# 1 矩阵

## 1.1 对称矩阵

性质：

a 实对阵矩阵的特征值()为实数。

b 对阵矩阵的特征向量相互正交。

二次型：

## 1.2 正定矩阵

对任意，二次型.

性质：

a 正定矩阵的行列式恒为正；

b 实对称矩阵A正定当且仅当A与单位矩阵合同；

c 若A是正定矩阵，则A的逆矩阵也是正定矩阵；

d 两个正定矩阵的和是正定矩阵；

e 正实数与正定矩阵的乘积是正定矩阵。

## 1.3 正交矩阵 ，

性质：

a 任何正交矩阵的行列式是+1或−1

b 所有的列向量都是单位正交向量

c 所有的行向量都是单位正交向量

d 正交矩阵对向量进行正交变换，且正交变换不改变向量的长度(范数)

## 1.4 矩阵变换

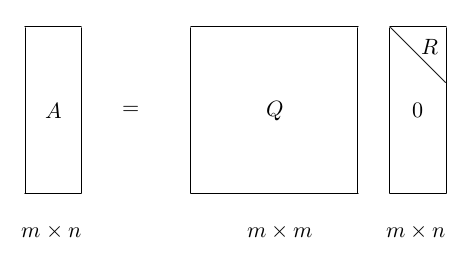
合同变换：，为可逆矩阵

相似变换：，为可逆矩阵

## 1.5 矩阵分解

LU分解：，M非奇异，L下三角，U上三角。

[QR分解](https://zh.wikipedia.org/wiki/QR%E5%88%86%E8%A7%A3)：,正交，上三角。



[奇异值分解](https://zh.wikipedia.org/wiki/%E5%A5%87%E5%BC%82%E5%80%BC%E5%88%86%E8%A7%A3)SVD：,和都是正交矩阵，非负对角。

[谱分解](https://zh.wikipedia.org/wiki/%E8%B0%B1%E5%88%86%E8%A7%A3)：,对称，正交，对角。

[极分解](https://zh.wikipedia.org/wiki/%E6%9E%81%E5%88%86%E8%A7%A3)：,正交，对称半正定。

# 2 导数

导数是微积分学中重要的基础概念。一个函数在某一点的导数描述了这个函数在这一点附近的变化率。导数的本质是通过极限的概念对函数进行局部的线性逼近。当函数的自变量在一点上产生一个增量时，函数输出值的增量与自变量增量的比值在趋于0时的极限如果存在，即为在处的导数，记作或或.

## 2.1 求导法则

1 线性性

2 乘积的导

3 商的导数

4 链式法则

例子：

## 2.2 偏导数

在数学中，一个多变量的函数的偏导数是它关于其中一个变量的导数，而保持其他变量恒定（相对于全导数，在其中所有变量都允许变化）。

例子：

## 2.3 雅可比矩阵

在向量分析中，雅可比矩阵是函数的一阶偏导数以一定方式排列成的矩阵，其行列式称为雅可比行列式。

例子：

## 2.4 黑塞矩阵

黑塞矩阵，又译作海森矩阵或海瑟矩阵等，是一个由多变量实值函数的所有二阶偏导数组成的方块矩阵，由德国数学家奥托·黑塞引入并以其命名。

# 3 算法

## 3.1 二分法

**介值定理**

连续函数，若满足或，则存在，使得.

**零点存在定理**

连续函数在闭区间异号，则必存在，使得.

例子：求一个数的平方根

s1 a=0,b=C,计算，其中

s2 若，a=，否则

s3 若，终止；否则转s1

## 3.2 牛顿切线法

设有函数，则其零点可通过下面式子迭代求解

例子：求一个数的平方根

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
|  | 1 | 1.500000 | 2.000000 | 2.500000 | 3.000000 |
|  | 1 | 1.416667 | 1.750000 | 2.050000 | 2.333333 |
|  | 1 | 1.414216 | 1.732143 | 2.000610 | 2.238095 |
|  | 1 | 1.414214 | 1.732051 | 2.000000 | 2.236069 |
|  | 1 | 1.414214 | 1.732051 | 2.000000 | 2.236068 |

## 3.3 梯度下降法

设有函数，则其极小值可通过下面式子迭代求解

被称为学习率或步长。

例子：

令，

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0.1 | 0.2 | 0.4 | 0.6 | 0.8 |
|  | 8 | 6 | 2 | -2 | -6 |
|  | 6.4 | 3.6 | 0.4 | 0.4 | 3.6 |
|  | 5.12 | 2.16 | 0.08 | -0.08 | -2.16 |
|  | 4.096 | 1.296 | 0.016 | 0.016 | 1.296 |
|  | 3.2768 | 0.7776 | 0.0032 | -0.0032 | -0.7776 |

实际可得时才收敛，最终当x=0时函数取极小值。

我们可以通过α来控制每一步走的距离，以保证不要步子跨的太大，就是不要走太快，错过了最低点。同时也要保证不要走的太慢，导致太阳下山了，还没有走到山下。所以α的选择在梯度下降法中往往是很重要的，α太小的话，可能导致迟迟走不到最低点；太大的话，会导致错过最低点。

## 3.4 最小二乘法

简单地说，最小二乘的思想就是要使得观测点和估计点的距离的平方和达到最小.

如果满秩，则.

如果不可逆，目标函数,,.

采用梯度下降法

*例子：拟合z=2x+3y*

|  |  |  |
| --- | --- | --- |
| *x* | *y* | *z* |
| *1* | *1* | ***4.9*** |
| *1* | *2* | ***8.1*** |
| *2* | *0* | ***3.9*** |
| *0* | *2* | ***6*** |
| *1* | *0* | ***1.9*** |
| *3* | *1* | ***8.9*** |

采用直接方法得到：

*采用梯度下降法，学习1000步解果为：*

## 3.5 反向传播算法

**1 神经网络**

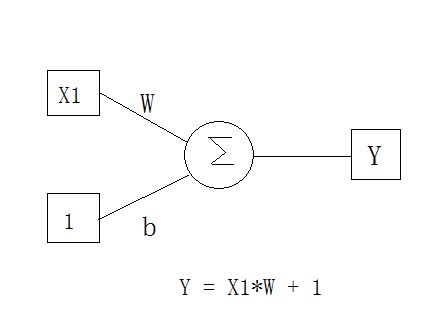


图 简单神经网络

神经网络Y =W\*X +B根据input X的值，得到output Y值。我们需要一个函数可以对得到的Y值进行分类，比如Y大于0，分类为1，小于0，分类为0。激活函数就是承担该角色，即实际上

是激活函数。f(x)必须是非线性函数，否则神经网络就只能学习线性模型。常用的激活函数为Sigmoid函数，简称S函数。

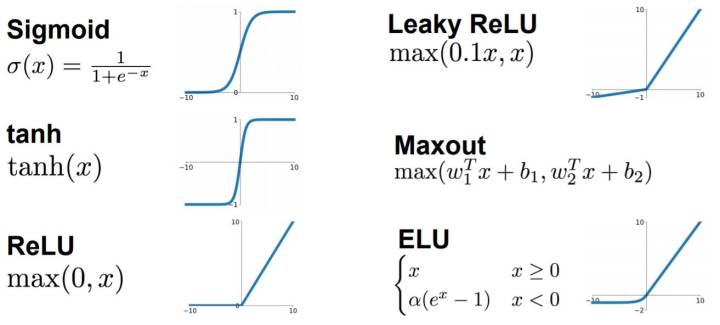


图 常用的激活函数

**一般神经网络**

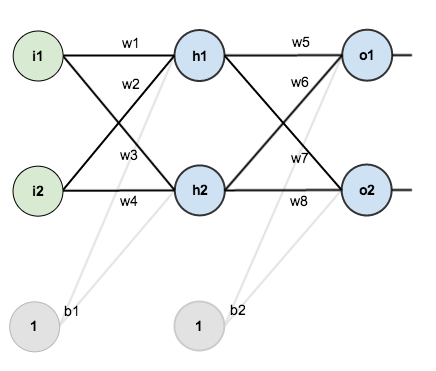


图 一般神经网络

矩阵序列就决定了整个神经网络，对神经网络的训练就是对矩阵序列进行参数求解。

以上面的神经网络为例子。

我们对矩阵赋上初始值:,

提供一个简单的训练集:,，激活函数:

下面将通过反向传播算法对神经网络进行训练。

**2 前向传播**

从output侧向input侧逐步求解矩阵的参数。首先通过前向传播，求取神经网络对于给定的训练集产生的输出值。

，

**3 反向传播**

反向传播的本质只是对链式法则的巧妙运用。我们通过前向传播获得了input经过神经网络后的output值，实际上我们需要的是target值，这便存在误差

一般取二范数，即.e是与矩阵参数w有关的函数，为了求解，研究f(w)关于w的梯度。

对于输出层，根据链式法则，.为上一层网络的输出值。

然后，采用梯度下降法更新输出层的参数。

参考资料：

[1] 反向传播演示 <https://google-developers.appspot.com/machine-learning/crash-course/backprop-scroll/?hl=zh-CN>

[2] 机器学习：一步步教你理解反向传播方法 <https://yongyuan.name/blog/back-propagtion.html>