# 矩阵求导(一)







# 矩阵求导(一)



thinkando (关注)



💖 1 2018.11.25 21:08:09 字数 278 阅读 33,451

• 矩阵求导(Matrix Derivative)也称作矩阵微分(Matrix Differential),在机器学习、图像处 理、最优化等领域的公式推导中经常用到。

## 1. 布局约定 (Layout conventions)

• 布局(Layout): 在矩阵求导中有两种布局,分别为分母布局(denominator layout)和分子布局 (numerator layout)。这两种不同布局的求导规则是不一样的。

布局(Layout): 在矩阵求导中有两种布局,分别为**分母布局(denominator layout)**和分子布局(numerator layout)。这两种不同布局的

向量 
$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$
,关于标量 $x$  的求导。

在分子布局下,为:

$$\frac{\partial \mathbf{y}}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x} \\ \frac{\partial y_2}{\partial x} \\ \vdots \\ \frac{\partial y_n}{\partial x} \end{bmatrix}$$
(1)

而在分母布局下,为

$$\frac{\partial \mathbf{y}}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x} & \frac{\partial y_2}{\partial x} & \cdots & \frac{\partial y_n}{\partial x} \end{bmatrix}$$

通过观察和推导我们可以知道,分子布局和分母布局之间刚好差一个转置,即在分子布局下与原来Y相同,而在分母布局下差一个转置。

对于正切矩阵 $\frac{\partial t}{\partial t}$ 采用分母布局,即 $\mathbf{Y}^{\mathsf{T}}$ ,很不符合表达的习惯,所以本文中我们采用的是**分子布局**。

image.png

## 2. 关于标量的导数

#### 2.1 标量与标量X的求导

这中情况就是我们平时的代数求导,直接就是 $\frac{\partial y}{\partial x}$ 

image.png

#### 2.2 向量与标量X的求导

#### 推荐阅读

推荐系统召回算法之——图模型

(Personal Rank)

阅读 443

机器学习高频面试题(41道)

阅读 3,720

YOLOv2网络

阅读 375

python用线性回归预测时间序列股票

价格

阅读 1,223

基于树模型的集成算法---Random

Forest

阅读 1,001





# 矩阵求导 (一)



thinkando(关注)



向量  $\mathbf{y} = \begin{bmatrix} y_2 \\ \vdots \end{bmatrix}$ , 关于标量x 的求导就是  $\mathbf{y}$  的每一个元素分别对x求导,可以表示为

image.png

## 2.3 矩阵与标量X的求导



$$\frac{\partial \mathbf{Y}}{\partial x} = \begin{bmatrix} \frac{\partial y_{11}}{\partial x} & \frac{\partial y_{12}}{\partial x} & \cdots & \frac{\partial y_{1n}}{\partial x} \\ \frac{\partial y_{21}}{\partial x} & \frac{\partial y_{22}}{\partial x} & \cdots & \frac{\partial y_{nn}}{\partial x} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_{n1}}{\partial x} & \frac{\partial y_{n2}}{\partial x} & \cdots & \frac{\partial y_{nn}}{\partial x} \end{bmatrix}$$

image.png

### 推荐阅读

推荐系统召回算法之——图模型 (Personal Rank)

阅读 443

机器学习高频面试题(41道)

阅读 3,720

YOLOv2网络

阅读 375

python用线性回归预测时间序列股票 价格

阅读 1,223

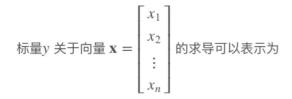
基于树模型的集成算法---Random

Forest

阅读 1,001

## 3. 关于向量的导数

#### 3.1 标量关于向量 x 的导数



$$\frac{\partial y}{\partial \mathbf{x}} = \left[ \begin{array}{ccc} \frac{\partial y}{\partial x_1} & \frac{\partial y}{\partial x_2} & \cdots & \frac{\partial y}{\partial x_n} \end{array} \right]$$



image.png

#### 3.2 向量关于向量 x 的导数

向量函数 (即函数组成的向量) 
$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$
关于向量 $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ 的导数记作 
$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial x_1} & \frac{\partial y_n}{\partial x_2} & \cdots & \frac{\partial y_n}{\partial x_n} \end{bmatrix}$$

此时获得的矩阵 $\frac{\partial y}{\partial x}$  叫做Jacobian 矩阵。

image.png



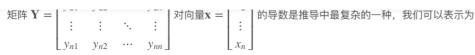


# 矩阵求导 (-









 $\frac{\partial \mathbf{Y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_{11}}{\partial x_1} & \frac{\partial y_{1n}}{\partial x_2} & \cdots & \frac{\partial y_{1n}}{\partial x_n} \\ \frac{\partial y_{21}}{\partial x_1} & \frac{\partial y_{22}}{\partial x_2} & \cdots & \frac{\partial y_{2n}}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_{n1}}{\partial x_1} & \frac{\partial y_{n2}}{\partial x_2} & \cdots & \frac{\partial y_{nn}}{\partial x_n} \end{bmatrix}$ 

image.png

### 4. 关于矩阵的导数

我们一般只考虑标量关于矩阵的导数(因为矩阵对向量和矩阵的导数与前面2.3节的内容一致或相似),即标量y 对矩阵 X 的导数为  $\frac{\partial y}{\partial X}$ 

$$\frac{\partial y}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial y}{\partial x_{11}} & \frac{\partial y}{\partial x_{21}} & \cdots & \frac{\partial y}{\partial x_{n1}} \\ \frac{\partial y}{\partial x_{12}} & \frac{\partial y}{\partial x_{22}} & \cdots & \frac{\partial y}{\partial x_{n2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y}{\partial x_{1n}} & \frac{\partial y}{\partial x_{2n}} & \cdots & \frac{\partial y}{\partial x_{m}} \end{bmatrix}$$
(8)

image.png



,此时的导数是**梯度矩阵**,可以表示为下式:

# 5. 维度分析 (没怎么看懂)

• 当我们对一些复杂的矩阵乘积求偏导的时候,直接求很难直接求出,这时候我们可以通过分析 矩阵的维度来得到结果。例如:

考虑以下导数  $\frac{\partial \mathbf{u}}{\partial \mathbf{u}}$ ,其中  $\mathbf{A}$  与 $\mathbf{x}$  无关 且有  $\mathbf{A} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{u} \in BbbR^{n \times 1}$  ,  $\mathbf{x} \in \mathbb{R}^{p \times 1}$  , 我们知道结果肯定和  $\frac{\partial \mathbf{u}}{\partial \mathbf{u}}$  有关,于是先把  $\mathbf{A}$  提出 求导式,至于到了哪暂时不知道,接着我们知道  $\frac{\partial \mathbf{u}}{\partial x} \in \mathbb{R}^{p \times n}$ ,于是  $\mathbf{A}$  只能转置后添加到后面。因此有

$$\frac{\partial \mathbf{A}\mathbf{u}}{\partial \mathbf{v}} = \$ \frac{\partial \mathbf{u}}{\partial \mathbf{v}} \mathbf{A}^{\top}$$
(9)

再考虑问题  $\frac{\partial \mathbf{x}^{\mathsf{T}} \mathbf{A} \mathbf{x}}{\partial \mathbf{x}}$  , 其中  $\mathbf{A} \in \mathbb{R}^{n \times n}$  ,  $\mathbf{x} \in \mathbb{R}^{n \times n}$ 为了分析这个问题我们考虑一个更一半的问题

$$\frac{\partial \mathbf{x}^{\top} \mathbf{A} \mathbf{x}}{\partial \mathbf{x}} \tag{10}$$

其中  $\mathbf{A} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{x} \in BbbR^{n \times n}$ , 且  $\mathbf{A}$  与 $\mathbf{x}$  和  $\mathbf{v}$  无关。于是我们利用维度分析,采用非精确的乘积法则,可以将它分为两个部分

$$\frac{\partial (\mathbf{x}^{\mathsf{T}} \mathbf{A}) \mathbf{y}}{\partial \mathbf{x}} \tag{11}$$

于是结果与两部分相关。 一个是

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} \in \mathbb{R}^{m \times n}$$
 (12)

另一个是

$$\frac{\partial \mathbf{x}^T \mathbf{A}}{\partial \mathbf{x}} = \mathbf{A} \in \mathbb{R}^{m \times n}$$
(13)

同样通过维度分析,我们可以得到

$$\frac{\partial (\mathbf{x}^{\top} \mathbf{A})\mathbf{y}}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \mathbf{A}^{\top} \mathbf{x} + \mathbf{A} \mathbf{y}$$
(14)

因此经过维度的比较我们可以得到

$$\frac{\partial \mathbf{X}^{\mathsf{T}} \mathbf{A} \mathbf{X}}{\partial \mathbf{X}} = (\mathbf{A}^{\mathsf{T}} + \mathbf{A}) \mathbf{X} \tag{14}$$

通过以上两个示例的学习,我们可以知道在求解复杂矩阵的求导问题时,通过维度来判断矩阵的导数形式很简便同时也不容易出错。下图 是机器学习中常见的矩阵求导形式,可供参考:

image.png

#### 推荐阅读

推荐系统召回算法之——图模型 (Personal Rank)

阅读 443

机器学习高频面试题(41道)

阅读 3.720

YOLOv2网络

阅读 375

python用线性回归预测时间序列股票 价格

阅读 1,223

基于树模型的集成算法---Random Forest

阅读 1,001



写下你的评论...

评论2

赞15

# 矩阵求导 (一)

● 赞 ■ 回复



穆新星

3楼 03.27 10:16

3.3好像不对

▲ 2 ■ 回复



thinkando ( 关注

赞赏支持

#### 推荐阅读

推荐系统召回算法之——图模型

(Personal Rank)

阅读 443

机器学习高频面试题(41道)

阅读 3,720

YOLOv2网络

阅读 375

python用线性回归预测时间序列股票

价格

阅读 1,223

基于树模型的集成算法---Random

Forest

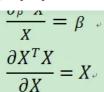
阅读 1,001





写下你的评论...

# 矩阵求导 (一)



$$\frac{\partial X}{\partial X^T A X} = (A + A^T) X_{e}$$

Andrew Ng 推荐的使用矩阵的迹的相关公式:

$$tr(a) = a$$

$$tr(AB) = tr(BA)$$

$$tr(ABC) = tr(CAB) = tr(BCA)_{\leftarrow}$$

$$\frac{\partial tr(AB)}{\partial A} = B^{T_{\phi}}$$

$$\operatorname{tr}(\mathbf{A}) = \operatorname{tr}(\mathbf{A}^T)_{\boldsymbol{\varphi}}$$

 $\partial tr(ABA^TC)$ 

image.png



thinkando( 关注 )



赞赏支持

#### 推荐阅读

推荐系统召回算法之——图模型

(Personal Rank)

阅读 443

机器学习高频面试题(41道)

阅读 3,720

YOLOv2网络

阅读 375

python用线性回归预测时间序列股票

价格

阅读 1,223

基于树模型的集成算法---Random

Forest

阅读 1,001



#### 参考文献

1. https://blog.csdn.net/u010976453/article/details/54381248



15人点赞>



■ 数学! 统计! …



"如能帮您解决问题,请随意打赏哈@"

赞赏支持

还没有人赞赏, 支持一下



thinkando 知行合一, 浩然正气; 没有记录就没有开始

总资产486 (约39.95元) 共写了32.3W字 获得2,048个赞 共1,815个粉丝



#### 终极在线保护

数字保护产品现在折上加折。购买 NordVPN 2 年套餐,享受三折优



写下你的评论...



写下你的评论...

按时间倒序 按时间正序



MLearner1

