TechX 深度强化学习课程（一）

何舜成

宽带网数字媒体实验室
清华大学自动化系

2018 年 7 月 26 日

目录

深度强化学习与人工智能

人工智能意味着什么？

人工智能（Artificial Intelligence, AI）指人制造出来的机器所表现出来的智能。1956 达特茅斯会议确定了 AI 的名称和任务，标志着 AI 的诞生。AI 发展了 60 多年，到现在已经成为一颗枝繁叶茂的大树。

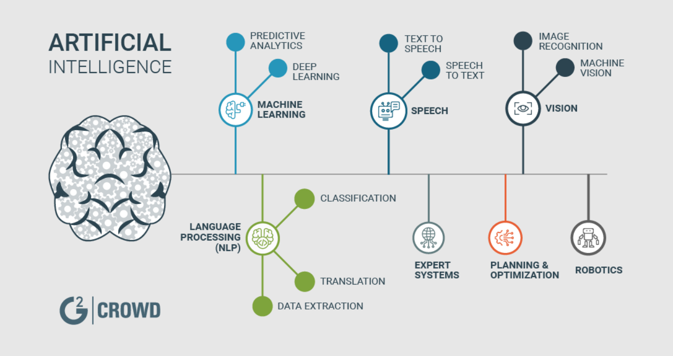


图 1: AI 的一种领域划分

人工智能意味着什么?

不同的 AI 技术解决不同的问题:

- 图像处理、卷积神经网络、LSTM——视觉与视频识别
- LSTM、机器翻译——自然语言处理
- 语音合成、语音识别——语音与文字转换
- 专家系统——推理与决策
- ? ——智能体、机器人

让机器人具有智能，需要感知环境，不断地探索和学习，应对环境的变化，这比传统的人工智能问题更复杂。这样一门学科就是**强化学习（Reinforcement Learning）**。

强化学习问题背景

倒立摆（Inverted Pendulum）

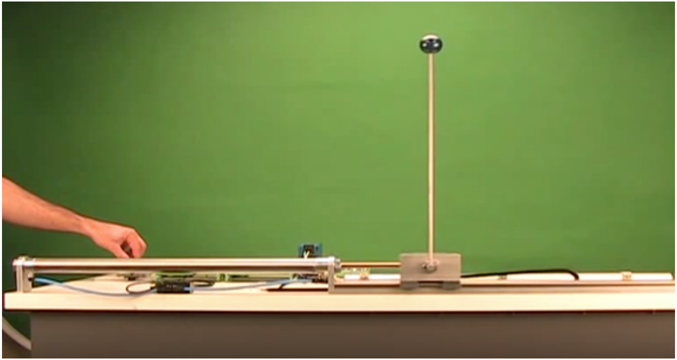


图 2: 倒立摆模型

检测物理量，设计控制器，驱动电机，使重物不掉下来。

强化学习问题背景

倒立摆问题的几大要素

- 状态（state）：当前系统的物理量如电机转角、角速度、固定点位置
- 动作（action）：可以控制的物理量如电机电流
- 控制器（agent）：根据输入的状态计算出输出电流的值
- 回报（reward）：给定时间内是否保持重物不掉落
- 环境（environment）：倒立摆的物理约束

强化学习问题背景

博弈问题的几大要素——以围棋为代表

- 状态：当前黑白棋子的位置
- 动作：在合法的地方落子
- 控制器：根据当前局势计算落子点
- 回报：一局结束后的胜负情况
- 环境：棋盘形状与围棋规则

图 3: Nature 杂志封面图

强化学习问题背景

实体：控制器（有时候叫“代理”）与环境
信号：动作、状态与回报

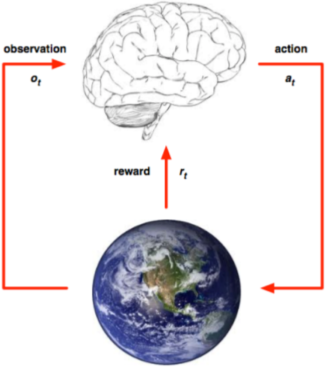


图 4: RL 示意图

深度学习的时代

图像分类的困境——从像素到语义

图 5: a cat?

图 6: a dog?

深度学习的时代

传统的图像处理方法：特征工程

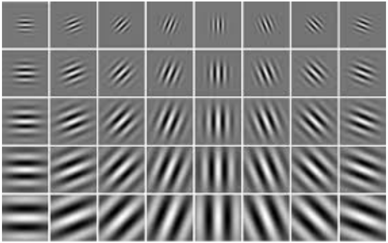


图 7: Gabor 滤波器的核

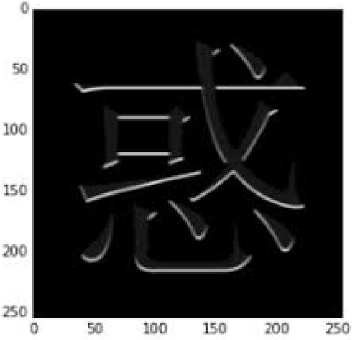


图 8: 边缘提取效果

深度学习的时代

传统的图像分类方法：浅层模型

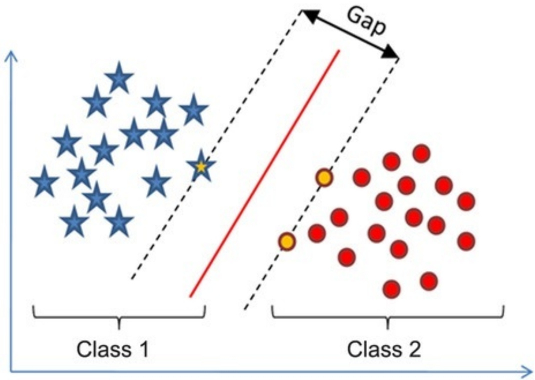


图 9: 用一个平面 $w^T x + b = 0$ 将数据点分开

深度学习的时代

传统机器学习时代：一个好的分类器 = 好的特征提取方法 + 合适的分类模型 + 有效、稳定的优化方法
深度学习时代：一个深层的神经网络

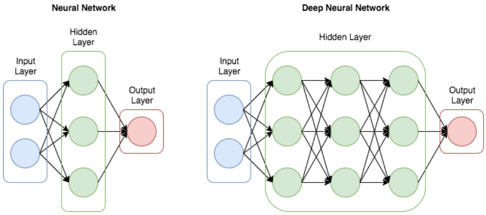


图 10: 深度神经网络

深度学习的时代

ImageNet Large Scale Visual Recognition Challenges (ILSVRC)

图 11: ImageNet 数据库的图片

从 2010 年开始举办，2017 年最后一届。计算机视觉与机器学习界最为重要的赛事。

深度学习的时代

从传统模型走向神经网络，从浅层模型走向深层模型。

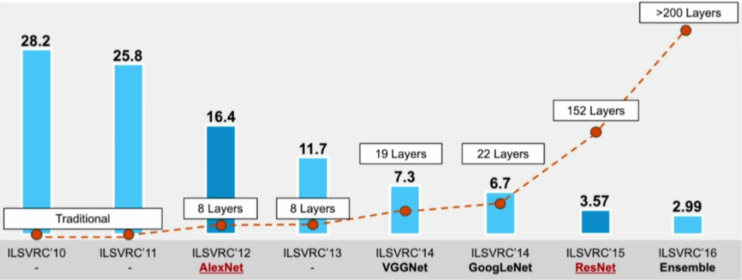


图 12: 物体识别误差

深度强化学习的含义

深度强化学习 ＝ 强化学习 ＋ 深度学习

- 强化学习为框架，一种特定的解决强化学习问题的思路
- 深度学习为手段，一种当前最有效的学习方式

Deep Reinforcement Learning 研究的是如果将二者结合起来。

课程知识体系

课程知识体系

本次课程将涉及的知识如下

- 微积分、线性代数、概率论、信息论、优化的相关内容——抽象和描述问题，及设计算法的工具
- Linux 操作系统、Python、TensorFlow 等编程知识——将算法实现的工具
- CNN 等深度神经网络算法
- 传统强化学习算法
- 深度强化学习算法

课程将以预习知识、上课讲授与答疑、课后练习、课程项目等形式进行。

线性代数

标量、向量、矩阵、张量

标量 a：一个单独的数，比如速度的大小，曲线在某一点的斜率/导数。

向量 $\boldsymbol{a}$：一组有序排列的数，比如空间中的速度矢量，曲面在某一点的梯度。通常情况下向量指列向量，即竖向排列的。向量中的第 i 个元素用 a_i 表示。

矩阵 $\boldsymbol{A}$：一个二维的数组，比如一张灰度数字图像。矩阵第 i 行，第 j 列的元素用 $A_{i,j}$ 表示。

张量 $\mathbf{A}$：一个维度超过二维的数组，比如一张彩色（RGB）的数字图像。其中坐标为 (i,j,k) 的元素用 $\mathsf{A}_{i,j,k}$ 表示。

矩阵转置

以矩阵的主对角线为轴，将元素翻转，定义为

$$(\boldsymbol{A}^{\top})_{i,j} = A_{j,i} \tag{1}$$

向量也可以进行转置，列向量转置后变成行向量。

标量转置后仍是自身（$a^{\top} = a$）。

矩阵乘积

两矩阵相乘须满足矩阵 $\boldsymbol{A}$ 的列数等于矩阵 MB 的行数，如 $\boldsymbol{A}$ 的形状是 $m \times n$，$\boldsymbol{B}$ 的形状是 $n \times p$，那么乘积得到的 $\boldsymbol{C}$ 的形状是 $m \times p$。乘法定义为

$$C_{i,j} = \sum_k A_{i,k} B_{k,j} \tag{2}$$

两个列向量的点积（内积）定义为

$$\boldsymbol{a}^\top \boldsymbol{b} = \sum_i a_i b_i \tag{3}$$

因此向量的内积也可以视作矩阵乘积，且矩阵乘积中 $C_{i,j}$ 的计算也可以看作是 $\boldsymbol{A}$ 的第 i 行与 $\boldsymbol{B}$ 的第 j 列之间作内积。

矩阵乘积

矩阵乘积不满足交换律，即 $\boldsymbol{AB} = \boldsymbol{BA}$ 不总是满足。分配率与结合律都满足。

向量内积满足交换律，即

$$\boldsymbol{a}^\top \boldsymbol{b} = \boldsymbol{b}^\top \boldsymbol{a} \tag{4}$$

矩阵乘积的转置等于矩阵分别转置后的乘积

$$(\boldsymbol{AB})^\top = \boldsymbol{B}^\top \boldsymbol{A}^\top \tag{5}$$

当线性方程组有 m 个方程和 n 个未知数，那么可以用系数矩阵 $\boldsymbol{A}$，未知数向量 $\boldsymbol{x}$ 和右侧向量 $\boldsymbol{b}$ 简洁地表示线性方程组。

$$\boldsymbol{Ax} = \boldsymbol{b} \tag{6}$$

单位矩阵与矩阵的逆

单位矩阵 $\boldsymbol{I}_n$ 是 $n \times n$ 的方阵，对角线全为 1，其余元素全为 0。例如

$$\boldsymbol{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

对于 $\boldsymbol{A}_{m \times n}$ 有，$\boldsymbol{A}\boldsymbol{I}_n = \boldsymbol{I}_m\boldsymbol{A} = \boldsymbol{A}$。

方阵 $\boldsymbol{A}$ 的逆 $\boldsymbol{A}^{-1}$ 定义为使 $\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}_n$ 成立的矩阵。矩阵的逆并不总是存在，如果存在，那么对于线性方程组 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ 则很容易写出解来

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b} \tag{7}$$

线性组合

一组向量的**线性组合**是各向量乘以对应的标量后相加

$$\sum_i c_i \boldsymbol{v}^{(i)} \tag{8}$$

那么线性方程组的左侧也可以看作是一组列向量的线性组合

$$\boldsymbol{A}\boldsymbol{x} = \sum_i x_i \boldsymbol{A}_{:,i} \tag{9}$$

若 x_i 均可任意取值，那么这个线性组合将产生一个点的集合，称为**生成子空间**。

以二元方程组为例：

$$\begin{aligned} 2x_1 + 0x_2 &= 4 \\ 1x_1 + 1x_2 &= 1 \end{aligned}$$

线性相关

再考虑下面这个方程组:

$$\begin{aligned} 2x_1 + 4x_2 &= 4 \\ 1x_1 + 2x_2 &= 1 \end{aligned}$$

显然方程组无解，此时方程组的列向量之间只有一个系数的差异，即第 2 个列向量可以由第 1 个列向量乘以一个系数得到，那么这两个列向量线性相关。推广到 n 个向量的情形，如果第 n 个向量可以写成其他 $n-1$ 个向量的线性组合，那么这 n 个向量**线性相关**。

$$\boldsymbol{x}_n = a_1\boldsymbol{x}_1 + a_2\boldsymbol{x}_2 + \cdots + a_{n-1}\boldsymbol{x}_{n-1} \tag{10}$$

线性方程组有唯一解的条件

现在我们可以来探讨 n 元线性方程组有唯一解的条件了。

首先方程的数目不能小于未知数的数目，否则可能存在很多组解。

其次系数矩阵的列向量不能线性相关，否则会存在冗余或冲突的方程。

即

- $\boldsymbol{A}$ 必须是方阵
- $\boldsymbol{A}$ 的列向量线性无关

这时 $\boldsymbol{A}$ 的逆 $\boldsymbol{A}^{-1}$ 存在且唯一。另外 $\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}_n$

向量范数

范数是描述向量某种意义上长度的函数。最常见的是欧几里得范数

$$\|\boldsymbol{x}\|_2 = \sqrt{\sum_i x_i^2} \tag{11}$$

一般意义下有一种 L^p 范数的定义

$$\|\boldsymbol{x}\|_p = \left(\sum_i x_i^p\right)^{\frac{1}{p}} \tag{12}$$

其中 L^1 范数可以写为

$$\|\boldsymbol{x}\|_1 = \sum_i |x_i| \tag{13}$$

L^∞ 范数可以写为

$$\|\boldsymbol{x}\|_\infty = \max_i |x_i| \tag{14}$$

正交矩阵

单位向量是 L^2 范数为 1 的向量，即 $\|\boldsymbol{x}\|_2 = 1$。

正交指向量内积为 0，即 $\boldsymbol{x}^\top \boldsymbol{y} = 0$。

如果一组单位向量两两正交，那么他们是标准正交的。

正交矩阵是行向量和列向量分别标准正交的方阵，有

$$\boldsymbol{A}^\top \boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^\top = \boldsymbol{I} \tag{15}$$

正交矩阵的逆很好求，因为逆即转置 $\boldsymbol{A}^{-1} = \boldsymbol{A}^\top$。

特征向量与特征值

若方阵 $\boldsymbol{A}$ 与向量 $\boldsymbol{v}$ 相乘，等价于对 $\boldsymbol{v}$ 进行缩放

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v} \tag{16}$$

那么 $\boldsymbol{v}$ 是 $\boldsymbol{A}$ 的一个特征向量，λ 是对应的特征值。

若 $\boldsymbol{A}$ 有 n 个线性无关的特征向量 $\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(n)}$，以及对应的特征值。将特征向量排成矩阵 $\boldsymbol{V} = [\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(n)}]$，将特征值排成向量 $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_n]$，那么

$$\boldsymbol{A} = \boldsymbol{V}\mathrm{diag}(\boldsymbol{\lambda})\boldsymbol{V}^{-1} \tag{17}$$

实对称矩阵可进一步进行正交分解

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\top} \tag{18}$$

正定矩阵

按特征值对矩阵分类：

- 正定矩阵：特征值全大于 0
- 半正定矩阵：特征值全不小于 0
- 负定矩阵：特征值全小于 0
- 半负定矩阵：特征值全不大于 0

正定矩阵可保证 $\forall \boldsymbol{x}, \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} \geq 0$ 且 $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} = 0$ 当且仅当 $\boldsymbol{x} = \boldsymbol{0}$

矩阵的几何理解

旋转矩阵（将二维向量逆时针旋转 θ 角度）

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

伸缩矩阵

$$\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

矩阵可以将一个向量映射成另一个向量，那么矩阵可以视作一种**线性变换**。矩阵的乘积对应着变换的复合。

矩阵的几何理解

行列式 $\det(\boldsymbol{A})$ 定义为 $\boldsymbol{A}$ 所有特征值的乘积。代表了线性变换在体积上的变化量。

例如一个 1×1 的方形可以视作 $(1,0)$ 和 $(0,1)$ 两个向量张成的一块面积为 1 的区域。当我们用下面两个矩阵作用上去

$$\boldsymbol{A}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

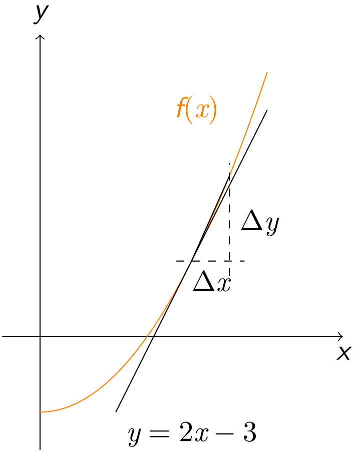
$$\boldsymbol{A}_2 = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -\sqrt{2}/2 \\ 0 & \sqrt{2}/2 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & \sqrt{2} \end{bmatrix}$$

$\boldsymbol{A}_1$ 和 $\boldsymbol{A}_2$ 的行列式都是 2，它们对原正方形的面积都扩大到了 2 倍。

微积分

导数的极限意义

函数 $f(x) = x^2/2 - 1$ 在 $x = 2$ 时的导数：



将导数定义为

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

导数与极值

我们经常关注导数为 0 的点，因为在这些地方可能出现函数的极值。一般来说有三种情况

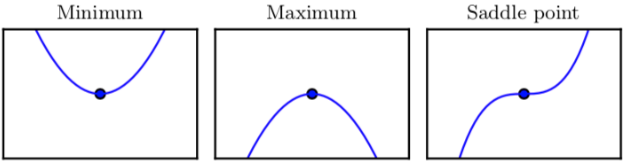


图 13: 极小值点、极大值点与鞍点

不同于极值点，鞍点左右的导数值符号相同，可以根据此来判断是不是真正的极值点。因此我们经常通过

$$f'(x) = 0 \tag{19}$$

来求解极值或最值。

多元函数与偏导

n 元函数表示为 $f(x_1, x_2, \dots, x_n)$，函数的值与多个自变量有关。

如果固定其中的一个自变量，那么就退化为一个 $n-1$ 元函数，如 $f(x_1 = 0, x_2, \dots, x_n)$。

多元函数对于其中的每一元都能定义类似导数的函数，称为**偏导数**。极限定义式

$$\frac{\partial f}{\partial x_i} = \lim_{h \to 0} \frac{f(x_1, \dots, x_i + h, \dots, x_n) - f(x_1, \dots, x_i, \dots, x_n)}{h} \tag{20}$$

可以将其他 $n-1$ 元看成是未知但固定的量，待求的偏导则是函数值沿 x_i 这一维上的变化率。

偏导与梯度

从刚才的定义上看，在计算偏导时，将其他变量视为常数即可。如 $f(x, y) = x^2 y$，有

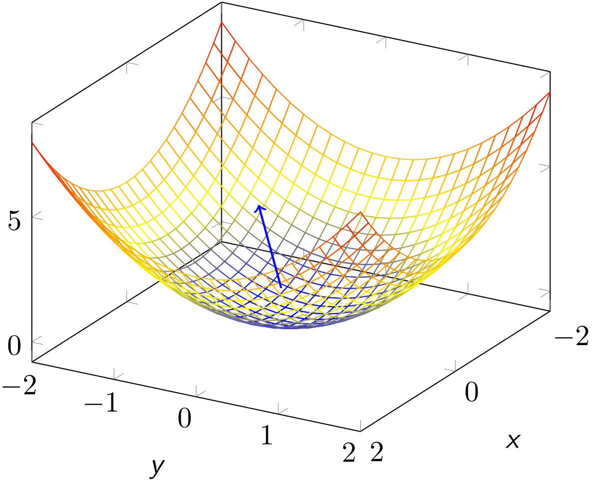
$$\frac{\partial f}{\partial x} = 2xy, \frac{\partial f}{\partial y} = x^2 \tag{21}$$

将各个偏导排列成向量，则得到了函数 f 的梯度

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right)^\top \tag{22}$$

偏导与梯度

可以证明，多元函数在某点的梯度是该函数在该点的**最速上升方向**。



高阶导数

n 阶导数是 $n-1$ 阶导数的导数。常用的是二阶导数，代表的是导数本身的变化率。

前面的例子里 $f(x)=x^2/2-1$，有

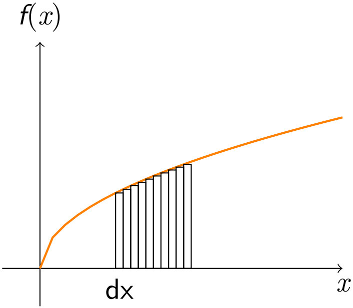
$$f'(x)=x, f''(x)=1 \tag{23}$$

定义高阶偏导：由于 n 元函数有 n 个偏导，那么二阶偏导会有 $n\times n$ 个，可以排列为矩阵，称为 **Hessian 矩阵**。

$$\nabla^2 f=\begin{bmatrix} \frac{\partial^2 f}{\partial^2 x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial^2 x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial^2 x_n} \end{bmatrix} \tag{24}$$

定积分

定积分来源于计算函数图像围起来的面积，比如下图函数 $f(x)=\sqrt{x}$ 在 $[1,2]$ 区间上的面积



这部分面积可以用定积分 $\int_1^2 \sqrt{x}dx$ 表示。
积分值有正有负。

不定积分

积分可以看做是导数的逆运算，一个函数 $f(x)$ 的不定积分指的是所有导数等于 $f(x)$ 的函数集合。例如函数 $F(x)$ 的导数是 $f(x)$，那么

$$\int f(x)dx = F(x) + C, C \text{ is a constant.} \tag{25}$$

不定积分与定积分有如下关系

$$\int_a^b f(x)dx = F(b) - F(a) \tag{26}$$

$$F(x) = \int_0^x f(x) + C \tag{27}$$

常见函数的导数

$$(x^n)' = nx^{n-1} \tag{28}$$

$$(e^x)' = e^x \tag{29}$$

$$(\ln(x))' = \frac{1}{x} \tag{30}$$

概率论

为什么要有概率

机器学习中概率是非常重要的工具，主要是在于现实问题往往具有不确定性，不确定性的来源主要有

- 物理系统内在的随机性。现代物理表明，在微观层面上，粒子具有不确定性，量子力学就是用概率去描述微观物质的。在宏观层面上，由于动力系统的特性，微小的扰动往往可以引起混沌效应。
- 不完全观测。有些情形下，信息对某一观测者是确定的，而对另一观测者是不确定的，例如抽签。
- 不完全建模。现实的物理系统非常复杂，在一些机器人学问题上，虽然我们可以写出整个机械臂的动力学方程，但由于无法计算，往往会进行简化和近似，这将引入不确定性。

为什么要有概率

这种情况下，概率描述了一种把握或信念。比如医生根据患者的症状进行诊断，往往会有一个诊断的把握度。疾病对应的相关症状出现得越多越明显，医生可能对确诊的把握度更高。

机器学习算法同样需要一个概率的描述。对于一张动物的图片，算法往往输出图片是 A 动物的概率、是 B 动物的概率等等。因此学习概率论是自然而然的要求。

随机变量

随机变量是可以随机地取不同值的变量，描述一个随机变量，需要知道随机变量可取值的范围，以及取不同值的概率分布。标量随机变量 x 的一个取值用 x 表示，向量随机变量 $\mathbf{x}$ 的一个取值用 $\boldsymbol{x}$ 表示。

离散型随机变量是指取值离散，例如抛一枚硬币的结果，有正反面两种取值。连续型随机变量的取值是实数值，例如一个放射性原子什么时候会发生衰变，取值是 $[0, +\infty)$。

离散型随机变量由概率质量函数 $P(\mathrm{x})$ 描述，在抛硬币的例子里面，$P(\mathrm{x} = head) = 0.5$，$P(\mathrm{x} = tail) = 0.5$。

连续型随机变量由概率密度函数 $p(\mathrm{x})$ 描述，在衰变时间的例子里面，$p(x) = \lambda e^{-\lambda x}$

概率密度函数的条件

离散型随机变量的概率质量函数与连续型随机变量的概率密度函数都需要满足三个条件

- 定义在所有可能的取值上
- 概率非负
- 和为 1，$\sum_x P(x) = 1$ 或 $\int p(x)dx = 1$

由第二三条还可以推出 $P(\mathrm{x} = x) \leq 1$。

联合概率分布与边际分布

抛两枚硬币，第一枚硬币的结果为 x，第二枚硬币的结果为 y，很容易写出联合概率分布：

$$
\begin{aligned}
P(\mathrm{x}=\mathit{head},\mathrm{y}=\mathit{head}) &= 0.25\\
P(\mathrm{x}=\mathit{head},\mathrm{y}=\mathit{tail}) &= 0.25\\
P(\mathrm{x}=\mathit{tail},\mathrm{y}=\mathit{head}) &= 0.25\\
P(\mathrm{x}=\mathit{tail},\mathrm{y}=\mathit{tail}) &= 0.25
\end{aligned}
$$

边际概率则定义为

$$
P(\mathrm{x}=x)=\sum_{y}P(\mathrm{x}=x,\mathrm{y}=y) \tag{31}
$$

连续情形下

$$
p(x)=\int p(x,y)dy \tag{32}
$$

条件概率与链式法则

条件概率是指多个事件中，给定其他事件出现后，剩下的事件发生的概率分布，计算方式如下

$$P(\mathrm{x} = x | \mathrm{y} = y) = \frac{P(\mathrm{x} = x, \mathrm{y} = y)}{P(\mathrm{y} = y)} \tag{33}$$

即联合概率除以边际概率。很容易就能推出

$$\begin{aligned} P(\mathrm{x}|\mathrm{y})P(\mathrm{y}) &= P(\mathrm{x}, \mathrm{y}) \\ P(\mathrm{x}|\mathrm{y}, \mathrm{z})P(\mathrm{y}|\mathrm{z})P(\mathrm{z}) &= P(\mathrm{x}, \mathrm{y}, \mathrm{z}) \end{aligned}$$

条件概率与链式法则

以预测题最后一题为例。

转移概率矩阵

$$\boldsymbol{A} = \{a_{ij}\} = \begin{bmatrix} 0.5 & 0.25 & 0.25 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \tag{34}$$

已知今天是晴天，求之后三天都是雨天的概率。设 s_1, s_2, s_3 分别表示晴、阴、雨，$\mathrm{t}_0, \ldots, \mathrm{t}_3$ 表示这几天的天气，那么

$$\begin{aligned} & P(\mathrm{t}_3 = s_3, \mathrm{t}_2 = s_3, \mathrm{t}_1 = s_3 | \mathrm{t}_0 = s_1) \\ = \quad & P(\mathrm{t}_3 = s_3 | \mathrm{t}_2 = s_3, \mathrm{t}_1 = s_3, \mathrm{t}_0 = s_1) \\ & P(\mathrm{t}_2 = s_3 | \mathrm{t}_1 = s_3, \mathrm{t}_0 = s_1) P(\mathrm{t}_1 = s_3 | \mathrm{t}_0 = s_1) \end{aligned}$$

贝叶斯定理

回顾条件概率的式子，$P(\mathrm{x},\mathrm{y})$ 可以写成两种形式

$$P(\mathrm{x},\mathrm{y}) = P(\mathrm{x}|\mathrm{y})P(\mathrm{y}) = P(\mathrm{y}|\mathrm{x})P(\mathrm{x}) \tag{35}$$

整理一下就有

$$P(\mathrm{x}|\mathrm{y}) = \frac{P(\mathrm{y}|\mathrm{x})P(\mathrm{x})}{P(\mathrm{y})} \tag{36}$$

这就是**贝叶斯公式**，其中 $P(\mathrm{y})$ 是 y 的边际概率分布

$$P(\mathrm{y}) = \sum_{x} P(\mathrm{y}|x)P(x) \tag{37}$$

独立性

如果随机变量 x 和 y 的联合概率分布可以写成下面的形式

$$\forall x, y, P(\mathrm{x} = x, \mathrm{y} = y) = P(\mathrm{x} = x) P(\mathrm{y} = y) \tag{38}$$

在抛硬币的例子里，两枚硬币的结果是相互独立的，在天气预测的例子里，每一天的天气情况不是相互独立的。同样还可以定义条件独立性。

$\mathrm{x} \perp \mathrm{y}$ 表示 x 和 y 相互独立，$\mathrm{x} \perp \mathrm{y} \mid \mathrm{z}$ 表示 x 和 y 在给定 z 时条件独立。

期望、方差与协方差

有一些随机变量的特征是我们关注的，比如期望刻画了随机变量的平均水平，定义如下

$$\mathbb{E}_{\mathrm{x} \sim P}[\mathrm{x}] = \sum_{x} x P(x) \tag{39}$$

$$\mathbb{E}_{\mathrm{x} \sim p}[\mathrm{x}] = \int x p(x) dx \tag{40}$$

方差则衡量了随机变量的取值围绕期望有多大的偏差

$$\mathrm{Var}(\mathrm{x}) = \mathbb{E}\left[(\mathrm{x} - \mathbb{E}[\mathrm{x}])^2\right] \tag{41}$$

协方差则衡量了两个变量线性相关的强度

$$\mathrm{Cov}(\mathrm{x}, \mathrm{y}) = \mathbb{E}\left[(\mathrm{x} - \mathbb{E}[\mathrm{x}])(\mathrm{y} - \mathbb{E}[\mathrm{y}])\right] \tag{42}$$

期望、方差与协方差

期望是线性的

$$\mathbb{E}[\alpha f(\mathrm{x})+\beta g(\mathrm{x})]=\alpha\mathbb{E}[f(\mathrm{x})]+\beta\mathbb{E}[g(\mathrm{x})] \tag{43}$$

方差的分解

$$\mathrm{Var}(\mathrm{x})=\mathbb{E}[\mathrm{x}^2]-(\mathbb{E}[\mathrm{x}])^2 \tag{44}$$

独立随机变量的协方差为 0，反之则不一定（为什么?)。

常见概率分布

伯努利分布是取值为 0 或 1 的分布，参数为 p

$$P(\mathrm{x}=1)=p \tag{45}$$

$$P(\mathrm{x}=0)=1-p \tag{46}$$

$$\mathbb{E}[\mathrm{x}]=p \tag{47}$$

$$\mathrm{Var}(\mathrm{x})=p(1-p) \tag{48}$$

常见概率分布

二项分布描述的是 n 个独立的服从参数为 p 的伯努利分布的随机变量之和

$$P(\mathrm{x}=k)=\binom{n}{k}p^k(1-p)^{n-k} \tag{49}$$

其中 $\binom{n}{k}=\frac{n!}{k!(n-k)!}$

$$\mathbb{E}[\mathrm{x}]=np \tag{50}$$

$$\mathrm{Var}(\mathrm{x})=np(1-p) \tag{51}$$

独立同分布简写为 i.i.d（independent identical distribution）

常见概率分布

参数为 λ 的指数分布有如下概率密度函数

$$p(x) = \lambda \boldsymbol{I}_{x \geq 0} e^{-\lambda x} \tag{52}$$

$\boldsymbol{I}_{x \geq 0}$ 是指示函数，表示在下标范围内取 1，其余地方取 0

$$\mathbb{E}[\mathrm{x}] = \mathrm{Var}(\mathrm{x}) = \lambda \tag{53}$$

常见概率分布

最常用的分布是正态分布

$$\mathcal{N}(x;\mu,\sigma^2)=\sqrt{\frac{1}{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right) \tag{54}$$

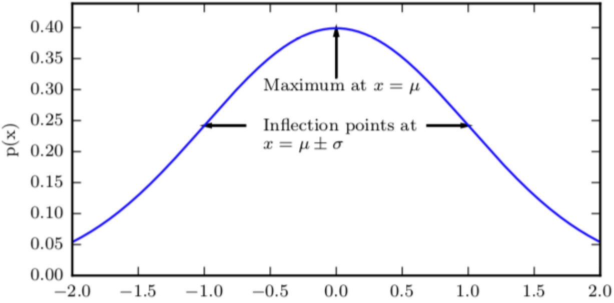


图 14: 正态分布的概率密度函数

常见概率分布

正态分布中 $\mathbb{E}[\mathrm{x}] = \mu, \mathsf{Var}(\mathrm{x}) = \sigma^2$，控制了分布的中心点和宽度。

在我们不知道某些量的分布的时候，通常会假定它服从正态分布。自然界和生活中有许多分布是近似正态分布的，例如考试分数的分布，工厂加工零件时尺寸误差的分布，噪声的频率分布等。在概率论里，中心极限定理证明了多个 i.i.d 的随机变量之和近似服从正态分布。

大数定律与中心极限定理

当 n 个随机变量 $x_1, \ldots, x_n$ 独立同分布，期望 $\mathbb{E}[x_i] = \mu$，方差 σ^2 有限时，它们的均值以概率为 1 收敛于期望值。(大数定律)

$$P(\lim_{n\to\infty} \bar{x}_n = \mu) = 1 \tag{55}$$

记 $M_n = \frac{\bar{x}_n - \mu}{\sigma\sqrt{n}}$，那么 M_n 收敛到标准正态分布。（中心极限定理）

课后练习：模拟 n 个 0-1 分布的均值，观察均值随 n 的变化。固定一个足够大的 n，再将本过程模拟 m 遍，验证中心极限定理。

Linux 系统熟悉

终端命令使用

本课程中需要对命令行操作十分熟悉，Ubuntu 和 OS X 系统皆可用于学习，最后的课程项目将在 Ubuntu 云服务器上完成。

终端演示：

- 调出和关闭终端
- 文件系统结构
- 用户权限
- 部分实用命令
- 文件编辑
- 包管理

详细内容可查阅预习内容。

编辑器

文本编辑器是编程第一生产力。

命令行模式下的 vim 熟悉：

- vim 安装
- 新建文件
- 光标移动
- 输入
- 查找
- 保存退出

Python 基础

Python 简介

本课程主要的编程语言是 Python，大部分 OS X 系统和 Ubuntu 系统已自带 python，不过课程中以 python3 为主，可能需要进行更新。

运行 Python 最简单的方式是在命令行中运行交互式 python，比如输入最简单的命令：

```python
print('Hello World!')
```

交互式对于简单的程序来说没问题，但是对于稍复杂一些的程序，就不适合了。我们将代码保存为.py 文件，然后调用 python 运行。一般来说有两种办法

- 用 vim、Sublime Text 等文本编辑器编辑之后在终端中调用 python 运行或调试
- 用 PyCharm 等集成环境编写、运行或调试

我们建议使用后者。

基本数据类型

Python 不需要显式声明数据类型，解释器将根据实际情况确定数据类型，比如当赋值a=2，b=2.5时，a和b就会分别推断为整型和浮点型。需要注意的是，地板除在 python3 中是//。

常用的数据类型还有字符串型，用单引号或双引号括起来。如果字符串中有双引号，那么需要进行“转义”。

格式化字符串可以将变量加到字符串里，如`print('%d-%f'% (b, a))`或是利用字符串类型的`format`方法。

类型之间可以进行强制转换，比如`c=int(a)`，相当于做一次取整操作。

条件判断与循环

Python 的条件判断关键字为`if...elif...else`，注意不要和其他编程语言混淆。

常用的循环语句有`for x in ...`和`while ...`。

Python 中代码块之间的层次关系由缩进决定，条件判断与循环语句关键字后的代码块要缩进一级。

List、Tuple、Dict、Set

List 的几个特性

- 可用下标进行索引，可用负数下标从后往前索引
- 可在`for`语句中进行迭代
- 一般用append方法添加元素，可用+连接两个 list
- 可用`len()`函数获得列表长度

Tuple 也可进行索引，用加号连接，但 tuple 中的元素一旦确定就不能再改变。

List、Tuple、Dict、Set

Dict 的几个特性

- 一般用 key 去索引对应的 value
- 可进行迭代，迭代根据 key 来进行
- 可直接用`a[key]=value`来添加元素，key 不能与已有的重复，否则将覆盖前值

Set 可以看做是没有 value 的字典。

函数

函数的定义：

```python
def func(p1, p2, p3):
    ...
    return x
```

位置参数、默认参数、可变参数与关键字参数：

```python
def func(p1, p2=0, *args, **kw):
    ...
    return x
```

类与实例

类出于编程时对于封装的考虑，将一个较为完整的模块通过类包装好，方便理清代码间的逻辑关系和数据间的归属关系。

类的定义以及实例化示例：

```
class Optimizer(object):

    def __init__(self, p1, p2=0, **kw):
        self.p=p1
        pass

    def func1(self):
        pass

opt1 = Optimizer(0.05, p2=4, x1=3, x2=1)
```

类与实例

父类：继承自 A 类，那么将拥有 A 类所定义的全部内部变量和方法，如无父类，可写`object`。

关键字参数：可以方便地将参数传给类内其他类的初始化实例。

初始化函数：名字一定是`__init__`。

实际运用 Python 的时候可能出现很多新的知识点，可以边用边学。

机器学习概念

分类问题

回到最开始的例子，判断图片里是哪种动物?

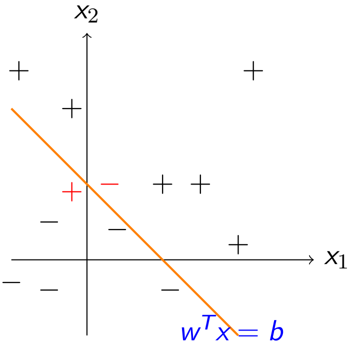
图 15: a cat?

图 16: a dog?

二分类问题：给定输入数据 x（如一张图片），分类器 $y = f(x)$ 输出两个类别中的一个，即 $y \in \{\pm 1\}$。

分类问题

举一个简单的例子，平面上有一堆散点，属于两个不同的类别，我们希望找到一条直线尽可能地将两个类别分开。

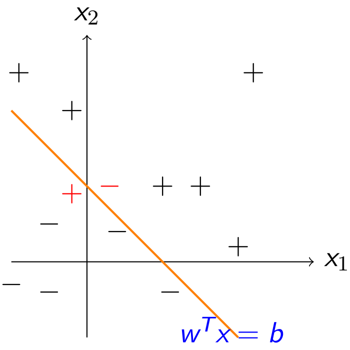


数据集 $\mathcal{D} = \{(x_1^{(1)}, x_2^{(1)}), \cdots, (x_1^{(n)}, x_2^{(n)})\}$
由超平面 $w^T x = b$ 组成线性分类器

$f(x) = sgn(w^T x - b)$

如何衡量分类器的好坏?

分类问题



True Positive:
$y = +1, f(x) = +1$
True Negative:
$y = -1, f(x) = -1$
False Positive:
$y = -1, f(x) = +1$
False Negative:
$y = +1, f(x) = -1$
准确率（accuracy）：$\frac{TP+TN}{n}$
错误率（error rate）：$\frac{FP+FN}{n}$

通常，我们希望分类器的准确率很高（错误率很低）

分类问题

准确率有时候并不能说明问题。比如检测仪器检测被试者是否携带某种病毒。

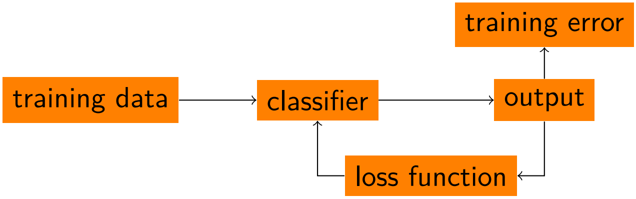
假设自然情况下只有 1% 的人携带某种病毒，如果检测仪器把所有的人都归类为未携带病毒，那么它的准确率将高达 99%！此时，我们需要另外两种指标

- 精确率（precision）：$TP/(TP+FP)$
- 召回率（recall）：$TP/(TP+FN)$

我们同样希望这两个数值足够高。在这个“犯懒”的检测仪器的例子里，precision 和 recall 都是 0。

基本概念

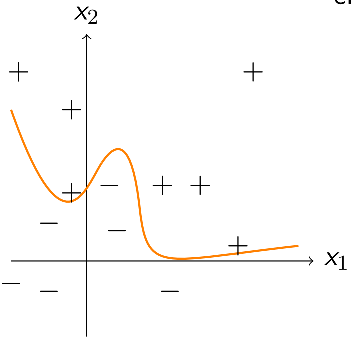
根据已有的样本训练得到好的分类器：

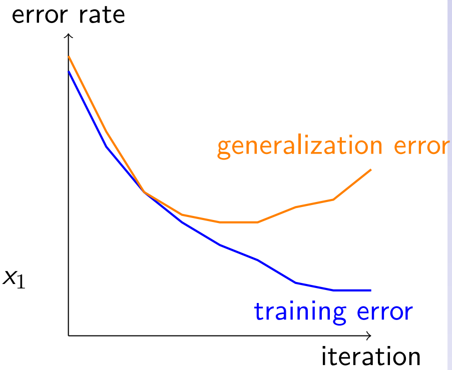


分类器在未遇到过的样本上的准确率更为重要，因此需要测试数据集。

- 测试数据集通常只占总数据的很小一部分
- 测试数据不参与训练
- 测试数据集上的误差通常可以代表泛化误差
- 测试误差通常高于训练误差

基本概念





过拟合通常是指分类器学到了样本中的噪声，不利于分类器的泛化。

多分类问题

ImageNet 数据集中含有 1000 个物体类别的图片，多分类算法将输出一个所有类别标签上的一个概率分布，如 $p_i(x)$ 表示分类器认为数据 x 属于第 i 类的把握程度，其中

$$\forall x, \sum_i p_i(x) = 1 \tag{56}$$

最终输出的标签是有最大把握的那个

$$y = \arg\max_i p_i(x) \tag{57}$$

回归问题

下图绿线是一正弦函数图像，加上一些噪声之后采样得到一系列数据点，我们能否找到一个合适的多项式函数

$$y(x, \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M \tag{58}$$

去拟合这些数据？

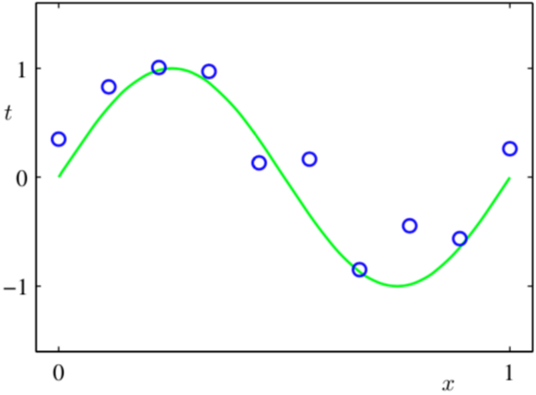


图 17: 待拟合数据

实例：最小二乘法

线性回归问题

在某个物理实验中测得四个数据点 (1,6)、(2,5)、(3,7)、(4,10)。需要找一条直线 $y = \beta_1 + \beta_2 x$，最能拟合这四个数据点。等价于解方程组

$$\begin{aligned} \beta_1 + 1\beta_2 &= 6 \\ \beta_1 + 2\beta_2 &= 5 \\ \beta_1 + 3\beta_2 &= 7 \\ \beta_1 + 4\beta_2 &= 10 \end{aligned}$$

方程组虽然无解，但是可以找到合适的 β_1 和 β_2 使得方程两边的误差最小。

线性回归问题

令 $\epsilon_i = y_i - (\beta_1 + \beta_2 x_i)$ 直觉上看，误差可以由等式两边差的绝对值之和衡量。但由于绝对值操作不可导，一般使用 2 范数的平方作为要优化的误差。即最小化

$$\begin{aligned} E &= \sum_{i=1}^{4} \epsilon_i^2 \\ &= [6-(\beta_1+1\beta_2)]^2 + [5-(\beta_1+2\beta_2)]^2 \\ &\quad +[7-(\beta_1+3\beta_2)]^2 + [10-(\beta_1+4\beta_2)]^2 \end{aligned}$$

联想到极值条件，该误差函数的最小值将出现在偏导都为 0 的时候。

线性回归问题

可得方程组

$$\frac{\partial E}{\beta_1} = 8\beta_1 + 20\beta_2 - 56 = 0$$

$$\frac{\partial E}{\beta_1} = 20\beta_1 + 60\beta_2 - 154 = 0$$

解得 $\beta_1 = 3.5, \beta_2 = 1.4$。