**深度学习**

V. 1.0.1

备注：本文档大量常考《Deep Learning》（[美] Ian Goodfellow ,[加] Yoshua Bengio,[加] Aaron Courvile 著），

《深度学习入门-基于Python的理论与实现》（[日] 斋藤康毅 著），请勿用做商业。

注：如涉及侵权，务必联系 [leep233@foxmail.com](mailto:leep233@foxmail.com) . 本人将立即整改；

目的：本文档只用作记录，分析以及学习，并无任何商业目的

目录

[线性代数 2](#_Toc5606)

[概念 2](#_Toc18742)

[导数与偏导数 2](#_Toc21674)

[积分 7](#_Toc25685)

[向量与矩阵 8](#_Toc30894)

[范数 8](#_Toc9544)

[运算 9](#_Toc1916)

[特殊向量与矩阵 10](#_Toc31240)

[概率 11](#_Toc9051)

[随机变量 11](#_Toc32558)

[随机变量与概率分布 11](#_Toc5186)

[边缘概率 12](#_Toc2955)

[条件概率 12](#_Toc12690)

[独立性与条件独立 13](#_Toc19849)

[期望 13](#_Toc31339)

[机器学习 13](#_Toc12450)

# 线性代数

## 概念

**实数（real number）**

**有理数和无理数的总称。用表示实数集。**

**标量（scalar）**

**表示一个具体实数。使用小写斜体来表示（如：标量 ）**

**向量（vector）**

**表示一列数。使用小写加粗来表示（如：向量 ）**

注意: 假设有一个向量 x , 这个向量里面有 *n* 个元素，并且每个元素属于实数，那么我们可以记作 ; **其中向量里面的元素使用标量+下角标的形式表示**

**矩阵（matrix）**

**矩阵是一个二维数组。**

**使用加粗大写字母表示（如：假设矩阵 ，是一个m 行 n 列 的矩阵 我们可以记作 ；如果我们需要表示A 中 第 i 行的 j 列 的元素 我们可以记作 ；）**

注意: 假设矩阵 A ，是一个m 行 n 列 的矩阵 我们可以记作 ；

如果我们需要表示A 中 第 i 行的 j 列 的元素 我们可以记作 ；

表示第i 行的所有列;

表示第 j 列的所有行;

## 导数与偏导数

导数（derivative）

**如下图所示，小明开车去旅游，图中是某段事件内，车速s 和 时间 t 对应的函数关系图。求t1到t2 时间里，小明开车的平均速度？**

**S**

**t**



t1

t2

s1

s2

**我们很简单可以得出小明t1~t2之间的平均速度为 ;**

**如果我们将上图中，s 和 t ，使用函数关系坐标系来表示 s = f(x) , t = x; 那么函数关系图如下：**

**x**



**通过上图我们用来代替轴，那么，之间的**平均变化率, 记作 ;

**通过上面求平均变化率的例子，当 无线趋近 时，即 ) , 那么求出来的就是**导数**，记作**， 或 ;

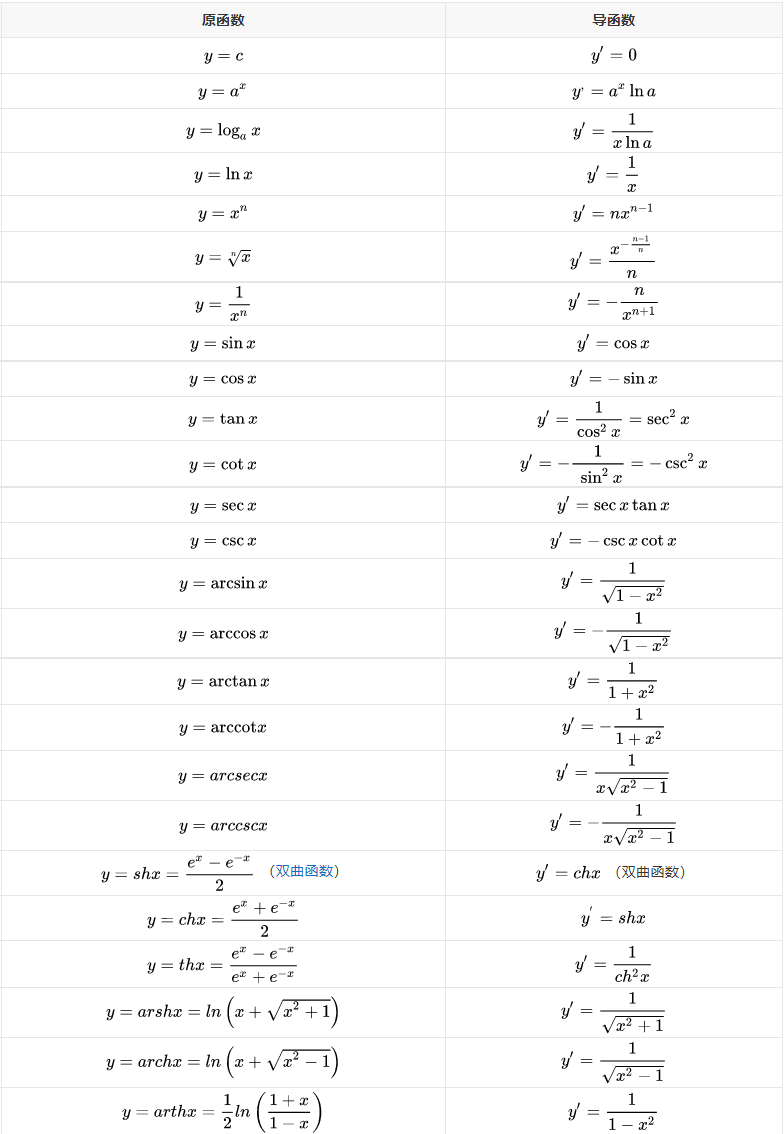
**导数的几何意义:在几何数学中，导数即时该点的斜率，同时我们可以通过点斜式来求得关于过目标点的切线方程。**

**点斜式：已知目标点（ , ）,斜率 *k*, 通过点斜式可以求出目标点的切线方程，即 ;**

**斜率与函数单调性：当斜率k>0时，表示函数单调递增，k<0时，函数单调递减，当k ≈ 0，时表示到达了函数的极值或鞍点**

* 加法:
* 乘法：
* 除法:
* 链式法则:

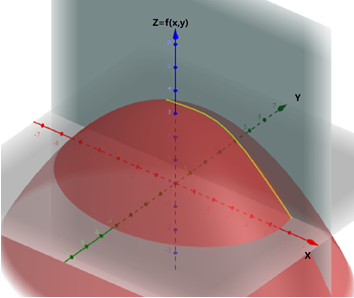
以下是常用的函数的求导公式



偏导数（partial derivative**）**

**当函数的具有多个自变量（输入参数，）求其中一个的导数时，我们称为偏导数**

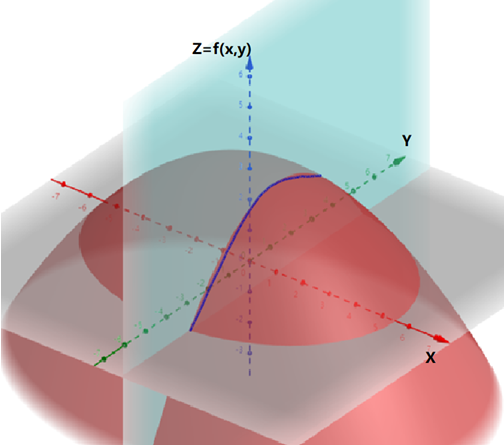
设函数 ,我们求关于函数 的偏导数时，需要将的值固定，的值来求偏导，记作 ,通过坐标系表示如下图



,

由图可以了解到求 关于 的偏导,其实就是求图中 在黄色线中的导数；

同样的如果我们需要求 ，我们只需要固定的值，就可以求出点的导数。如下图

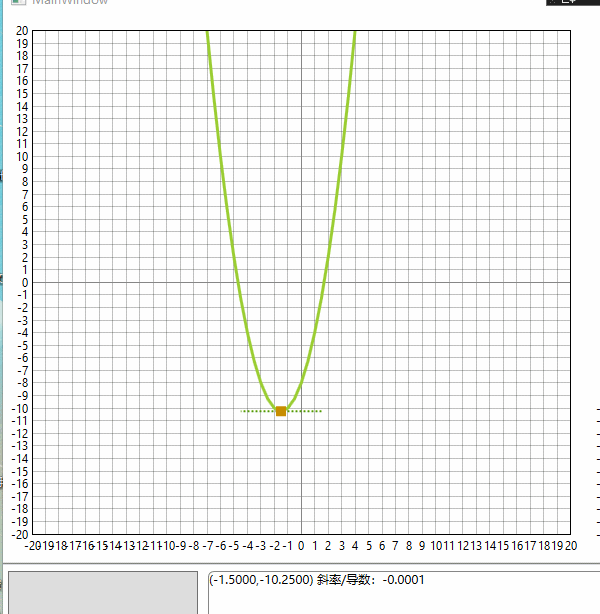


,

梯度下降（Gradient descent）

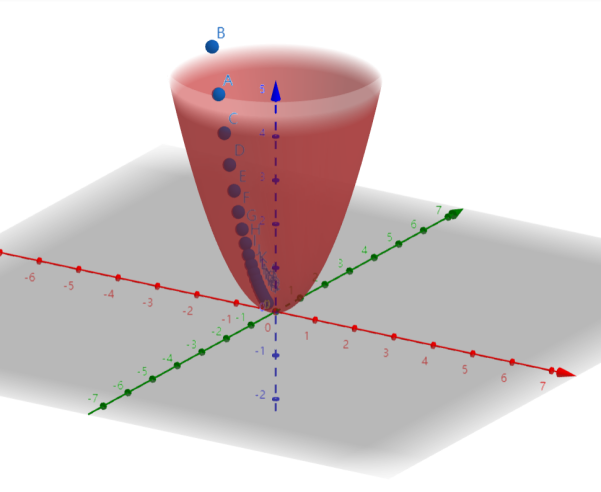
已知函数 ，求该方程的极值。

由上述方程开口向上，有极小值的抛物线，并且导数求方程;

首先我们假设每次我们变动的量，这样的话我们可以求出 处的斜率，通过斜率可以得知当 时，即找到了极值，同时由于斜率可以表示函数单调性的原因,也可以表示 往哪个方向变化才能找到极值，所以我们可以让 , 其中常量 我们也叫做学习率，那么只要我们一直循环n次 之后，当 时，当前点就可以近似认为时函数的极值。我们将这种方法叫做**梯度下降（gradient descent）**；下图为梯度下降的计算的变化图；

以上为2D坐标空间内的梯度下降概念；

假设我们在多维度的空间内，假设函数,同样的我们求出 ，同时我们定义学习率，那么我们可以得到向量,,同样我们循环N次这种操作将会得到一个最低点(极值)；当我们面对的时多维度计算的时候我们判断是否到达极值时,即();下图描述了函数求梯度变化的规律



## 积分

下图时关于函数f(x) 和 x 的函数关系图，求a,b区间内（阴影区域）的面积；

a

b



如图可知，由于我们的线性关系时曲线所以我们没有办法通过寻常求面积的方式求得阴影的面积，那么我们可以使往一个方向移动极小的距离，同时我们可以也可以得出移动 距离后的对应的函数的值，记作，那么我们可以近似这小块的面积等于 ,只要一直将这个区间内所有的小块面积相加就可以得到整个大块阴影的面积，我们记作;

同时如果我们的函数是关于多个自变量求积分则；如,需要求在 区域内的积分记作

## 向量与矩阵

### 范数

在机器学习中，我们使用范数来衡量向量的大小；

**范数:**

**, 其中 ,其实范数是衡量向量x和原点（所有元素都是零的点）之间的距离。我们使用函数来代表范数即，则范数需要满足以下条件**

* **正定性：**
* **三角不等式：**
* **正值齐次性：**

**范数**

**机器学习中，我们需要区分零与非零但是值很小的元素，这种情况下，我们使用各个位置的斜率相同，同时保持简单的数学形式的函数：范数**

**机器学习中，零和非零元素之间的差异非常重要时，通常使用范数。**

**范数（欧几里得范数）**

**当时，称为范数，也叫做欧几里得范数，他表示从原点出发到向量x点的欧几里得距离。由于此范数在机器学习中使用的非常频繁所有我们简写成|| x || , 同时经常用来衡量向量的大小，**

**范数（最大范数）**

**在机器学习中，也经常出现最大范数，它表示向量中具有最大幅值得元素得绝对值；**

**Frobenius范数**

**在机器学习中，当我们需要对矩阵大小进行衡量得时候，我们通常使用Frobenius范数**

### 运算

**转置**

**矩阵的转置即以主对角线（主对角线：从左上角到右下角的对角线）为轴的镜像，即矩阵的行变列，列变行。**

**矩阵的转置为，定义如下：;**

**标量可以看作一个1x1的矩阵，所以标量的转置等于自身 ；**

**向量可以看作是一个列的矩阵，它转置之后变成一个一行的矩阵 ();**

**加减法**

**矩阵与矩阵相加相当于每个对应的元素相加。如;**

**标量与矩阵相加或相乘时，相当于矩阵的每个元素乘以或加上这个标量。**

**如 ;**

**深度都学习中，我们允许矩阵与向量相加，产生一个新的矩阵。如,其中 ; 换言之，将向量b 与 矩阵 A 的每一行相加，这种隐式将向量b 复制到很多位置的方式叫做广播**

**乘积**

**两个想同纬度的向量和的点积（dot product）可以看作。**

**两个矩阵和相乘等到一个新的矩阵。矩阵相乘必须满足左边矩阵的列数和右边矩阵的行数相等。如果矩阵是,矩阵是 ,那么他们相乘等到的矩阵 为 ，即 ，乘法操作定义为：，我们可以看作 的行 和 的列 进行点积。**

* **分配律：**
* **结合律：**
* **交换律（矩阵相乘是不满足交换律的，但是两个向量点积满足交换律）：**

**Hadamard乘积**

**两个矩阵A和B 进行Hadamard乘积时，等于两个矩阵对应行列的元素相乘，等到新矩阵C，操作定义为**

**轨运算**

**迹运算返回矩阵对角线的和，记作 ;**

**迹运算提供了另一种Frobenius的范数方式 : ;**

**迹运算的转置运算下是不变的，即**

**多矩阵的积运算等于将最后一个矩阵移到最前面一个矩阵的位置:或记作**

**标量的迹运算等于标量自己:**

### 特殊向量与矩阵

**单位向量**

**方阵**

**矩阵的行数和列数相等时，我们称为方阵**

**单位矩阵**

**所有沿主对角线得元素都是1，而且其他位置得元素都是0的方阵叫做单位矩阵。任意向量和单位矩阵相乘都不会进行任何改变。**

**通常我们使用 来表示一个n行n列的单位矩阵。下图为 矩阵：**

**逆矩阵**

**矩阵，是否存在逆矩阵必须满足公式：**

**对角矩阵**

# 概率

概率即用于描述不确定性的声明。

机器学习中通常需要处理不确定量，也可能需要处理随机量

在机器学习中不确定性由3中可能的来源

* 被建模系统内在的随机性
* 不完全观测性
* 不完全建模

### 随机变量

**离散型**

**离散型随机变量拥有有限或者可数无限多的状态。**

**连续型**

**连续型变量拥有不可数实数值**

### **随机变量与概率分布**

**离散型变量和概率质量函数（probability mass function , PMF）**

**对于离散型随机变量的概率分布可以使用概率质量函数来描述，我们通常使用大写字母来表示， 的概率我们使用来表示，当概率=1时，表示事件时必然发生的，概率=0时，事件一定不会发生表示遵循分布。**

**概率质量函数同时可以作用于多个随机变量，我们称为联合概率分布（joint probability distribution）。或,表示 同时发生的概率，也可以写成。**

**如果函数P是随机变量 的PMF，必须满足以下条件**

* **P的定义域必须是的所有可能状态集合**

**连续型变量和概率密度函数（probability density function , PDF）**

**对于连续型随机变量时，我们使用概率密度函数 ，我们时使用p 来表述概率密度函数，必须满足以下条件**

* **的定义域必须是的所有可能状态集合**
* **，其中表示一个极小值**

### **边缘概率**

一组变量的联合概率分布，需要了解其中一个子集的概率分布，这种定义在子集上的概率分布被称为**边缘概率分布（marginal probability distribution），**

已知随机变量和，并且我们知道

* 离散型随机变量边缘概率分布:
* 连续型随机变量边缘概率分布：

### 条件概率

当某个事件在给定其他事件发生时出现的概率我们叫做**条件概率。**

如果事件发生时发生的概率我们记作。此概率的计算公式为：

条件概率的链式法则：

### 独立性与条件独立

两个随机变量x和y，对于的概率分布是完全不相关的我们叫做相互独立事件，如果要计算两个事件同时发生的概率 我们需要 ;

关于x和y的条件概率分布对于z的每一个值都可以写成乘积的形式，我们就称作随机变量x,y 在给定随机变量z是**条件独立**的：

;

### 期望

下表是的分布列

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

当我们 使用 E = 时，我们得出的值E就叫做 关于分布的数学**期望**。

如果我们将x 看作f(x)函数，那么期望值表示为

* 离散型随机变量：
* 连续性随机变量：

假设,那么;

### 方差

下面分别为大明，小明明进行射击的分布列

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

通过两个分布列可以得知大明的数学期望 为 E() = , 小明为 E() = ，当我们需要对大明和小明的射击水平进行比较我们不能简单的比较期望值的大小，这样我们引入方差的概念，通过比较大明和小明的方差来对比两人的稳定程度，方差越小说明射击水平越稳定，数学公式表示为

* 离散型随机变量：
* 连续性随机变量：

如果随机变量 ,则

### 协方差

# 机器学习