



# 《深度学习》花书

## 手推笔记（v1.1）--王博（Kings）

这个公式看起来很熟，却怎么也搞不懂怎么办？

作者：王博（Kings）、Sophia  
博士微信：**Kingsplus** （添加时请备注 学校/单位+专业）  
Github: <https://github.com/Sophia-11/DeepLearningNotes>  
(荣登趋势榜)，QQ 交流群：**1072190864**  
公众号【计算机视觉联盟】持续更新  
后台回复**【深度学习手推笔记】**可下载 pdf 打印版本



其他笔记：《机器学习手推笔记》、《无人驾驶手推笔记》、《SLAM 十四讲》  
请继续关注公众号【计算机视觉联盟】最新消息或 Github

## Update log

- 2020/01/02 \* - 更新深度学习符号
- 2020/01/20 \* - 更新第一章
- 2020/02/03 \* - 更新第二章
- 2020/02/16 \* - 更新第三章
- 2020/02/25 \* - 更新第四章
- 2020/03/04 \* - 更新第五章

## 第一部分 机器学习基础

- [第一章 前言](#)
- [第二章 线性代数](#)
- [第三章 概率论](#)
- [第四章 数值计算](#)
- [第五章 机器学习基础](#)

## 第二部分 深度神经网络核心知识

- [第六章 前馈神经网络](#)
- [第七章 正则化方法](#)
- [第八章 优化方法](#)
- [第九章 卷积神经网络](#)
- [第十章 循环神经网络](#)
- [第十一章 实战经验](#)
- [第十二章 深度学习应用](#)

## 第三部分 深度学习前沿研究

- [第十三章 线性因子模型](#)
- [第十四章 自编码器](#)
- [第十五章 表示学习](#)
- [第十六章 结构化概率模型](#)
- [第十七章 蒙特卡洛方法](#)
- [第十八章 配分函数](#)
- [第十九章 近似推断](#)
- [第二十章 生成模型](#)

# 深度学习

## 数学符号

### 1. 数和数组

a

标量

$\vec{a}$

向量

A

矩阵

A

张量

$I_n$

n行n列的单位矩阵

I

单位矩阵

$e^{(i)}$

标准基向量  $[0, \dots 0, 1, 0, \dots 0]$  第*i*处为1

$\text{diag}(a)$

对角矩阵

a

标量随机变量

a

向量随机变量

A

矩阵随机变量

### 2. 集合和图

A

集合

R

实数集

{0,1}

包含0和1的集合

{0,1, ..., n}

包含0和n之间所有整数的集合

[a,b]

a,b 定义空间

(a,b]

不包含a, 但包含b 的实数空间

A \ B

差集, 包含A不包含B

G

图

$\text{Par}_G(x_i)$

图 G 中  $x_i$  的父节点

【公众号  
计算机视觉联盟】

# 深度学习

## 数学符号

### 3. 索引

$a_i$ : 向量  $a$  的第  $i$  个元素，其中索引从 1 开始

$a_{:,i}$ : 除了第  $i$  个元素， $a$  的所有元素

$A_{i,j}$ : 矩阵  $A$  的  $i,j$  元素

$A_{:,i}$ : 矩阵  $A$  的第  $i$  行

$A_{i,j,k}$ : 3 维张量  $A$  的  $(i,j,k)$  元素

$A_{::,:i}$ : 3 维张量的 2 维切片

$a_i$ : 随机向量  $a$  的第  $i$  个元素

### 4. 线性代数中的操作

$A^T$ :  $A$  的转置

$A^+$ :  $A$  的 Moore-Penrose 假逆

$A \odot B$ :  $A$  和  $B$  的逐元素乘积 (Hadamard 乘积)

$\det(A)$ :  $A$  的行列式

### 5. 微积分

$\frac{dy}{dx}$ :  $y$  关于  $x$  的导数

$\frac{\partial y}{\partial x}$ :  $y$  关于  $x$  的偏导

$\nabla_x y$ :  $y$  关于  $x$  的梯度

$\nabla_x y$ :  $y$  关于  $X$  的矩阵导数

【公众号  
计算机视觉联盟】

# 深度学习

## 数学符号

### 5. 微积分

$$\nabla_x y$$

关于  $X$  求导后的张量

$$\frac{\partial f}{\partial x}$$

$f: \mathbb{R}^n \rightarrow \mathbb{R}^m$  的 Jacobian 矩阵  $J \in \mathbb{R}^{m \times n}$

$$\nabla_x^2 f(x) \text{ or } H(f)(x)$$

$f$  在点  $x$  处的 Hessian 矩阵

$$\int f(x) dx$$

$X$  整个域上的定积分

$$\int_S f(x) dx$$

集合  $S$  上关于  $x$  的定积分

### 6. 概率和信息论

$$a \perp b$$

$a$  和  $b$  相互独立的随机变量

$$a \perp b | c$$

给定  $c$  后条件独立

$$P(a)$$

离散变量上的概率分布

$$p(a)$$

连续变量上的概率分布

$$a \sim P$$

具有分布  $P$  的随机变量  $a$

$$E_{x \sim P}[f(x)] \text{ or } E[f(x)]$$

$f(x)$  关于  $P(x)$  的期望

$$\text{Var}(f(x))$$

$f(x)$  在分布  $P(x)$  下的方差

$$\text{Cov}(f(x), g(x))$$

$f(x)$  和  $g(x)$  在分布  $P(x)$  下的协方差

$$H(x)$$

随机变量  $x$  的香农熵

$$D_{KL}(P || Q)$$

$P$  和  $Q$  的 KL 故度

$$N(x; \mu, \Sigma)$$

均值为  $\mu$  协方差为  $\Sigma$ ,  $x$  上的高斯分布

【计算机视觉联盟】  
公众号

# 深度学习

## 数学符号

### 7. 函数

$f: A \rightarrow B$

定义域为  $A$  值域为  $B$  的函数  $f$

$f \circ g$

$f$  和  $g$  的组合

$f(x; \theta)$

由  $\theta$  参数化，关于  $x$  的函数。有时忽略  $\theta$  记为  $f(x)$

$\log x$

$x$  的自然对数

$s(x)$

Logistic sigmoid.  $\frac{1}{1 + \exp(-x)}$

$S(x)$

Softplus.  $\log(1 + \exp(x))$

$\|x\|_p$

$x$  的  $L^p$  范数

$\|x\|$

$x$  的  $L^\infty$  范数

$x^+$

$x$  的正数部分，即  $\max(0, x)$

$1_{\text{condition}}$

如果条件为真则为 1，否则为 0

### 8. 数据集和分布

$P_{\text{data}}$

数据生成分布

$\hat{P}_{\text{train}}$

由训练集定义的经验分布

$X$

训练样本的集合

$x^{(i)}$

数据集的第  $i$  个样本（输入）

$y^{(i)} \text{ or } \hat{y}^{(i)}$

监督学习中与  $x^{(i)}$  关联的目标

$X$

$m \times n$  矩阵，其中  $X_{i,:}$  为输入样本  $x^{(i)}$

【计算机视觉联盟】  
公众号

# 深度学习

## 第一章 引言

让计算机从经验中学习，并根据层次化概念来理解世界。从经验获取知识，可避免由人类来给计算机形式化地指定所需知识。

层次化的概念让计算机构建简单的概念来学习复杂概念。

这些概念建立在彼此之上的图，很“深”，称为 AI 深度学习  
*deep learning*

### 一些时间线：

IBM 的 Deep Blue 1997 年国际象棋打败世界冠军 Garry Kasparov

抽象和形式化任务对人类而言最为困难，对计算机却很容易  
人工智能的挑战：如何将这些非形式化的知识传达给计算机

### 知识库方法 (knowledge base)：

将世界的知识用形式化的语言进行硬编码 (hard code)  
计算机使用逻辑推理规则来自动理解形式化语言中的声明。

最著名的项目是 Cyc (1989)。Cyc 包含一个推断引擎  
和一个使用 CycL 语言描述的声明数据库。声明是人类监督的。  
缺点：这是一个笨拙的过程，无法设计出足够的复杂的形式化  
规则来精确描述世界。它的推理引擎可能会有不一致性。

### 机器学习 machine learning

依靠硬编码知识体系面对的困难表明，AI 系统需要具备  
自己获取知识的能力。

从原始数据提取模式的能力。

公众号  
【计算机视觉党联盟】

# 深度学习

公众号

【计算机视觉联盟】

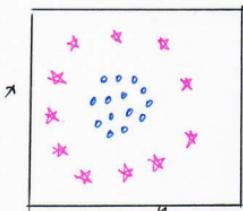
## 第一章 引言

引入机器学习使计算机能够解决涉及现实世界知识的问题，并能做出有理的主观的判断。

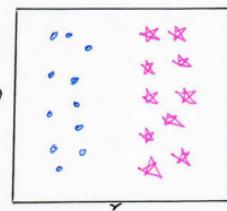
逻辑回归 logistic regression 决定做不做某事

朴素贝叶斯 naive Bayes 可分为垃圾及合法邮件

简单的机器学习算法的性能很大程度上依赖于数据的表示 (representation)



笛卡尔坐标表示



极坐标表示

不同的表示方式将会影响算法的性能。

许多人工智能任务可通过以下方式解决：选取一个合适的特征集，将这些特征提供给简单的机器学习算法。

然而，对于许多任务而言，很难知道应该提取哪些特征。比如车辆可能受光照影响不好表示出提取出特征。

表示学习 representation learning：使用机器学习来发现表示本身，而不仅仅把表示映射到输出。学习到的表示往往比手动设计表现得更好。

典型例子：自编码器 (autoencoder)。

自编码器由一个编码器 encoder 和一个解码器 decoder 组成

# 深度学习

## 第一章 引言

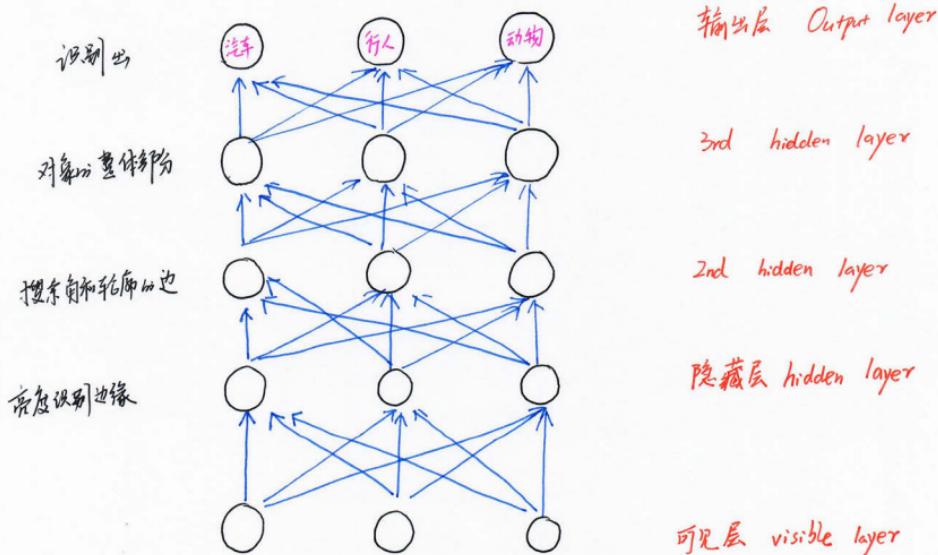
当设计特征或设计用于学习特征的算法时，目标通常是分离出能够解释观察数据的变差因素 (factor of variation)

许多现实任务中，困难在于可能多个变差因素同时影响每一个观察数据。虽然，从原始数据中提取高层次、抽象的特征非常困难。

深度学习 deep learning 通过其它简单的表示来表达复杂表示，解决了表示学习中的核心问题。

公众号

【计算机视觉联盟】



深度学习模型示意图

深度学习模型典型的例子是前馈深度网络或多层感知机  
(multilayer perceptron, MLP)

多层感知机仅仅是将一组输入值映射到输出值的数学函数。  
该函数由许多简单函数组合而成。

# 深度学习

## 篇章引言

学习数据的正确表示的想法是解释深度学习的一个视角。

另一个视角是深度促使计算机学习一个多层次的计算机程序。

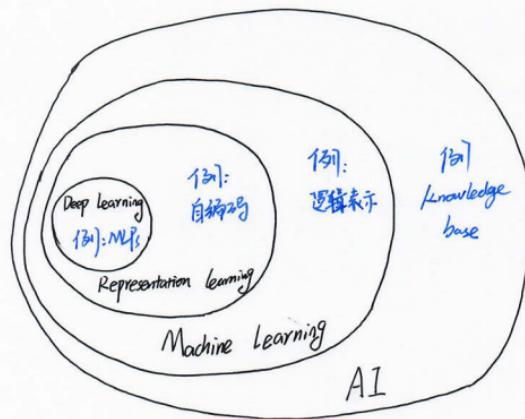
目前主要有两种度量模型深度的方式：第一种是基于评估架构所需执行的顺序指令的数目。将流程图最长路径视为深度。

取决于计算步骤的定义。

使用加(减)、乘、逻辑作为元素，模型深度为3。

将逻辑回归视为元素本身，那么这个深度为1。

另一种是在深度概率模型中使用的方法，将描述概念彼此如何关联的图深度视为模型深度。

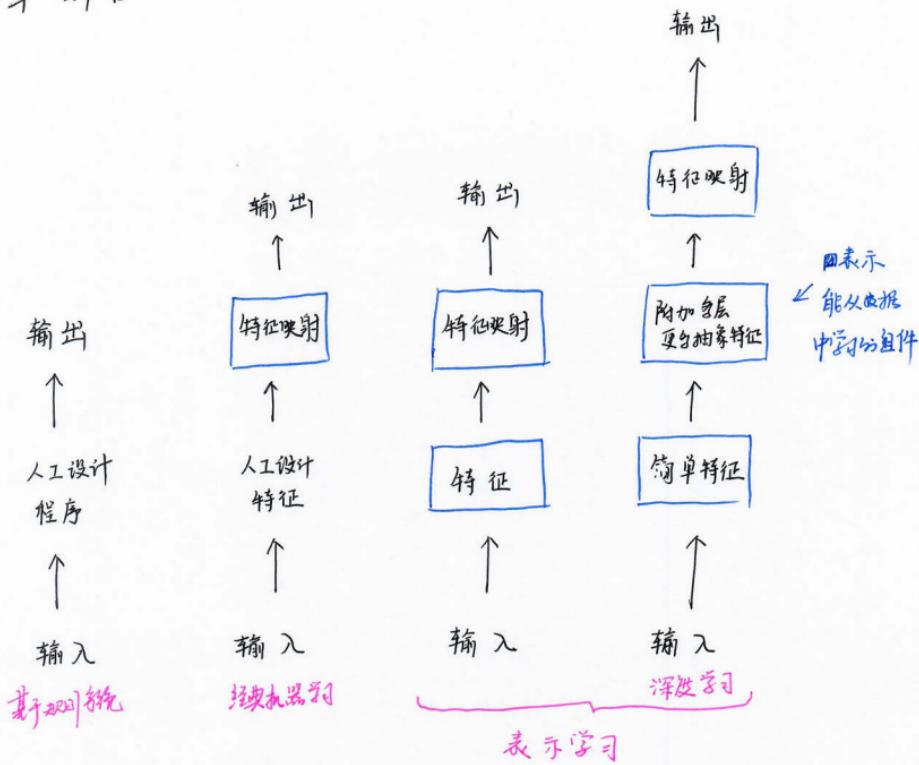


深度学习是一种表示学习  
也是一种机器学习  
可用于许多AI方法

公众号  
【计算机视觉联盟】

# 深度学习

## 第一章 引言



## 1.1 本书面向的读者

- ① 数学工具和机器学习概念
- ② 最成熟 的深度学习算法
- ③ 层层深入

# 深度学习

## 第一章 引言

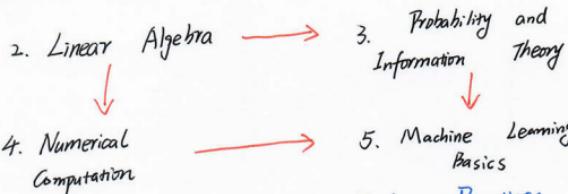
### 1.1 本书面向的读者

公众号

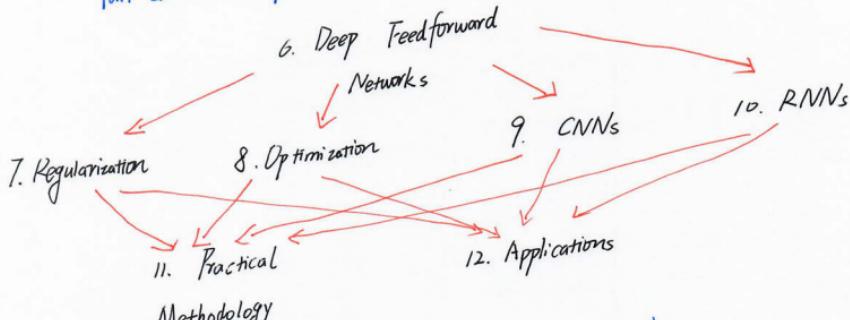
【计算机视觉联盟】

#### 1. Introduction

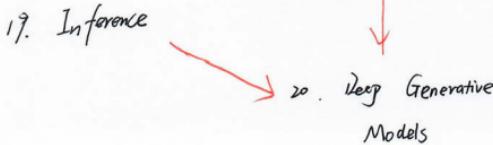
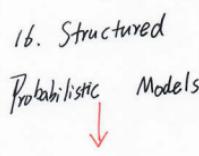
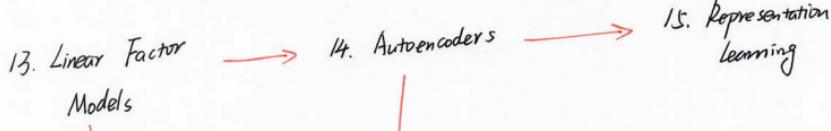
#### Part I : Applied Math and Machine learning Basics



#### Part II : Deep Networks: Modern Practices



#### Part III : Deep learning Research



# 深度学习

## 第一章 引言

### 1.2 深度学习的历史趋势

#### 1.2.1 神经网络的众多名称和命运变迁

深度学习历经三次发展浪潮：

20世纪40~60年代 雏形 控制论 cybernetics

20世纪80~90年代 联结主义 connectionism

2006年 才真正以深度学习之名复兴

最早的一些学习算法，旨在模拟生物学习计算模型。  
即 人工神经网络 (artificial neural network, ANN)

深度学习的神经观点受两个主要思想启发：

一个是大脑作为例子证明智能行为是可能的，逆向大脑计算原理。

一个是理解大脑和人类智能背后的原理也非常有趣。

二是理解大脑和人类智能背后的原理也非常有趣。

现代术语“深度学习”超越了目前机器学习模型的神经科学观点。

诉诸于多层次组合。

现代深度学习最早前身是从神经科学角度出发的简单线性模型。

输出  $f(x, w) = x_1w_1 + \dots + x_nw_n$  第一波神经网络称为控制论  
输入 权重.

McCulloch-Pitts 神经元 1943 是脑功能最早期模型  
该线性模型通过逻辑函数  $f(x, w)$  区分两种不同类别 输入。

公众号

【计算机视觉联盟】

# 深度学习

公众号

【计算机视觉联盟】

## 第一章 引言

### 1.2.1 神经网络名称命运变迁

20世纪50年代，感知机 (Rosenblatt, 1956, 1958) 成为第一个能根据每个类别所输入样本来学习权重模型。

自适应线性单元 (adaptive linear element, ADALINE) 简单地返回函数  $f(x)$  本身的值来预测一个实数，还可以学习从数据中叫这些数。

用于调节 ADALINE 权重的训练算法被称为随机梯度下降  
*stochastic gradient descent*  
而一种特例。

基于感知机和 ADALINE 中使用的函数  $f(x, w)$  模型  
称为线性模型 (linear model)，但是目前最广泛使用的机器学习模型

线性模型的局限性：

最著名的是无法学习异或函数，即  
 $\begin{cases} f([0, 1], w) = 1 & \text{和 } f([1, 0], w) = 1 \\ f([1, 1], w) = 0 & \text{和 } f([0, 0], w) = 0 \end{cases}$   
1969年抵触，热潮大衰退

神经科学给了我们依靠单一深度学习算法解决许多不同任务的理由。

新认知机 (1980) 引入一个处理图片的强大模型架构，成为卷积网络的基础 (1998 LeCun)。

目前大多数神经网络是基于整流线性单元 (rectified linear unit) 神经单元模型。

不应该认为深度学习在模拟大脑，而是借鉴。

别忘了还有一个学科 “计算神经科学”

# 深度学习

## 第一章 引言

### 1.2.1 神经网络的名称和历史变迁

20世纪80年代，神经网络第二次浪潮：

联结主义 (connectionism) . 并行分布处理 (parallel distributed processing)

联结主义思想是当网格将大量简单的计算单元连接在一起时可实现的智能行为。

分布式表示 (distributed representation) (Hinton 1986)

思想：系统每一个输入都应该由多个特征表示，并且每一个特征都应该参与许多可能输入的表示。

联结主义潮流的一个重要成就是：反向传播在训练具有内部表示的深度神经网络中的成功使用以及反向传播算法的普及。

Hochreiter and Schmidhuber (1997) 引入长短期记忆 (long short-term memory LSTM)

浪潮持续到上世纪90年代中期，一直到2007年

神经网络第三次浪潮始于 2006 年

Hinton 2006 年，深度信念网络的神经网络可使用一种称为自下而上逐层反训练的策略来有效地训练。

### 1.2.2 与日俱增的数据量

MNIST 数据集 (Modified National Institute of Standards and Technology)

国家标准和技术研究所

公众号

【计算机视觉联盟】

# 深度学习

## 第一章 引言

### 1.2.3 与日俱增的模型规模

公众号

【计算机视觉联盟】

1. 感知机 Rosenblatt, 1958, 1962
2. 自适应线性单元 Widrow and Hoff 1960
3. 神经认知机 Fukushima, 1980.
4. 早期前向传播网络 Rumelhart et al. 1986 b.
5. 用于语音识别的循环神经网络 Robison and Fallside, 1991
6. 用于语言识别的多层感知机 Bengio et al. 1991
7. 均匀加权 sigmoid 信息网络 Saul et al. 1996
8. LeNet-5 LeCun 1998
9. 固声状态网络 Jaeger and Haas 2004
10. 深度信息网络 Hinton et al. 2006 a
11. GPU - 加速卷积网络 Chellapilla 2006
12. 深度玻尔兹曼机 Salakhutdinov, Hinton 2009
13. GPU - 加速深度信息网络 Raina 2009
14. 无监督卷积网络 Jarrett 2009
15. GPU - 加速多层感知机 Ciregan 2010
16. OMPI 网络 Coates and Ng 2011
17. 分布式自编码器 Le et al 2012
18. Multi-GPU 卷积网络 Krizhevsky 2012
19. COTS HPC 无监督卷积网络 Coates 2013
20. GoogleNet Szegedy 2014

## 应用数学与机器学习基础

## 第二章 线性代数

## 2.1 标量、向量、矩阵和张量

标量 scalar : 一个标量就是一个单独的数

如定义实数标量时,  $s \in \mathbb{R}$

定义自然数标量时,  $n \in \mathbb{N}$

向量 vector : 一个向量是一列数, 有序的.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (2.1)$$

有时需要索引向量中的一些元素, 比如  $x_1, x_3, x_6$ ,

定义一个包含索引元素的集合  $S = \{1, 3, 6\}$  记  $\mathbf{x}_S$

定义一个包含索引元素的集合  $S = \{1, 3, 6\}$  记  $\mathbf{x}_{-S}$

$\mathbf{x}_{-1}$  去除  $x_1$  外所有元素  $\mathbf{x}_{-S}$  去除  $x_1, x_3, x_6$  外所有元素

矩阵 matrix : 矩阵是一个二维数组.

比如一个实数矩阵高为  $m$ , 宽为  $n$ ,  $A \in \mathbb{R}^{m \times n}$

$A_{i,:}$  表 i 行,  $A_{:,i}$  表 i 列

$$\begin{bmatrix} A_{0,0} & A_{0,1} & \dots \\ A_{1,0} & A_{1,1} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix} \quad (2.2)$$

$f(A)_{ij}$  表示函数  $f$  作用在  $A$  上输出的矩阵第  $i$  行第  $j$  列元素

超过二维的数组. 一个数组中的元素分布在若干维坐标规则网格中, 称为张量.

张量  $A$  中坐标  $(i,j,k)$  元素记为  $A_{i,j,k}$ .

# 深度学习

公众号

【计算机视觉联盟】

## 第二章

### 2.1 标量、向量、矩阵和张量

$$\text{转置 transpose: } (A^T)_{ij} = A_{j,i} \quad (2.3)$$

向量可看作只有一列的矩阵。例如  $x = [x_1, x_2, x_3]^T = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

标量的转置等于本身:  $a = a^T$

注意一个:  $C = A + b$  矩阵 = 矩阵 + 向量

$$C_{ij} = A_{ij} + b_j \quad \text{与每一行相加}$$

### 2.2 矩阵和向量相乘

$$C_{m \times p} = A_{m \times n} B_{n \times p} \quad (2.4)$$

↓ 定义

$$C_{ij} = \sum_k A_{ik} B_{kj} \quad (2.5)$$

两矩阵对应元素相乘称为元素对应乘积 (element-wise product) 或 Hadamard 乘积, 记为  $A \odot B$

两个相同维数向量  $x$  和  $y$  的点积 (dot product) 可记为  $x^T y$ .  
 $C_{ij} = \sum_k A_{ik} B_{kj}$  可看作  $A$  第  $i$  行与  $B$  第  $j$  列点积

分配律:

$$A(B+C) = AB + AC \quad (2.6)$$

结合律:

$$A(BC) = (AB)C \quad (2.7)$$

两向量点积满足交换律, 矩阵不满足交换律

$$x^T y = y^T x \quad (2.8)$$

# 深度学习

## 第二章

### 2.2 矩阵和向量相乘

公众号

【计算机视觉联盟】

矩阵乘积的转置：

$$(AB^T) = B^TA^T \quad (2.9)$$

两向量点积结果是标量，标量转置是自身

$$xy = (x^Ty)^T = y^Tx \quad (2.10)$$

线性方程组：

$$Ax = b \quad (2.11)$$

$A \in \mathbb{R}^{m \times n}$  已知矩阵  
 $x \in \mathbb{R}^n$  未知  
 $b \in \mathbb{R}^m$  已知向量

矩阵的一行和  $b$  中对各元素构成一个约束

$$\left\{ \begin{array}{l} A_{1,:} x = b_1 \\ A_{2,:} x = b_2 \\ \dots \\ A_{m,:} x = b_m \end{array} \right. \quad (2.12) \quad (2.13) \quad (2.14) \quad (2.15)$$

$$\left\{ \begin{array}{l} A_{1,1}x_1 + A_{1,2}x_2 + \dots + A_{1,n}x_n = b_1 \\ A_{2,1}x_1 + A_{2,2}x_2 + \dots + A_{2,n}x_n = b_2 \\ \dots \\ A_{m,1}x_1 + A_{m,2}x_2 + \dots + A_{m,n}x_n = b_m \end{array} \right. \quad (2.16) \quad (2.17) \quad (2.18) \quad (2.19)$$

# 深度学习

公众号

【计算机视觉联盟】

## 第二章

### 2.3 单位矩阵和逆矩阵

单位矩阵 identity matrix. 注意向量和单位矩阵相乘都不变.

$$I_n \in \mathbb{R}^{n \times n}$$

$$\forall x \in \mathbb{R}^n, I_n x = x \quad (2.20)$$

所有对角线元素都是 1. 其它为 0.

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

矩阵 A 的逆 (matrix inversion)  $A^{-1}$ , 定义如下:

$$A^{-1} A = I_n \quad (2.21)$$

如何求解?

$$Ax = b \quad (2.22)$$

$$A^{-1} A x = A^{-1} b \quad (2.23)$$

$$I_n x = A^{-1} b \quad (2.24)$$

$$x = A^{-1} b \quad (2.25)$$

### 2.4 线性相关和生成子空间

$Ax = b$  中 x 的解要么没有, 要么 1 个, 要么无数个, 因为如果 x 和 y 都是方程组解

$$z = \alpha x + (1-\alpha)y \quad (2.26)$$

z 也是该方程组的解

线性组合:  $Ax = \sum_i x_i A_{:,i} \quad (2.27)$

一组向量的线性组合, 是指每个向量乘以对应坐标系数之后的和, 即:

$$\sum_i c_i v^{(i)} \quad (2.28)$$

生成子空间 span.

# 深度学习

## 第二章

### 2.4 线性相关和生成子空间

确定  $Ax = b$  是否有解相当于向量  $b$  是否在  $A$  列向量生成子空间中。  
这个特征的生成子空间被称为  $A$  的列空间 (column space)  
 $A$  的值域 (range)

要想唯一解，矩阵方阵 (square) 所有列向量无关。

列向量线性相关的矩阵称为奇异的 (singular)

矩阵右乘：

$$AA^{-1} = I \quad (2.29)$$

公众号

【计算机视觉联盟】

### 2.5 范数

衡量一个向量的大小

$L^p$  范数定义：  $\|x\|_p = \left( \sum_i |x_i|^p \right)^{\frac{1}{p}}$  (2.30)

$p \in R, p \geq 1$

范数是将向量映射到非负值的函数。直观而言，范数衡量从原点到点  $x$  的距离。

范数是满足下列性质的函数  $\left\{ \begin{array}{l} f(x) = 0 \Rightarrow x = 0 \\ f(x+y) \leq f(x) + f(y) \quad \text{三角不等式} \\ \forall \alpha \in R, f(\alpha x) = |\alpha| f(x) \end{array} \right.$

当  $p=2$  时， $L^2$  范数被称为欧几里得范数 Euclidean norm

$L^2$  范数简化为  $\|x\|$ ，忽略下脚标 2。

平方  $L^2$  范数也常用来衡量大小可通过点积  $x^T x$  计算

# 深度学习

公众号

【计算机视觉联盟】

## 2.5 范数

$L^1$  范数简化如下：

$$\|x\|_1 = \sum_i |x_i| \quad (2.31)$$

适用场景：零和非零元素之间差异非常重要。每当  $x$  中某个元素从 0 增加  $\epsilon$ ，对应  $L^1$  也会增加  $\epsilon$

同时  $L^1$  范数经常作为表示非零元素数目的一种代价函数。

$L^\infty$  称为 最大范数 max norm. 最大元素绝对值

$$\|x\|_\infty = \max_i |x_i| \quad (2.32)$$

衡量矩阵大小，使用 Frobenius 范数：

$$\|A\|_F = \sqrt{\sum_{i,j} A_{i,j}^2} \quad (2.33)$$

类似于向量的  $L^2$  范数

两向量点积也用范数表示：

$$x^T y = \|x\|_2 \|y\|_2 \cos \theta \quad (2.34)$$

$x$  和  $y$  之间夹角

## 2.6 特殊类型的矩阵和向量

对角矩阵

$$D_{i,j} = 0.$$

diag( $v$ ) 表示向量  $v$  中元素给定的对角矩阵。

$$\text{diag}(v) = \text{diag}[1/v_1, \dots, 1/v_n]^T$$

当且仅当对角都非零  $\text{diag}(v)^{-1} = \text{diag}[1/v_1, \dots, 1/v_n]$ 。  
不是所有对角矩阵都是方阵。非方阵的对角矩阵没有逆矩阵。

对称

$$A = A^T \quad (A_{i,j} = A_{j,i})$$

(2.35)

单位向量是具有单位范数的向量。

$$\|x\|_2 = 1 \quad (2.36)$$

在  $R^n$  中，至多有  $n$  个范数非零向量互相正交，不仅互相正交，范数都为 1。  
即标准正交

## 2.7 特征分解

特征分解 (eigen decomposition) 将矩阵分解为一组特征向量和特征值

方阵  $A$  的特征向量 (eigenvector) 指与  $A$  相乘后相当于对该向量进行缩放的非零向量  $v$ .

$$A v = \lambda v \quad (2.39)$$

↓  
特征值.

右特征向量，若  $V^T A = \lambda V^T$  定义为左特征向量

假设矩阵  $A$  有  $n$  个线性无关的特征向量  $\{v^{(1)}, \dots, v^{(n)}\}$

对应着特征值  $\{\lambda_1, \dots, \lambda_n\}$

将特征向量连接成一个矩阵，每一列是一个特征向量。 $V = [v^{(1)}, \dots, v^{(n)}]$

同理，将特征值连成一个向量  $\Lambda = [\lambda_1, \dots, \lambda_n]^T$

$A$  的特征分解：

$$A = V \operatorname{diag}(\Lambda) V^{-1} \quad (2.40)$$

不是每一个矩阵都可分解成特征值、特征向量。

具体而言，每个实对称矩阵都可分解成实特征向量和实特征值。

$$A = Q \Lambda Q^{-1} \quad (2.41)$$

↓  
正交矩阵      ↓ 对称矩阵

正定 positive definite      特征值都是正数       $x^T A x = 0 \Rightarrow x = 0$

半正定 positive semidefinite      都是非负数

负定 negative definite      都是负数

半负定 negative semidefinite      都是非正数。 $Bx, x^T A x \geq 0$

## 2.8 奇异值分解

奇异值分解 (singular value decomposition, SVD)

奇异向量 singular vector. 奇异值 singular value

每个实数矩阵都有一个奇异值分解，但不一定有特征分解。

非方阵的矩阵没有特征分解，只能使用奇异值分解

特征分解：

$$A = V \text{diag}(\lambda) V^{-1} \quad (2.42)$$

奇异值分解：

$$A = U D V^T \quad (2.43)$$

↓ 改  
 ↓ 对角  
 ↓ 改  
 U的列向量 左奇异向量

V的列向量 右奇异向量

U的列向量 左奇异向量

$A$ 的左奇异向量 ( $U$ 的列向量) 是  $AA^T$  的特征向量

$A$ 的右奇异向量 ( $V$ 的列向量) 是  $A^TA$  的特征向量。

$A$ 的非零奇异值是  $A^TA$  特征值的平方根，也是  $AA^T$  特征值的平方根。

## 2.9 Moore-Penrose 伪逆

对于非方矩阵而言，逆矩阵没有定义。

希望通过矩阵  $A$  的左逆  $B$  来求解线性方程：

$$Ax = y \quad (2.44)$$

↓ 左乘左逆  $B$

$$x = By \quad (2.45)$$

## 2.9 Moore-Penrose 伪逆

Moore-Penrose pseudoinverse 矩阵A的伪逆定义为：

$$A^+ = \lim_{\alpha \rightarrow 0} (A^T A + \alpha I)^{-1} A^T \quad (2.46)$$

实际伪逆 | 计算公式

$$A^+ = V D^+ U^T \quad (2.47)$$

$U, D, V$  都是A奇异值分解的矩阵。

对前矩阵  $D$  的伪逆  $D^+$  是其非零元素取倒数之后再转置得训

当列数 > 行数，使用伪逆求得是众多解法中的一种。

特别地， $x = A^+ y$  是方程所有可行解中范数得最小  $\|x\|_2$ ，最小的一个。

当行数 > 列数，可能没有解。

通过伪逆得到的  $x$  使得  $Ax$  和  $y$  的欧几里得距离  $\|Ax - y\|_2$  最小

## 2.10 迹运算

迹运算返回矩阵对角元素之和

$$\text{Tr}(A) = \sum_i A_{i,i} \quad (2.48)$$

迹运算提供了另一种描述矩阵 Frobenius 范数方式

$$\|A\|_F = \sqrt{\text{Tr}(A^T A)} \quad (2.49)$$

迹运算一些特殊公式：

$$\text{Tr}(A) = \text{Tr}(A^T) \quad (2.50)$$

即使  $ACR^{m \times n}, BER^{n \times m}$   
可得：

$$\text{Tr}(ABC) = \text{Tr}(CAB) = \text{Tr}(BCA) \quad (2.51)$$

$$\text{Tr}(AB) = \text{Tr}(BA) \quad (2.53)$$

$$\text{Tr}\left(\prod_{i=1}^n F^{(i)}\right) = \text{Tr}\left(F^{(n)} \prod_{i=1}^{n-1} F^{(i)}\right) \quad (2.52)$$

恒等  $a = \text{Tr}(a)$

# 深度学习

2.12 实例]：主成分分析

公众号

【计算机视觉联盟】

principal components analysis , PCA

假设空间中  $R^n$  有  $m$  个点  $\{x^{(1)}, \dots, x^{(m)}\}$ , 我们希望有损压缩.

一种编码这些点用低维表示:

对于每个点  $x^{(i)} \in R^n$ , 会对应一个编码向量  $c^{(i)} \in R^l$ , 如果  $l < n$ , 则压缩.

我们希望找到一个编码函数  $f(x) = c$

一个解码函数  $x \approx g(f(x))$

PCA 由选择的解码函数决定

为简化, 使用矩阵乘法将编码映射回  $R^n$ . 即  $g(c) = Dc$

$D \in R^{n \times l}$  解码矩阵

而  $g(c) = Dc$  可看: 可能有多个解.

比如按比例缩放  $c_i$ , 只需按比例放大  $D_{:, i}$ .

为使唯一解, 限制  $D$  中所有列向量都有单位范数.

为使更简单 PCA 限制  $D$  的列向量彼此正交

如何根据每一个输入得到一个最优编码  $c^*$ ?

一种方式是最大化原始输入  $x$  和重构向量  $g(c^*)$  之间的距离.

PCA 中, 使用  $L^2$  范数.

$$c^* = \underset{c}{\operatorname{argmin}} \|x - g(c)\|_2 \quad (2.54)$$

↓ 用平方替代

$$c^* = \underset{c}{\operatorname{argmin}} \|x - g(c)\|_2^2 \quad (2.55)$$

# 深度学习

公众号

【计算机视觉联盟】

2.12 实例：主成分分析

最小化函数可简写为：

$$(x - g(c))^T(x - g(c)) \quad (2.56)$$



$$= x^T x - x^T g(c) - g(c)^T x + g(c)^T g(c) \quad (2.57)$$



$$= \underbrace{x^T x}_{\uparrow} - 2x^T g(c) + g(c)^T g(c) \quad (2.58)$$

不依赖c，可忽略。

优化目标： $c^* = \underset{c}{\operatorname{argmin}} \left[ -2x^T g(c) + g(c)^T g(c) \right] \quad (2.59)$

代入  $g(c)$  定义， $g(c) = Dc$

$$c^* = \underset{c}{\operatorname{argmin}} -2x^T Dc + c^T D^T Dc \quad (2.60)$$

$$= \underset{c}{\operatorname{argmin}} -2x^T Dc + \underbrace{c^T I_L c}_{\uparrow} \quad (2.61)$$

考虑矩阵D 正交和单位行数约束， $I_L$  是单位矩阵

$$= \underset{c}{\operatorname{argmin}} -2x^T Dc + c^T c \quad (2.62)$$

采用向量微积分

$$\nabla_c (-2x^T Dc + c^T c) = 0 \quad (2.63)$$

$$-2D^T x + 2c = 0 \quad (2.64)$$

$$c = D^T x \quad (2.65)$$

# 深度学习

2.12 实例：主成分分析

公众号

【计算机视觉联盟】

(四)

$$C = D^T X$$

编码函数  $f(x) = C$ ,    解码函数  $x \approx g(f(x))$

最优编码  $x$  只需一个矩阵-向量乘法操作，则编码函数：

$$f(x) = D^T x \quad (2.66)$$

进一步定义重构操作：

$$r(x) = g(f(x)) = DD^T x \quad (2.67)$$

此时需要挑选编码矩阵  $D$  即可。

回顾最小化输入和重构之间  $L^2$  距离想法。

必须最小化所有维数 和 所有点上误差矩阵 Frobenius 范数：

$$D^* = \arg \min_D \sqrt{\sum_{i,j} (x_j^{(i)} - r(x_j^{(i)}))^2} \quad \text{subject to } D^T D = I_L \quad (2.68)$$

每个点都要考虑。

首先考虑  $L=1$  情况，也就是编码向量是 1 维的。 $D$  简化为单一向量  $d$ 。

$$d^* = \arg \min_d \sum_i \| x^{(i)} - d x^{(i)} \|_2^2 \quad \text{subject to } \| d \|_2 = 1 \quad (2.69)$$

$\Downarrow$

$$d^* = \arg \min_d \sum_i \| x^{(i)} - d^T x^{(i)} d \|_2^2 \quad \text{subject to } \| d \|_2 = 1 \quad (2.70)$$

$\Downarrow$  条件极值取值和自身相等。 $[d^T x^{(i)}]^T = x^{(i)} d^T$

$$d^* = \arg \min_d \sum_i \| x^{(i)} - x^{(i)^T} d d \|_2^2 \quad \text{subject to } \| d \|_2 = 1 \quad (2.71)$$

# 深度学习

## 2.12 实例：主成分分析

将各点的向量堆叠成一个矩阵， $X \in \mathbb{R}^{m \times n}$ ，其中  $X_{i,:} = x^{(i)}^T$

公众号

(2.71) 可重写：

$$d^* = \arg \min_d \|X - Xdd^T\|_F^2 \quad \text{subject to } d^T d = 1 \quad (2.72)$$

增加约束。↓ 沿 Frobenius 范数简化为：

$$\arg \min_d \|X - Xdd^T\|_F^2 \quad (2.73)$$

$$= \arg \min_d \text{Tr} ((X - Xdd^T)^T (X - Xdd^T)) \quad (2.74)$$

$$= \arg \min_d \text{Tr}(X^T X - X^T X dd^T - dd^T X^T X + dd^T X^T X dd^T) \quad (2.75)$$

$$= \arg \min_d \underbrace{\text{Tr}(X^T X)}_{与 d 无关} - \text{Tr}(X^T X dd^T) - \text{Tr}(dd^T X^T X) + \text{Tr}(dd^T X^T X dd^T) \quad (2.76)$$

$$= \arg \min_d -\text{Tr}(X^T X dd^T) - \text{Tr}(dd^T X^T X) + \text{Tr}(dd^T X^T X dd^T) \quad (2.77)$$

$$= \arg \min_d -2\text{Tr}(X^T X dd^T) + \text{Tr}(dd^T X^T X dd^T) \quad (2.78)$$

$$= \arg \min_d -2\text{Tr}(X^T X dd^T) + \text{Tr}(X^T X dd^T dd^T) \quad (2.79)$$

此时，参数约束条件  $d^T d = 1$

$$= \arg \min_d -2\text{Tr}(X^T X dd^T) + \text{Tr}(X^T X dd^T) \quad \text{subject to } d^T d = 1$$

$$= \arg \min_d -\text{Tr}(X^T X dd^T) \quad \text{subject to } d^T d = 1$$

$$= \arg \max_d \text{Tr}(X^T X dd^T) \quad \text{subject to } d^T d = 1$$

=  $\arg \max_d \text{Tr}(X^T X dd^T) \rightarrow$  增加的  $d$  是  $X^T X$  最大特征值对应的特征向量

【计算机视觉联盟】

# 深度学习

公众号

【计算机视觉联盟】

## 第三章 概率与信息论

### 3.1 为什么要使用概率？

机器学习通常必须处理不确定量，有时也需要处理随机（非确定性）量。

不确定性有三种可能的来源：

1. 被建模系统内在的随机性

2. 不完全观测

3. 不完全建模

两种情况：

1. 扑克牌抽出一手特定的牌，可计算概率  $P$ ，反复实验无限次，有  $P$  的比例可导致这样的结果。与事件发生频率相联系，称为频率派概率 (frequentist probability)

2. 然而，看病时个体，患流感概率  $40\%$ ，这种概率来表示信任度。  
确定性水平，称为贝叶斯概率 (Bayesian probability)

默认的一点：将贝叶斯概率和频率派概率视为等同

### 3.2 随机变量

随机变量 (random variable) 是可以随机地取不同值的变量。

就其本身而言，一个随机变量只是对可能的状态的描述

它必须伴随着一个概率分布来指定每个状态的可能性

随机变量是离散或者连续的。

离散随机变量拥有有限或可数无限多的状态，不一定是整数。  
也可能是一些命名状态而没有数值。

连续随机变量伴随着实数值

# 深度学习

## 3.3 概率分布

概率分布 probability distribution 用来描述随机变量或一族随机变量在每一个可能取到的状态  $x$  大小。

### 3.3.1 离散型变量和概率质量函数

离散型变量的概率分布可用概率质量函数 (probability mass function, PMF)

$$x = x_i, P(x) \text{ 概率为 1 肯定} \quad \left. \begin{array}{l} \text{为 0 一定不} \\ \end{array} \right\} x \sim P(x)$$

多个变量的概率分布被称为联合概率分布 (joint probability distribution)

$$P(x=x_i, y=y_j) \text{ 简写 } P(x, y).$$

如果一个函数  $P$  是随机变量  $x$  的 PMF, 必须满足条件:

- ①  $P$  定义域必须是  $x$  所有可能状态的集合
- ②  $\forall x \in X, 0 \leq P(x) \leq 1$ . 不可能概率为 0, 一定为 1
- ③  $\sum_{x \in X} P(x) = 1$ , 把这条性质称为归一化 (normalized)

假设考虑一个高维空间随机变量  $x$  有  $k$  个不同状态。假设  $x$  均匀分布 uniform distribution.

通过将它 PMF 设为

$$P(x=x_i) = \frac{1}{k} \quad (3.1)$$

因为  $k$  是一个正整数, 所以  $\frac{1}{k}$  是正的.

$$\sum_i P(x=x_i) = \sum_i \frac{1}{k} = \frac{k}{k} = 1 \quad (3.2)$$

公众号

【计算机视觉联盟】

## 3.3.2 连续型变量和概率密度函数

概率密度函数 (probability density function, PDF)

$P$  是概率密度函数  $\left\{ \begin{array}{l} \text{① } P \text{ 的定义域必须是 } X \text{ 所有可能状态的集合} \\ \text{② } \forall x \in X, p(x) \geq 0, \text{ 并不要求 } p(x) \leq 1 \\ \text{③ } \int p(x) dx = 1 \end{array} \right.$

概率密度函数  $p(x)$  并没有直接对特定的状态给出概率. 相反的, 它给出了落在面积为  $\delta x$  的无限小分区域内的概率为  $p(x)\delta x$ .

以一个连续型随机变量的 PDF 例子, 考虑实数区间上均匀分布

$$u(x; a, b)$$

为确保区间外没有概率  $x \notin [a, b], \quad u(x; a, b) = 0$

$$[a, b], \quad u(x; a, b) = \frac{1}{b-a}$$

任何一点都非负, 积分为 1. 用  $x \sim U(a, b)$  表示  $x$  在  $[a, b]$  上是均匀分布

## 3.4 边缘概率

假如知道了一组联合概率分布, 想知道其中一个子集的概率分布, 这种定义在子集上的概率分布称为 边缘概率分布 marginal probability distribution

已知  $P(x, y)$ , 根据求和法则求  $P(x)$

$$\forall x \in X, \quad P(x=x) = \sum_y P(x=x, y=y) \quad (3.3)$$

对于连续型变量, 用积分代替求和

$$P(x) = \int p(x, y) dy \quad (3.4)$$

# 深度学习

## 3.5 条件概率

$$P(y=y | x=x) = \frac{P(y=y, x=x)}{P(x=x)} \quad (3.5)$$

给定  $x=x$  条件下， $y=y$  发生的概率

## 3.6 条件概率的链式法则

$$P(x^{(1)}, \dots, x^{(n)}) = P(x^{(1)}) \prod_{i=2}^n P(x^{(i)} | x^{(1)}, \dots, x^{(i-1)}) \quad (3.6)$$

链式法则 chain rule , 条件概率 product rule

$$P(a, b, c) = P(a|b, c) P(b, c)$$

$$P(b, c) = P(b|c) P(c)$$

$$P(a, b, c) = P(a|b, c) P(b|c) P(c)$$

## 3.7 独立性和条件独立性

两个随机变量  $x$  和  $y$ ，如果它们的 概率分布 可表示成两个因子乘积。  
称为 相互独立 independent

$$\forall x \in X, y \in Y, P(x=x, y=y) = P(x=x) \cdot P(y=y) \quad (3.7)$$

如果关于  $x$  和  $y$  条件概率分布对于  $z$  的每一个值都可写成乘积的形式。  
那么  $x, y$  在给定随机变量  $z$  时是 条件独立的 (conditionally independent)

$$\forall x \in X, y \in Y, z \in Z, P(x=x, y=y | z=z) = P(x=x | z=z) P(y=y | z=z) \quad (3.8)$$

公众号

【计算机视觉联盟】

# 深度学习

## 3.8 期望、方差和协方差

期望 expectation / expected value 当  $x$  由  $P$  产生,  $f$  作用于  $x$ ,  $f(x)$  平均值.

对离散型随机变量, 求和得列:

$$E_{x \sim p}[f(x)] = \sum_x P(x) f(x) \quad (3.9)$$

对连续型:

$$E_{x \sim p}[f(x)] = \int p(x) f(x) dx \quad (3.10)$$

几种简写方式:  $E_x[f(x)] \Rightarrow E[f(x)] \Rightarrow E[\cdot]$

期望是线性的.

$$E_x[\alpha f(x) + \beta g(x)] = \alpha E_x[f(x)] + \beta E_x[g(x)] \quad (3.11)$$

方差 variance 衡量的是当我们对  $x$  依据它的概率分布采样时, 随机变量  $x$  的函数值呈现多大的差异.

$$\text{Var}(f(x)) = E[(f(x) - E[f(x)])^2]$$

方差平方根为标准差 standard deviation.

协方差 covariance 某种意义上给出的变量线性相关性强度及尺度:

$$\text{Cov}(f(x), g(y)) = E[(f(x) - E[f(x)])(g(y) - E[g(y)])] \quad (3.12)$$

协方差绝对值很大, 变量值变化很大, 它们同时距均值很远

正的, 同时取得相对较大值

负的, 一个取得较大值的同时, 另一个倾向于取得相对较小值.

协方差矩阵 covariance matrix 是  $n \times n$  矩阵.

$$\text{Cov}(x)_{i,j} = \text{Cov}(x_i, x_j) \quad (3.14)$$

对角元素是方差

$$\text{Cov}(x_i, x_i) = \text{Var}(x_i) \quad (3.15)$$

公众号

【计算机视觉联盟】

## 3.9 常用概率分布

## 3.9.1 Bernoulli 分布

Bernoulli 分布 (Bernoulli distribution) 单个二值随机变量的分布。  
它由单个参数  $\phi \in [0, 1]$  控制,  $\phi$  给出了随机变量等于 1 的概率

$$P(x=1) = \phi \quad (3.16)$$

$$P(x=0) = 1 - \phi \quad (3.17)$$

$$P(x=x) = \phi^x (1-\phi)^{1-x} \quad (3.18)$$

$$E_x[x] = \phi \quad (3.19)$$

$$\text{Var}_x[x] = \phi(1-\phi) \quad (3.20)$$

## 3.9.2 Multinoulli 分布

Multinoulli 分布是多项式分布 (multinomial distribution) 特例。

多项式分布是  $[0, \dots, n]^k$  中的向量分布, 表示当对 Multinoulli 分布采样  $n$  次时  $k$  项中的每一个被访问的次数。

很多“强分布”其实说的就是 Multinoulli 分布, 但并没有说是  $n=1$  情况, 需注意。

Multinoulli 分布 (multinoulli distribution) 或范畴分布 (categorical distribution) 是指在且有  $k$  个不同状态的单个离散型随机变量上的分布, 其中  $k$  是一个有限值。

Multinoulli 分布由向量  $P \in [0, 1]^{k-1}$  参数化, 其中每个分量  $P_i$  表示第  $i$  个状态的概率。最后的第  $k$  个状态概率可以通过  $1 - P^T P$  给出。

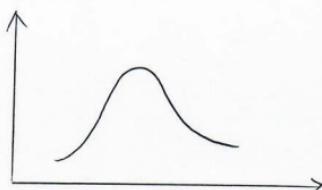
我们必须限制  $P^T P \leq 1$

# 深度学习

## 3.9.3 高斯分布

正态分布 normal distribution, 又称高斯分布 Gaussian distribution.

$$N(x; \mu, \sigma^2) = \sqrt{\frac{1}{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right) \quad (3.21)$$



公众号  
【计算机视觉联盟】

正态分布由两个参数控制,  $\mu \in \mathbb{R}$  和  $\sigma \in (0, \infty)$

中心位置  $E[X] = \mu$  标准差  $\sigma$ , 方差  $\sigma^2$

取  $\beta = \frac{1}{\sigma^2}$ ,  $\beta \in (0, \infty)$  控制分布的精度 (precision)

$$N(x; \mu, \beta^{-1}) = \sqrt{\frac{\beta}{2\pi}} \exp\left(-\frac{1}{2}\beta(x-\mu)^2\right) \quad (3.22)$$

- 正态分布是  
最以致密分布  
选择**
- (1) 很多真实情况接近正态分布。中心极限定理说明很多独立随机变量的和近似服从正态分布。
  - (2) 在具有相同方差的所有可能的概率分布中, 正态分布在本质上具有最大的不确定性。

多维正态分布, 参数是一个正定对称矩阵  $\Sigma$ :

$$N(\mathbf{x}; \boldsymbol{\mu}, \Sigma) = \sqrt{\frac{1}{(2\pi)^n \det(\Sigma)}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})\right) \quad (3.23)$$

对概率密度函数求值时需对  $\Sigma$  求逆, 使用一个稠密矩阵  $\beta$  替代:

$$N(\mathbf{x}; \boldsymbol{\mu}, \beta^{-1}) = \sqrt{\frac{\det(\beta)}{(2\pi)^n}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \beta(\mathbf{x}-\boldsymbol{\mu})\right) \quad (3.24)$$

常把协方差矩阵固定成对角阵, 且简单是否向同性高斯分布。

# 深度学习

## 3.9.4 指数分布和 Laplace 分布

在深度学习中，需要在  $x=0$  点处取得边界点 (sharp point) 分布。

公众号

可以使用 **指数分布** (exponential distribution)

【计算机视觉联盟】

$$p(x; \lambda) = \lambda e^{-\lambda x} \quad (3.25)$$

使用 **指示函数** (indicator function)  $I_{x \geq 0}$  使得当  $x$  取负值时概率为零。

**Laplace 分布**：允许在任意一点附近设置概率质量矩阵值。

$$\text{Laplace}(x; \mu, \gamma) = \frac{1}{2\gamma} \exp\left(-\frac{|x-\mu|}{\gamma}\right) \quad (3.26)$$

## 3.9.5 Dirac 分布 和 经验分布

某些情况，希望概率分布中的所有质量都集中在一个点上。

通过 Dirac delta 函数  $\delta(x)$  定义概率密度函数实现：

$$p(x) = \delta(x - \mu) \quad (3.27)$$

除了  $\mu$  外所有值为 0，但积分为 1。

Dirac delta 函数不像普通函数对  $x$  每一个值都有一个实值输出。

它是一种不同类型数学对象，广义函数 (generalized function)，依据积分性定义。

通过把  $p(x)$  定义成向函数左移  $-n$  个单位，得到一个  $x=\mu$  处且有无限窄也无限高的峰值概率质量。

Dirac 分布常作为经验分布的一个组成部分出现：

$$\hat{p}(x) = \frac{1}{m} \sum_{i=1}^m \delta(x - x^{(i)})$$

集合

经验分布将概率密度函数给  $m$  个点  $x^{(1)}, \dots, x^{(m)}$  中的每一个点是给定数据集或样本集模型，Dirac delta 函数才有必要。

高维，经验分布可定义成一个 Multinomial 分布

## 3.9.6 分布的混合

通过组合一些简单的概率分布定义新的概率分布。

混合分布 mixture distribution 由一些组件 component 部分构成

每次实验，样本由哪部分组件分布产生的取决于从一个 Multinomial 分布中采样的结果：

$$P(x) = \sum_i P(c=i) P(x|c=i) \quad (3.29)$$

这里是  $P(c)$  对于组件的一个 Multinomial 分布

潜变量 (latent variable) 是我们不能直接观测到的随机变量。

高斯混合模型 Gaussian Mixture Model :

它的组件  $P(x|c=i)$  是高斯分布，每个组件都有各自参数。

均值  $\mu^{(i)}$  和协方差矩阵  $\Sigma^{(i)}$ 。

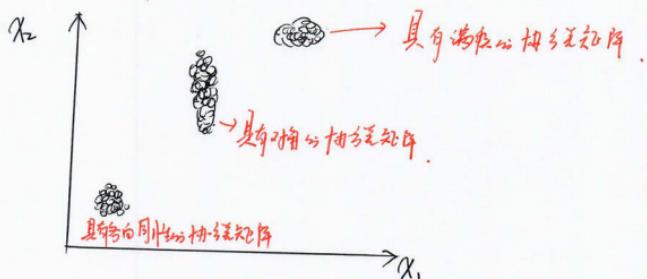
高斯混合模型的参数 指明了每个组件  $i$  的先验概率  $\alpha_i = P(c=i)$

先验：表示了在观测到  $x$  之前往递归模型关于  $c$  的信念。

$P(c|x)$  是后验概率 posterior probability:

在观测到  $x$  之后进行计算。

高斯混合模型样本

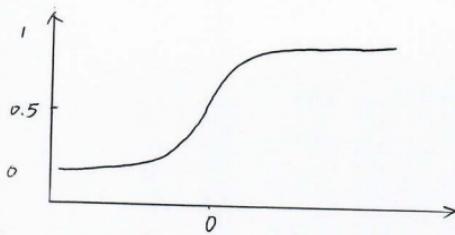


# 深度学习

## 3.10 常用函数的有用性质

logistic sigmoid 函数

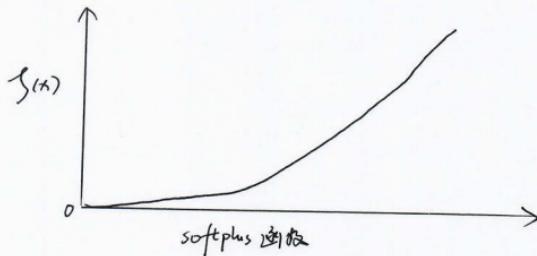
$$s(x) = \frac{1}{1 + \exp(-x)} \quad (3.30)$$



softplus 函数

$$\zeta(x) = \log(1 + \exp(x)) \quad (3.31)$$

softplus 函数可用来产生正态分布  $\mu$  和  $\sigma$  参数，范围  $(0, \infty)$



还有一个函数：

$$x^+ = \max(0, x) \quad (3.32)$$

负部函数。  
 $x^- = \max\{0, -x\}$

一些公式：

$$s(x) = \frac{e^x}{e^x + 1} \quad (3.33)$$

$$\forall x \in (0, 1), s^{-1}(x) = \log\left(\frac{x}{1-x}\right) \quad (3.38)$$

$$\frac{d s(x)}{dx} = s(x)(1-s(x)) \quad (3.34)$$

$$\forall x > 0, \zeta^{(+)}(x) = \log(e^x - 1) \quad (3.39)$$

$$1 - s(x) = s(-x) \quad (3.35)$$

$$s(x) = \int_{-\infty}^x s(y) dy \quad (3.40)$$

$$\log s(x) = -\zeta(-x) \quad (3.36)$$

$$\zeta(x) - \zeta(-x) = x \quad (3.41)$$

$$\frac{d \zeta(x)}{dx} = s(x) \quad (3.37)$$

# 深度学习

## 3.11 贝叶斯规则

如何通过  $P(y|x)$  求  $P(x|y)$

贝叶斯规则 (Bayes' Rule):

$$P(x|y) = \frac{P(x) P(y|x)}{P(y)} \quad (3.42)$$

$\downarrow$

$$P(y) = \sum_x P(y|x) P(x)$$

## 3.12 连续型变量的技术细节

连续型随机变量和概率密度函数 [深入理解概率论 (measure theory)]

密度论使得计算时不会出现悖论。

零测试 measure zero : 几乎处处 almost everywhere

注意细节:

假设  $x$  和  $y$  满足  $y = g(x)$ ,  $g$  是可逆可微函数

那  $P_y(y) = P_x(g^{-1}(y))$  不一定对!

例)  $y = \frac{x}{2}$ ,  $x \sim U(0,1)$ , 若  $P_y(y) = P_x(g(y))$  则  $P_y$  在  $[0, \frac{1}{2}]$  上都为 0, 在此区间外为 1.

则  $\int P_y(y) dy = \frac{1}{2} \quad (3.43)$

不满足概率密度定义, 权分为 1

联想只落在无穷小的区域内的概率为  $P(g(S))$ .

原因在于引入函数  $g$  后造成的空间变形.

同时衡量, 需保持  $|P_g(g(x)) dy| = |P_x(x) dx| \quad (3.44)$

$P_g(y) = P_x(g^{-1}(y)) \left| \frac{\partial x}{\partial y} \right| \quad (3.45)$

$P_x(x) = P_g(g(x)) \left| \frac{\partial g(x)}{\partial x} \right| \quad (3.46)$

拓展为 Jacobian 矩阵:  $J_{i,j} = \frac{\partial x_i}{\partial y_j}$

$$P_x(x) = P_g(g(x)) \left| \det \left( \frac{\partial g(x)}{\partial x} \right) \right| \quad (3.47)$$

# 深度学习

公众号

【计算机视觉联盟】

## 3.13 信息论

信息论的基本想法是一个不太可能发生的事情居然发生了，要比一个非常可能的事件发生能提供更多的信息。(如太阳出现日食)

**量化信息**  $\left\{ \begin{array}{l} \text{① 非常可能发生的事情信息量较少，并且极端情况下，确保} \\ \text{能够发生的事情应该没有信息量。} \\ \text{② 较不可能发生的事件具有更高的信息量} \\ \text{③ 独立事件应具有增量的信息。} \end{array} \right.$

为满足上述性质，定义一个事件  $x=x$  的自信息 self-information:

$$I(x) = -\log P(x) \quad (3.48)$$

↓  
定义单位是奈特 nats. 一奈特是它由概率观测一个事件的信息量。

其它教材可使用底数为 2，单位是 bit 或香农 shannons.

比特度量的信息是奈特度量的  $\log_2 e$  倍。

自信息只处理单个输出，可用香农熵 Shannon entropy 对整个概率分布的不确定性总量进行优化：

$$H(x) = E_{x \sim p}[I(x)] = -E_{x \sim p}[\log P(x)] \quad (3.49)$$

↓

香农熵指遵循这个分布的事件所产生期望信息总量

当  $x$  是连续的，香农熵被称为微分熵 (differential entropy)

## 3.13 信息论

同一个随机变量  $x$  有两个单独的概率分布  $P(x)$  和  $Q(x)$   
可使用  $KL$  故度 ( $Kullback-Leibler$  divergence) 衡量这两个分布的差异：

$$D_{KL}(P||Q) = E_{x \sim P} \left[ \log \frac{P(x)}{Q(x)} \right] = E_{x \sim P} [\log P(x) - \log Q(x)] \quad (3.50)$$

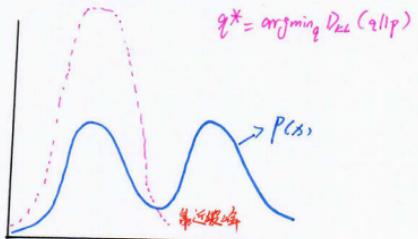
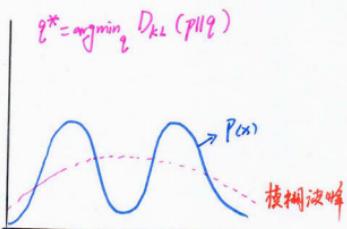
在离散型变量情况下， $KL$  故度衡量的是，当使用一种使得  
概率分布  $Q$  产生的消息的长度最小化时，发送包含由概率分布  $P$   
产生的符号消息时，所需要的额外信息量。

$KL$  故度非负

$KL$  故度为 0 当且仅当  $P$  和  $Q$  在离散型下分布相同，连续型处处相同。

$KL$  故度常被用作分布之间的距离，但并不对称。

$$D_{KL}(P||Q) \neq D_{KL}(Q||P)$$



$KL$  故度不对称，选择使用  $KL$  故度哪个方向取决于问题。

$$\text{交叉熵 } H(P, Q) = H(P) + D_{KL}(P||Q)$$

与  $KL$  故度很像：

$$H(P, Q) = -E_{x \sim P} \log Q(x) \quad (3.51)$$

$Q$  最小化交叉熵等价于最小化  $KL$  故度，因为  $Q$  并不参与被商除一项。

$$\lim_{x \rightarrow 0} x \log x = 0$$

# 深度学习

## 3.14 结构化概率模型

我们可以把概率分布分解成许多因子乘积，而不是使用单一函数表示：

$$p(a, b, c) = p(a) p(b|a) p(c|b) \quad (3.52)$$

↓  
这种分解可极大减少用 $\lambda$ 来描述一个分布的参数数量

当我们用图来表示这种概率分布的分解，称为结构化概率模型  
(structured probabilistic model)

### 图模型 (graphical model)

两种：有向图、无向图。两种图模型都使用图 $G$

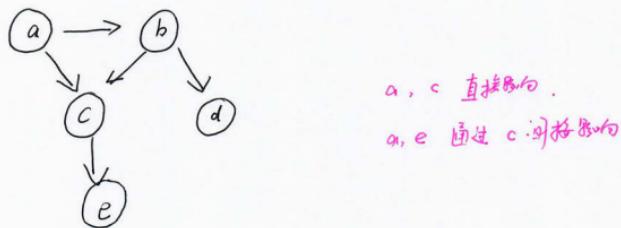
公众号

有向 (directed) 使用带有序向边的图。

【计算机视觉联盟】

$$p(x) = \prod_i p(x_i | \underbrace{\text{父节点}}_{\downarrow} \text{为 } \text{父节点}, \text{为 } \text{P}_{\text{par}}(x_i)) \quad (3.53)$$

组成 $x_i$ 条件概率的影响因子称为 $x_i$ 父节点，为  $P_{\text{par}}(x_i)$



$$p(a, b, c, d, e) = p(a) \underbrace{p(b|a)}_{\text{在 } a \text{ 影响 } b} \underbrace{p(c|a, b)}_{\text{在 } a, b \text{ 影响 } c} p(d|b) p(e|c) \quad (3.54)$$

无向使用无向边图。

$G$  中任何满足两两之间有边连接的顶点集合被称为团。

每个团 $C^{(i)}$ 都伴随着一个因子  $\phi^{(i)}(C^{(i)})$ ，仅是函数，不是概率分布。

# 深度学习

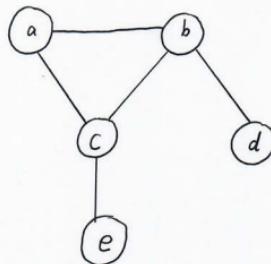
## 3.14 结构化概率模型

随机变量的联合概率与所有这些因子的乘积成比例，因子值越大，则可能性越大。

不能保证乘积求和为1，需要除以一个归一化常数 $\Gamma$ 来得归一化的概率分布，归一化常数 $\Gamma$ 被定义为中断且所有状态求和或积分。

概率分布为：

$$p(x) = \frac{1}{\Gamma} \prod_i \phi^{(i)}(C^{(i)}) \quad (3.55)$$



$$p(a, b, c, d, e) = \frac{1}{\Gamma} \phi^{(1)}(a, b, c) \phi^{(2)}(b, d) \phi^{(3)}(c, e) \quad (3.56)$$

公众号  
【计算机视觉联盟】