sum_{j=1}^{m} \theta_j^2}$$

正则化项

正则化参数 λ 是一个可调的参数（误差项和正则项之间的平衡系数）,如果该值非常大,则获得的参数就会非常小! 容易产生"欠拟合",因此在算法编程中，可调试该参数。

正则化是机器学习也是智能信息处理中的重要一环，其目的是为了降低**模型复杂度**，增强模型的**抗噪声能力**，防止模型对训练数据集的**过拟合**（过学习）。换句话说，正则化是用于对模型参数的一种约束。

$$\min_{\theta} \frac{1}{N}\sum_{i=1}^{N}\left(h_{\theta}\left(x_i\right)-t_i\right)^2+\boxed{\lambda\cdot\sum_{j=1}^{m}\theta_j^2}$$

正则化项

实际上,正则化是对模型参数的约束,带有约束的模型可以进一步改写

正则化是机器学习也是智能信息处理中的重要一环，其目的是为了降低模型复杂度，增强模型的抗噪声能力，防止模型对训练数据集的过拟合（过学习）。换句话说，正则化是用于对模型参数的一种约束。

$$\min_{\theta} \frac{1}{N}\sum_{i=1}^{N}\left(h_{\theta}(x_i)-t_i\right)^2 + \boxed{\lambda \cdot \sum_{j=1}^{m}\theta_j^2}$$

正则化项

等价于

$$\boxed{\begin{aligned} &\min_{\theta} \frac{1}{N}\sum_{i=1}^{N}\left(h_{\theta}(x_i)-t_i\right)^2 \\ &s.t. \sum_{j=1}^{m}\theta_j^2 \le \xi \end{aligned}}$$

相当于在原模型基础上，加了一个约束项

等价性证明

$$\min_{\theta} \frac{1}{N}\sum_{i=1}^{N}\left(h_{\theta}\left(x_i\right)-t_i\right)^2+\lambda\cdot\sum_{j=1}^{m}\theta_j^2$$

$$\boxed{\begin{aligned}&\min_{\theta} \frac{1}{N}\sum_{i=1}^{N}\left(h_{\theta}\left(x_i\right)-t_i\right)^2\\ &s.t. \sum_{j=1}^{m}\theta_j^2\le\xi\end{aligned}}$$

对下面的模型利用拉格朗日乘子法构造目标函数

$$\min_{\theta} J(\theta)=\frac{1}{N}\sum_{i=1}^{N}\left(h_{\theta}\left(x_i\right)-t_i\right)^2+\lambda\cdot\left(\sum_{j=1}^{m}\theta_j^2-\xi\right)$$

□多维下的线性回归模型表达

定义一组观测数据集（**X**,**Y**）,其中$\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N]$, $\mathbf{Y}=[\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_N]$，**X**为属性，**Y**为响应标签。$\mathbf{x}_i$ $(i=1,\ldots,N)$ 是d-维的向量。

$$\min_{\boldsymbol{\theta}} \frac{1}{2N}\sum_{i=1}^{N}\left\|h_{\boldsymbol{\theta}}(\mathbf{x}_i)-\mathbf{y}_i\right\|_2^2$$

其中, $\boldsymbol{\theta}$ 是模型参数（向量或矩阵）。

□多维下的正则化线性回归模型表达

定义一组观测数据集（**X**,**Y**）,其中**X**=[$\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N$], **Y**=[$\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_N$], **X**为属性，**Y**为响应标签。$\mathbf{x}_i$ (i=1,...,N) 是d-维的向量。

$$\min_{\boldsymbol{\theta}} \frac{1}{2N}\sum_{i=1}^{N}\left\|h_{\boldsymbol{\theta}}(\mathbf{x}_i)-\mathbf{y}_i\right\|_2^2+\lambda\cdot\left\|\boldsymbol{\theta}\right\|_2^2$$

其中, $\boldsymbol{\theta}$ 是模型参数（向量或矩阵）。

与非正则化相比，目标函数中多了一项，<u>即在最小化误差的同时，还要保持参数值比较小。</u>

□ 正则化最小二乘模型（岭回归）

定义一组观测数据集（**X**,**Y**）,其中$\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N]$, $\mathbf{Y}=[\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_N]$，**X**为属性，**Y**为响应标签。$\mathbf{x}_i$ (i=1,…,N) 是d-维的向量。

$$\min_{\boldsymbol{\theta}}\sum_{i=1}^{N}\left\|\boldsymbol{\theta}^{\mathrm{T}}\mathbf{x}_i-\mathbf{y}_i\right\|_2^2+\lambda\cdot\left\|\boldsymbol{\theta}\right\|_2^2$$

等价于 $\Updownarrow$

$$\min_{\boldsymbol{\theta}}\left\|\mathbf{X}^{\mathrm{T}}\boldsymbol{\theta}-\mathbf{Y}\right\|_{\mathrm{F}}^2+\lambda\cdot\left\|\boldsymbol{\theta}\right\|_{\mathrm{F}}^2$$

其中, $\boldsymbol{\theta}$ 是模型参数（向量或矩阵）。

□ 正则化最小二乘模型（岭回归）

定义一组观测数据集（$\mathbf{X},\mathbf{Y}$）,其中$\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N]$, $\mathbf{Y}=[\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_N]$, $\mathbf{X}$为属性，$\mathbf{Y}$为响应标签。$\mathbf{x}_i$ (i=1,…,N) 是d-维的向量。

$$\min_{\boldsymbol{\theta}} \left\| \mathbf{X}^{\mathrm{T}}\boldsymbol{\theta} - \mathbf{Y} \right\|_{\mathrm{F}}^{2} + \lambda \cdot \left\| \boldsymbol{\theta} \right\|_{\mathrm{F}}^{2}$$

模型求解：

令 $J(\boldsymbol{\theta}) = \left\| \mathbf{X}^{\mathrm{T}}\boldsymbol{\theta} - \mathbf{Y} \right\|_{\mathrm{F}}^{2} + \lambda \cdot \left\| \boldsymbol{\theta} \right\|_{\mathrm{F}}^{2}$ 为目标函数。

求导：

$$\frac{dJ(\boldsymbol{\theta})}{d\boldsymbol{\theta}} = \mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\boldsymbol{\theta} - \mathbf{Y}\right) + \lambda \cdot \boldsymbol{\theta} = 0 \Rightarrow \boldsymbol{\theta} = \left(\mathbf{X}\mathbf{X}^{\mathrm{T}} + \lambda \cdot \mathbf{I}\right)^{-1}\mathbf{X}\mathbf{Y}$$

直接写出解析解

□多维下的正则化线性回归模型表达

定义一组观测数据集（**X**,**Y**）,其中**X**=[$\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N$], **Y**=[$\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_N$], **X**为属性， **Y**为响应标签。$\mathbf{x}_i$ (i=1,...,N) 是d-维的向量。

$$\min_{\boldsymbol{\theta}} \frac{1}{2N}\sum_{i=1}^{N}\left\|h_{\boldsymbol{\theta}}(\mathbf{x}_i)-\mathbf{y}_i\right\|_2^2+\lambda\cdot\left\|\boldsymbol{\theta}\right\|_2^2$$

其中, $\boldsymbol{\theta}$ 是模型参数（向量或矩阵）。

注：<u>这里正则化项的设计采用的是L2-norm(</u><u>Tikhonov regularization</u>**<u>),</u>**
<u>是一种很平滑的约束，但并不稀疏。</u>

□吉洪诺夫正则化(Tikhonov Regularization)

定义一组观测数据集（**X**,**Y**）,其中$\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N]$, $\mathbf{Y}=[\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_N]$，**X**为属性，**Y**为响应标签。$\mathbf{x}_i$ (i=1,…,N) 是d-维的向量。

$$\min_{\boldsymbol{\theta}} \frac{1}{2N}\sum_{i=1}^{N}\left\|h_{\boldsymbol{\theta}}(\mathbf{x}_i)-\mathbf{y}_i\right\|_2^2+\lambda\cdot\left\|\Gamma\boldsymbol{\theta}\right\|_2^2$$

其中，Γ是**吉洪诺夫**矩阵,在许多情况下$\Gamma=\alpha\cdot\mathbf{I}$。

对于L2-Norm,解的稀疏性不好，（**如何能保证求解的稀疏性？**）

□多维下的稀疏正则化线性回归模型表达

定义一组观测数据集（**X**,**Y**）,其中**X**=[$\mathbf{x}_1$,$\mathbf{x}_2$,…,$\mathbf{x}_N$], **Y**=[$\mathbf{y}_1$,$\mathbf{y}_2$,…,$\mathbf{y}_N$], **X**为属性，**Y**为响应标签。$\mathbf{x}_i$ (i=1,…,N) 是d-维的向量。

$$\min_{\boldsymbol{\theta}} \frac{1}{2N}\sum_{i=1}^{N}\left\| h_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right)-\mathbf{y}_i \right\|_2^2+\lambda\cdot\left\|\boldsymbol{\theta}\right\|_0$$

L0-范数表示向量θ中非零元素的个数。

注：这里正则化项的设计采用的是L0-norm,是一种强稀疏约束。

但求解时，由于L0的非凸性，难以求解NP-hard。通常采用松弛化。

□多维下的稀疏正则化线性回归模型表达 (Lasso Regression)

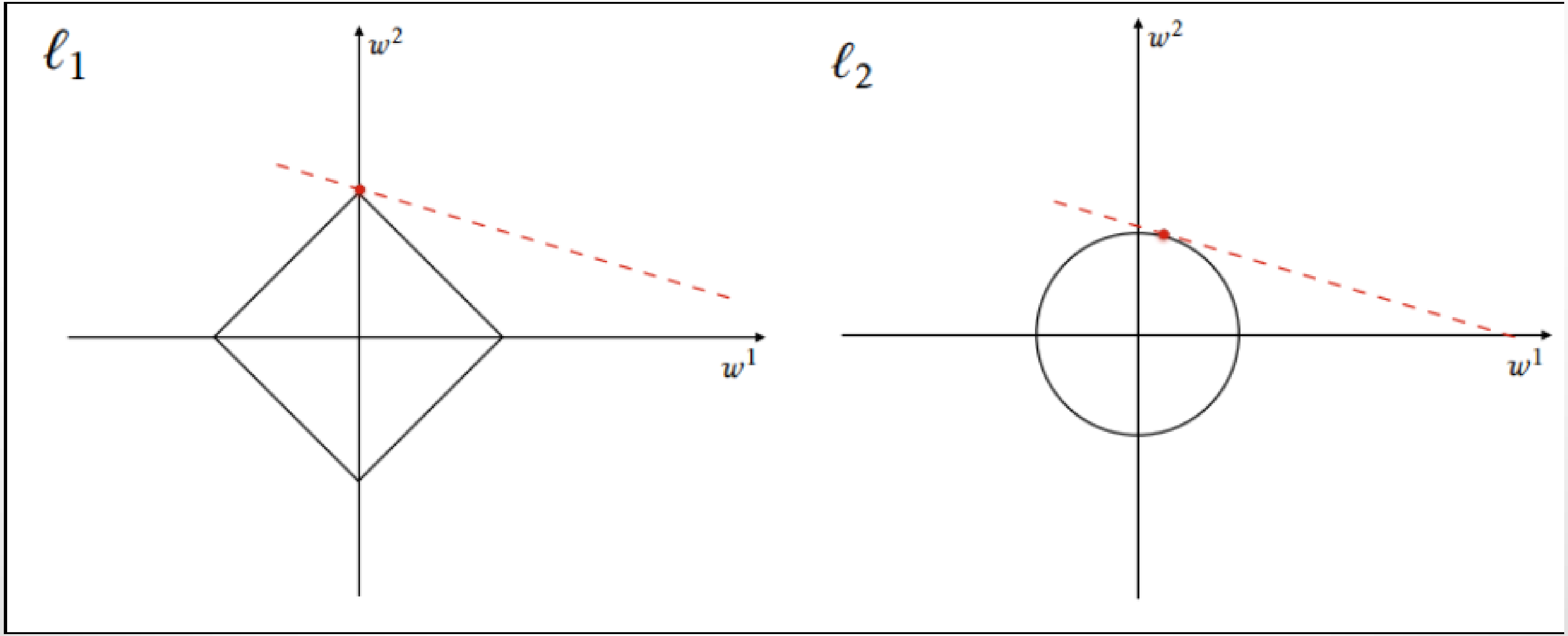
定义一组观测数据集（**X**,**Y**）,其中**X**=[$\mathbf{x}_1$,$\mathbf{x}_2$,…,$\mathbf{x}_N$], **Y**=[$\mathbf{y}_1$,$\mathbf{y}_2$,…,$\mathbf{y}_N$], **X**为属性， **Y**为响应标签。$\mathbf{x}_i$ (i=1,...,N) 是d-维的向量。

$$\min_{\boldsymbol{\theta}} \frac{1}{2N}\sum_{i=1}^{N}\left\|h_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right)-\mathbf{y}_i\right\|_2^2+\lambda\cdot\left\|\boldsymbol{\theta}\right\|_1$$

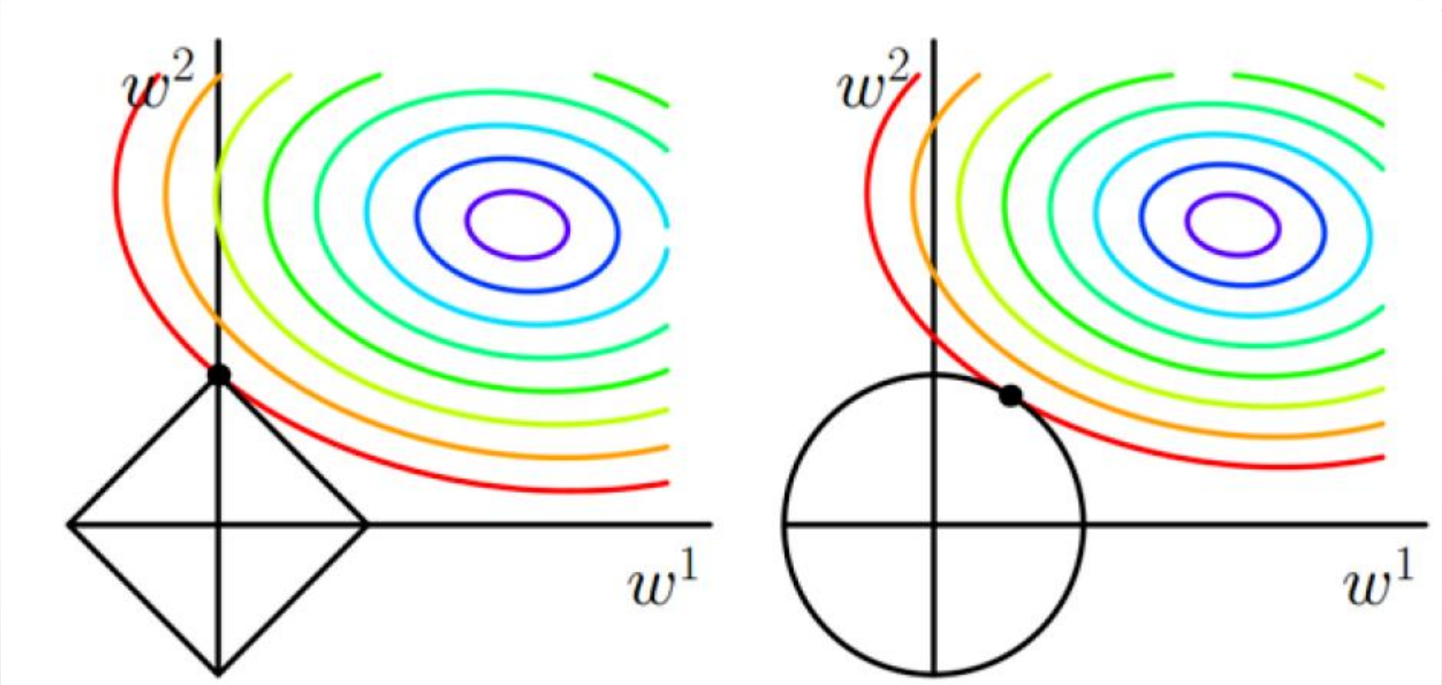
L1-范数的表达式。 $\left\|\boldsymbol{\theta}\right\|_1=\sum_{d=1}^{D}\left|\theta_d\right|$

注：<u>**这里正则化项的设计采用的是L1-norm,是L0的凸松弛，近似的稀疏解，可以通过Lasso算法迭代求解(虽然凸，但在x=0处不严格可导/可微，或者说在x=0处不存在导数)。**</u>

第五章：正则化regularization



第五章：正则化regularization



<u>**L1**有角，解落在角点的可能性更大，而**L2**没有角，因此不可能落在坐标系上。</u>

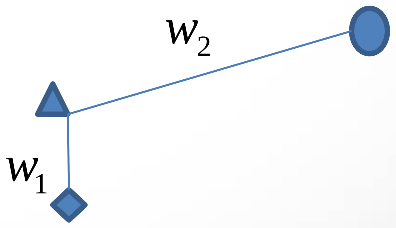
□半监督正则化（流形正则）

定义一组新的无标签的观测数据集**X**,其中**X**=[$\mathbf{x}_1$,$\mathbf{x}_2$,…,$\mathbf{x}_N$]，**X**为属性，无标签。$\mathbf{x}_i$ (i=1,…,N) 是d-维的向量。

因为样本无标签，则不能再用损失函数对函数 *f* 进行定义。

不妨做出以下假设：在欧式空间内，距离较近的两个样本$\mathbf{x}_1$,$\mathbf{x}_2$，其具有相同响应即$f(\mathbf{x}_1)= f(\mathbf{x}_2)$的概率更大。

设两点之间的连线, w存在一个权重即重要性系数。



□半监督正则化（流形正则）

按照上面的分析，半监督正则项的表达式可以写成

$$R(f)=\sum_{i,j} w_{i,j}\left(f(x_i)-f(x_j)\right)^2$$

其中，$w_{i,j}$是连接样本x_i和x_j之间的权重（重要性系数），属于给定的值，其定义方法可以根据欧式距离来表达。比如，

$$w_{i,j}=\exp\left(-\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2\right)$$

显然，权重矩阵W是一个对称阵。

■基于半监督正则化（流形正则）的线性建模

按照上面的分析，半监督正则项的表达式可以写成

$$\min_f \sum_k \left(f(x_k)-y_k\right)^2+\alpha\cdot\|f\|_2+\beta\cdot R(f)$$

$$\boxed{\min_f \sum_k \left(f(x_k)-y_k\right)^2+\alpha\cdot\|f\|_2+\beta\cdot\sum_{i,j} w_{i,j}\left(f(x_i)-f(x_j)\right)^2}$$

□正则化、泛化能力、过拟合三者关系

正则化的提出是用于提升模型的泛化能力，避免过拟合问题。

1. 泛化能力是指，训练模型不仅要保证训练集的误差最小化，同时还要保证测试集的性能，通常称为“偏置-方差”平衡问题（或者折中问题）。偏置是针对训练集，方差针对测试集。
2. 过拟合是指模型复杂度较高，导致训练集偏离程度很小（偏置很小），但测试误差的方差很大。
3. 欠拟合是指模型复杂度较低，模型越简单，训练集偏差较大；
4. 寻找泛化、欠拟合、过拟合之间的平衡问题，就是“偏置-方差”折中问题。

第六章：概率建模