

第六章 向量与矩阵微分

第 18 讲 向量与矩阵微分

黄定江

DaSE @ ECNU

djhuang@dase.ecnu.edu.cn

- ① 18.1 向量和矩阵函数的梯度
- ② 18.2 向量和矩阵函数微分与迹微分法
- ③ 18.3 向量值和矩阵值函数的梯度
- ④ 18.4 链式法则与一些有用的梯度公式
- ⑤ 18.5 反向传播与自动微分
- ⑥ 18.6 高阶导数与泰勒展开

- 1 18.1 向量和矩阵函数的梯度
- 2 18.2 向量和矩阵函数微分与迹微分法
- 3 18.3 向量值和矩阵值函数的梯度
- 4 18.4 链式法则与一些有用的梯度公式
- 5 18.5 反向传播与自动微分
- 6 18.6 高阶导数与泰勒展开

18.1.1 向量和矩阵函数梯度：引入

机器学习模型的求解通常会转变成为一个优化问题：

例 1

- 逻辑回归：

$$\min_{\mathbf{w}} \sum_{i=1}^N [y_i(\mathbf{w}^T \mathbf{x}_i) - \log(1 + \exp(\mathbf{w}^T \mathbf{x}_i))]$$

- 线性可分支持向量机模型：

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 > 0 \end{aligned}$$

- PCA:

$$\begin{aligned} \min_{\mathbf{W}} \quad & -\text{Tr}(\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W}) \\ \text{s.t.} \quad & \mathbf{W}^T \mathbf{W} = \mathbf{I} \end{aligned}$$

例 2

在深度学习中我们可能会构造一个两层的神经网络

$$\mathbf{h} = \text{ReLU}(\mathbf{A}_1 \mathbf{x} + \mathbf{b}_1)$$

$$\mathbf{y}' = \text{ReLU}(\mathbf{A}_2 \mathbf{h} + \mathbf{b}_2)$$

并且我们有关于数据集的标签向量 \mathbf{y} , 那么我们需要求解以下优化问题:

$$\min \|\mathbf{y} - \mathbf{y}'\|_2^2$$

- 上述例子中优化的目标函数都是向量函数或者矩阵函数，优化问题的求解通常都需要利用到函数的梯度信息，对于像牛顿法这种二阶方法还需要知道函数的 Hessian 矩阵，而且这些函数都是多元函数，含有的变量非常多。
- 例如在深度学习领域，2019 年 OpenAI 开放了一个文本生成模型 GPT-2，有 7.74 亿个参数，而完整模型则有 15 亿的参数，这就意味着我们需要求解同等规模的梯度，如果要一个一个去计算他们的偏导数是不可能的。
- 本讲将主要介绍如何使用一些较为方便的方法来求解梯度或者 Hessian 矩阵。

18.1.2 向量函数梯度：一元函数导数回顾

定义 1

函数 $f: \mathbb{R} \rightarrow \mathbb{R}$ 关于 x 的导数定义为

$$\frac{df}{dx} := \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

定义 2

函数 $f: \mathbb{R} \rightarrow \mathbb{R}$ 在 x_0 的 n 阶泰勒多项式为

$$T_n(x) := \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k$$

一元函数泰勒级数与多元函数偏导数

定义 3

光滑函数 $f: \mathbb{R} \rightarrow \mathbb{R}, f \in \mathbb{C}^\infty$ 在 x_0 处的泰勒级数为

$$T_\infty(x) := \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k$$

定义 4

函数 $f(\mathbf{x}): \mathbb{R}^n \rightarrow \mathbb{R}$ 关于 \mathbf{x} 的 n 个分量的偏导为

$$\begin{aligned} \frac{\partial f}{\partial x_1} &= \lim_{h \rightarrow 0} \frac{f(x_1 + h, x_2, \dots, x_n) - f(\mathbf{x})}{h} \\ &\vdots \\ \frac{\partial f}{\partial x_n} &= \lim_{h \rightarrow 0} \frac{f(x_1, x_2, \dots, x_n + h) - f(\mathbf{x})}{h} \end{aligned}$$

向量函数梯度

定义 5

若 $\mathbf{x} \in \mathbb{R}^n$, $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ 是一实值函数, 其中 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$, 则定义

$$\frac{\partial}{\partial \mathbf{x}} f = \nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

例 3

假设函数 $f(\mathbf{x}) : \mathbb{R}^2 \rightarrow \mathbb{R}$ 为

$$f(\mathbf{x}) = \sin x_1 + 2x_1x_2 + x_2^2$$

其中 $\mathbf{x} = (x_1, x_2)^T$, 则 f 的偏导数分别为

$$\frac{\partial f}{\partial x_1}(\mathbf{x}) = \cos x_1 + 2x_2$$

$$\frac{\partial f}{\partial x_2}(\mathbf{x}) = 2x_1 + 2x_2$$

因此梯度为

$$\nabla f(\mathbf{x}) = (\cos x_1 + 2x_2, 2x_1 + 2x_2)^T$$

例 4

设 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$, $\mathbf{a} = (a_1, a_2, \dots, a_n)^T \in \mathbb{R}^n$, $\mathbf{b} = (b_1, b_2, \dots, b_n)^T \in \mathbb{R}^n$ 以及 $f(x_1, x_2, \dots, x_n) = f(\mathbf{x}) = \mathbf{a}^T \mathbf{x} + \mathbf{b}$, 求 $f(\mathbf{x})$ 的梯度 $\nabla f(\mathbf{x})$ 。

将 $f(\mathbf{x})$ 写成分量的形式:

$$f(\mathbf{x}) = \mathbf{a}^T \mathbf{x} + \mathbf{b} = \sum_{i=1}^n a_i x_i + b_i$$

那么 $f(\mathbf{x})$ 对第 h 个分量的偏导数为

$$\frac{\partial(\mathbf{a}^T \mathbf{x} + \mathbf{b})}{\partial x_h} = a_h$$

从而就有

$$\nabla f = \mathbf{a}$$

例 5

设 $\mathbf{p} \in \mathbb{R}^n$ 是 \mathbb{R}^n 中的一个点, 函数 $f(\mathbf{x})$ 表示点 \mathbf{x} 和 \mathbf{p} 的距离:

$$f(\mathbf{x}) = \|\mathbf{x} - \mathbf{p}\|_2 = \sqrt{\sum_{i=1}^n (x_i - p_i)^2}$$

函数 $f(\mathbf{x})$ 在 $\mathbf{x} \neq \mathbf{p}$ 处处可微, 并且梯度为

$$\nabla f(\mathbf{x}) = \frac{1}{\|\mathbf{x} - \mathbf{p}\|_2} (\mathbf{x} - \mathbf{p})$$

18.1.3 向量函数导数的运算法则

与一元函数类似，向量函数导数有如下运算法则：

- 线性法则：若 $f(\mathbf{x})$ 和 $g(\mathbf{x})$ 分别是向量 \mathbf{x} 的实值函数， c_1 和 c_2 为实常数，则

$$\frac{\partial [c_1 f(\mathbf{x}) + c_2 g(\mathbf{x})]}{\partial \mathbf{x}} = c_1 \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} + c_2 \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \quad (1)$$

- 乘法法则：若 $f(\mathbf{x})$ 和 $g(\mathbf{x})$ 都是向量 \mathbf{x} 的实值函数，则

$$\frac{\partial f(\mathbf{x}) g(\mathbf{x})}{\partial \mathbf{x}} = g(\mathbf{x}) \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} + f(\mathbf{x}) \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \quad (2)$$

- 商法则：若 $g(\mathbf{x}) \neq 0$ ，则

$$\frac{\partial f(\mathbf{x}) / g(\mathbf{x})}{\partial \mathbf{x}} = \frac{1}{g^2(\mathbf{x})} \left[g(\mathbf{x}) \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} - f(\mathbf{x}) \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \right] \quad (3)$$

关于复合函数的链式法则我们在 18.4 节再进行介绍。

18.1.4 矩阵函数的梯度

定义 6

若 $\mathbf{A} \in \mathbb{R}^{n \times m}$, $f(\mathbf{A}) : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}$ 是一实值函数, 其中 $\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}$,

则定义矩阵函数的梯度为

$$\frac{\partial}{\partial \mathbf{A}} f = \begin{pmatrix} \frac{\partial f}{\partial a_{11}} & \frac{\partial f}{\partial a_{12}} & \cdots & \frac{\partial f}{\partial a_{1m}} \\ \frac{\partial f}{\partial a_{21}} & \frac{\partial f}{\partial a_{22}} & \cdots & \frac{\partial f}{\partial a_{2m}} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f}{\partial a_{n1}} & \frac{\partial f}{\partial a_{n2}} & \cdots & \frac{\partial f}{\partial a_{nm}} \end{pmatrix}$$

例 6

令 $f: \mathbb{R}^{n \times m} \rightarrow \mathbb{R}, f(\mathbf{A}) = \sum_{i,j} a_{ij}$, 其中 a_{ij} 为矩阵 \mathbf{A} 的第 ij 个元素, 求 $\frac{\partial f}{\partial \mathbf{A}}$ 。

解

我们对每一分量进行求导可得

$$\frac{\partial f}{\partial a_{ij}} = 1$$

故根据定义6, 则有

$$\frac{\partial f}{\partial \mathbf{A}} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix}$$

- 注意在向量函数梯度定义5中 \mathbf{x} 是一列向量。
- 若将行向量和列向量均看做矩阵的特殊情况，则我们只需给出矩阵函数梯度的定义6，由此可导出向量函数梯度定义5。
- 通过定义6我们可以自然地导出对 \mathbf{x}^T 求偏导的结果。

定理 1

若 $\mathbf{x} \in \mathbb{R}^n, f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ 是一实值函数, 则有

$$\frac{\partial}{\partial \mathbf{x}^T} f = \left(\frac{\partial}{\partial \mathbf{x}} f \right)^T$$

证明.

通过定义6, 有

$$\frac{\partial}{\partial \mathbf{x}^T} f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}^T = \left(\frac{\partial}{\partial \mathbf{x}} f \right)^T$$



例 7

在例4中我们考虑了一个非常简单的多元线性函数 $f(\mathbf{x}) = \mathbf{a}^T \mathbf{x} + \mathbf{b}$, 我们知道

$$\frac{\partial f}{\partial \mathbf{x}} = \mathbf{a}$$

利用上述定理我们有

$$\frac{\partial f}{\partial \mathbf{x}^T} = \left(\frac{\partial f}{\partial \mathbf{x}} \right)^T = \mathbf{a}^T$$

注意我们在这个例子中实际上仅仅使用了定义。之后我们将使用矩阵性质来展示相同的结果, 并且不需要使用 $\frac{\partial f}{\partial \mathbf{x}}$ 作为桥梁。

例 8

对于一个可分的支持向量机, 相应的优化问题为

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 > 0 \end{aligned}$$

我们考虑其目标函数的梯度

$$\frac{\partial}{\partial \mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2$$

我们逐分量地求其偏导数有

$$\frac{\partial}{\partial w_i} \frac{1}{2} \|\mathbf{w}\|^2 = \frac{\partial}{\partial w_i} \frac{1}{2} \sum_{i=1}^n w_i^2 = \underline{w_i}$$

所以

$$\frac{\partial}{\partial \mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2 = \underline{\mathbf{w}}$$

18.1.5 矩阵函数导数的运算法则

实值函数相对于矩阵变元的梯度具有以下性质。

- 线性法则: 若 $f(\mathbf{A})$ 和 $g(\mathbf{A})$ 分别是矩阵 \mathbf{A} 的实值函数, c_1 和 c_2 为实常数, 则

$$\frac{\partial [c_1 f(\mathbf{A}) + c_2 g(\mathbf{A})]}{\partial \mathbf{A}} = c_1 \frac{\partial f(\mathbf{A})}{\partial \mathbf{A}} + c_2 \frac{\partial g(\mathbf{A})}{\partial \mathbf{A}}$$

- 乘积法则: 若 $f(\mathbf{A})$, $g(\mathbf{A})$ 和 $h(\mathbf{A})$ 分别是矩阵 \mathbf{A} 的实值函数, 则

$$\frac{\partial f(\mathbf{A}) g(\mathbf{A})}{\partial \mathbf{A}} = g(\mathbf{A}) \frac{\partial f(\mathbf{A})}{\partial \mathbf{A}} + f(\mathbf{A}) \frac{\partial g(\mathbf{A})}{\partial \mathbf{A}}$$

- 商法则: 若 $g(\mathbf{A}) \neq 0$, 则

$$\frac{\partial f(\mathbf{A}) / g(\mathbf{A})}{\partial \mathbf{A}} = \frac{1}{g^2(\mathbf{A})} \left[g(\mathbf{A}) \frac{\partial f(\mathbf{A})}{\partial \mathbf{A}} - f(\mathbf{A}) \frac{\partial g(\mathbf{A})}{\partial \mathbf{A}} \right]$$

- 1 18.1 向量和矩阵函数的梯度
- 2 18.2 向量和矩阵函数微分与迹微分法**
- 3 18.3 向量值和矩阵值函数的梯度
- 4 18.4 链式法则与一些有用的梯度公式
- 5 18.5 反向传播与自动微分
- 6 18.6 高阶导数与泰勒展开

18.2.1 矩阵的微分：定义

尽管大多数时候我们想要的是矩阵导数，但是因为微分形式不变性，将问题转化为求矩阵微分会更容易求解。

定义 7

设 $\mathbf{A} \in \mathbb{R}^{m \times n}$, 矩阵 \mathbf{A} 的微分定义为

$$d\mathbf{A} = \begin{pmatrix} da_{11} & da_{12} & \dots & da_{1n} \\ da_{21} & da_{22} & \dots & da_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ da_{m1} & da_{m2} & \dots & da_{mn} \end{pmatrix}$$

与上面类似，我们也可以将矩阵微分的定义推广到向量上。

定义 8

设 $\boldsymbol{x} \in \mathbb{R}^n$ ，向量 \boldsymbol{x} 的微分定义为

$$\underline{d\boldsymbol{x}} = \begin{pmatrix} dx_1 \\ dx_2 \\ \vdots \\ dx_n \end{pmatrix}; d\boldsymbol{x}^T = (dx_1, dx_2, \dots, dx_n)$$

18.2.1 矩阵微分的性质

性质 1

矩阵微分有如下性质

- $d(cA) = cdA$ 其中 $A \in \mathbb{R}^{n \times m}$
- $d(A + B) = dA + dB$ 其中 $A, B \in \mathbb{R}^{n \times m}$
- $d(AB) = dAB + AdB$ 其中 $A \in \mathbb{R}^{n \times m}, B \in \mathbb{R}^{m \times k}$
- $dA^T = (dA)^T$ 其中 $A \in \mathbb{R}^{n \times m}$

证明.

这些性质都能通过矩阵微分的定义自然推出, 我们只在这里证明第 3 个性质。

注意等式成立需要两边每一个对应元素都相等, 我们考虑两边的第 ij 个元素, 并记 A, B 的第 ij 个元素分别为 a_{ij}, b_{ij} 。

$$\begin{aligned}
 \text{左边}_{ij} &= d \left(\sum_k a_{ik} b_{kj} \right) \\
 &= \sum_k (da_{ik} b_{kj} + a_{ik} db_{kj}) \\
 \text{右边}_{ij} &= (dA)_{ij} + (A dB)_{ij} \\
 &= \sum_k da_{ik} b_{kj} + \sum_k a_{ik} db_{kj} \\
 &= \text{左边}_{ij}
 \end{aligned}$$

□

定理 2

微分运算和迹运算可交换, 即设 $\mathbf{A} \in \mathbb{R}^{n \times n}$, 则

$$d\text{Tr}(\mathbf{A}) = \text{Tr}(d\mathbf{A})$$

证明.

$$\begin{aligned} \text{左边} &= d\left(\sum_i a_{ii}\right) = \sum_i da_{ii} \\ \text{右边} &= \text{Tr} \begin{bmatrix} da_{11} & da_{12} & \cdots & da_{1n} \\ da_{21} & da_{22} & \cdots & da_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ da_{n1} & da_{n2} & \cdots & da_{nn} \end{bmatrix} = \sum_i da_{ii} = \text{左边} \end{aligned}$$



18.2.2 矩阵微分与偏导数的联系

回顾迹函数的性质：

性质 2

1. $\text{Tr}(\mathbf{A} + \mathbf{B}) = \text{Tr}(\mathbf{A}) + \text{Tr}(\mathbf{B})$ 其中 $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$
2. $\text{Tr}(c\mathbf{A}) = c\text{Tr}(\mathbf{A})$ 其中 $\mathbf{A} \in \mathbb{R}^{n \times n}, c \in \mathbb{R}$
3. $\text{Tr}(\mathbf{AB}) = \text{Tr}(\mathbf{BA})$ 其中 $\mathbf{A} \in \mathbb{R}^{n \times m}, \mathbf{B} \in \mathbb{R}^{m \times n}$
4. $\text{Tr}(\mathbf{A}_1 \mathbf{A}_2 \dots \mathbf{A}_n) = \text{Tr}(\mathbf{A}_n \mathbf{A}_1 \dots \mathbf{A}_{n-1})$ 其中 $\mathbf{A}_1 \in \mathbb{R}^{c_n, c_1}; \mathbf{A}_i \in \mathbb{R}^{c_{i-1} \times c_i}, i = 2, 3, \dots, n$
5. $\text{Tr}(\mathbf{A}^T \mathbf{B}) = \sum_i \sum_j \mathbf{A}_{ij} \mathbf{B}_{ij}$ 其中 $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times m}$
6. $\text{Tr}(\mathbf{A}) = \text{Tr}(\mathbf{A}^T)$ 其中 $\mathbf{A} \in \mathbb{R}^{n \times n}$

多元函数的微分和偏导的关系如下

$$df(x_1, x_2, \dots, x_n) = \frac{\partial f}{\partial x_1} dx_1 + \frac{\partial f}{\partial x_2} dx_2 + \dots + \frac{\partial f}{\partial x_n} dx_n$$

这里 df 是一个标量，从分量的角度来看， df 就是将 $\frac{\partial f}{\partial x}$ 与 dx 相同位置的元素相乘后再求和。

我们希望对于矩阵微分与偏导数能够得到一个类似的形式。此时我们可以利用迹函数第 5 条性质来给出下面这个定理：

定理 3

对于实值函数 $f: \mathbb{R}^{n \times m} \rightarrow \mathbb{R}$ 和 $A \in \mathbb{R}^{n \times m}$ 有

$$df = \text{Tr} \left[\left(\frac{\partial f}{\partial A} \right)^T dA \right]$$

证明.

$$\begin{aligned}
 \text{左边} &= df = \sum_{ij} \frac{\partial f}{\partial x_{ij}} dx_{ij} \\
 \text{右边} &= \text{Tr} \left[\left(\frac{\partial f}{\partial \mathbf{A}} \right)^T d\mathbf{A} \right] \\
 &= \sum_{ij} \left(\frac{\partial f}{\partial \mathbf{A}} \right)_{ij} (d\mathbf{A})_{ij} \\
 &= \sum_{ij} \frac{\partial f}{\partial x_{ij}} dx_{ij} = \text{左边}
 \end{aligned}$$

□

注意对于向量也有类似的结果。这里不再叙述。

18.2.3 迹微分法

- 迹函数在处理矩阵微分的问题中具有很重要的地位。下面我们将给出一种利用迹函数和矩阵微分来求解实值函数的梯度的方法——迹微分法。
- 我们知道对于一个标量 c 来说 $c = \text{Tr}(c)$ ，这也就意味着对于一个实值函数 $f(\mathbf{A})$ 有 $f(\mathbf{A}) = \text{Tr}(f(\mathbf{A}))$ 从而就有 $df(\mathbf{A}) = d\text{Tr}(f(\mathbf{A})) = \text{Tr}(df(\mathbf{A}))$ 。
- 通过矩阵微分与迹运算的交换性、迹函数性质、矩阵微分的性质以及定理3我们可以总结出如下迹微分法：

1. $df = d\text{Tr}(f) = \text{Tr}(df)$

2. 使用迹函数的性质和矩阵微分的性质来得到如下形式

$$df = \text{Tr}(\mathbf{A}^T d\mathbf{x})$$

3. 应用定理3得到结果

$$\frac{\partial f}{\partial \mathbf{x}} = \mathbf{A}$$

例 9

给定函数 $f(x) = x^T A x$, 其中 A 是一方阵, x 是一列向量, 我们计算

$$\begin{aligned}
 df &= d\text{Tr}(x^T A x) \\
 &= \text{Tr}(d(x^T A x)) \\
 &= \text{Tr}(d(x^T) A x + x^T d(A x)) \\
 &= \text{Tr}(d(x^T) A x + x^T dA x + x^T A dx) \\
 &= \text{Tr}(dx^T A x + x^T A dx) \\
 &= \text{Tr}(dx^T A x) + \text{Tr}(x^T A dx) \\
 &= \text{Tr}(x^T A^T dx) + \text{Tr}(x^T A dx) \\
 &= \text{Tr}(x^T A^T dx + x^T A dx) \\
 &= \text{Tr}((x^T A^T + x^T A) dx)
 \end{aligned}$$

我们可以得到

$$\frac{\partial f}{\partial \mathbf{x}} = \left(\mathbf{x}^T \mathbf{A}^T + \mathbf{x}^T \mathbf{A} \right)^T = \mathbf{A} \mathbf{x} + \mathbf{A}^T \mathbf{x}$$

如果 \mathbf{A} 是对称矩阵，我们还可以将其简化为

$$\frac{\partial f}{\partial \mathbf{x}} = 2 \mathbf{A} \mathbf{x}$$

令 $\mathbf{A} = \mathbf{I}$ 我们则有

$$\frac{\partial (\mathbf{x}^T \mathbf{x})}{\partial \mathbf{x}} = 2 \mathbf{x}$$

例 10

根据上面的推导可以知道，在谱聚类中我们要求解的优化问题

$$\begin{aligned} \min_x & \mathbf{x}^T \mathbf{L} \mathbf{x} \\ \text{s.t.} & \mathbf{x}^T \mathbf{1} = 0 \end{aligned}$$

中目标函数的梯度为

$$\nabla_{\mathbf{x}} \mathbf{x}^T \mathbf{L} \mathbf{x} = 2\mathbf{L}\mathbf{x}$$

我们再看一个关于矩阵函数的例子。

例 11

在 PCA 中, 我们需要求解优化问题

$$\begin{aligned} \min_W & -\text{Tr}(\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W}) \\ \text{s.t. } & \mathbf{W}^T \mathbf{W} = \mathbf{I} \end{aligned}$$

我们现在考虑求梯度 $\nabla_W -\text{Tr}(\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W})$ 。

解

利用迹微分法有

$$\begin{aligned} d(-\text{Tr}(\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W})) &= -\text{Tr}(d(\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W})) \\ &= -2\text{Tr}(\mathbf{W}^T \mathbf{X} \mathbf{X}^T d\mathbf{W}) \end{aligned}$$

所以

$$\nabla_W -\text{Tr}(\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W}) = -2\mathbf{X} \mathbf{X}^T \mathbf{W}$$

18.2.4 含 F 范数的函数

我们可以使用迹微分法来处理含 F 范数的函数。

例 12

设 $\mathbf{A} \in \mathbb{R}^{n \times m}$, 求 $\frac{\partial \|\mathbf{A}\|_F^2}{\partial \mathbf{A}}$, 其中 $\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m a_{ij}^2}$

解

$$\begin{aligned} d\|\mathbf{A}\|_F^2 &= d\text{Tr}(\mathbf{A}^T \mathbf{A}) \\ &= \text{Tr}(d(\mathbf{A}^T \mathbf{A})) \\ &= \text{Tr}((d\mathbf{A})^T \mathbf{A}) + \text{Tr}(\mathbf{A}^T d\mathbf{A}) \\ &= \text{Tr}(2\mathbf{A}^T d\mathbf{A}) \end{aligned}$$

故

$$\frac{\partial \|\mathbf{A}\|_F^2}{\partial \mathbf{A}} = 2\mathbf{A}$$

18.2.5 逆矩阵

对于一个非奇异方阵 \mathbf{X} , 我们有 $\mathbf{X}\mathbf{X}^{-1} = \mathbf{I}$ 对两边同时作用于微分, 则有

$$0 = d\mathbf{I} = d(\mathbf{X}\mathbf{X}^{-1}) = d\mathbf{X}\mathbf{X}^{-1} + \mathbf{X}d(\mathbf{X}^{-1})$$

整理可得

$$d(\mathbf{X}^{-1}) = -\mathbf{X}^{-1}d\mathbf{X}\mathbf{X}^{-1}$$

这样我们就得到了关于逆矩阵微分的一个结论。

例 13

若 $A \in \mathbb{R}^{n \times n}$, 非奇异, $x \in \mathbb{R}^{n \times 1}$, $y \in \mathbb{R}^{n \times 1}$ 求

$$\frac{\partial x^T A^{-1} y}{\partial A}$$

解

$$\begin{aligned} d(x^T A^{-1} y) &= \text{Tr}(d(x^T A^{-1} y)) \\ &= \text{Tr}(x^T dA^{-1} y) \\ &= \text{Tr}(-x^T A^{-1} dA A^{-1} y) \\ &= \text{Tr}(-A^{-1} y x^T A^{-1} dA) \end{aligned}$$

所以

$$\frac{\partial x^T A^{-1} y}{\partial A} = -A^{-T} x y^T A^{-T}$$

例 14

设函数 $f(\mathbf{X}) = \|\mathbf{A}\mathbf{X}^{-1}\|_F^2$, 其中 $\mathbf{A} \in \mathbb{R}^{n \times m}$, $\mathbf{X} \in \mathbb{R}^{m \times m}$ 且 \mathbf{X} 可逆, 求 $\frac{\partial f}{\partial \mathbf{X}}$

解

$$\begin{aligned}
 f(\mathbf{X}) &= \text{Tr}(\mathbf{X}^{-T} \mathbf{A}^T \mathbf{A} \mathbf{X}^{-1}) \\
 df(\mathbf{X}) &= \text{Tr}[d(\mathbf{X}^{-T} \mathbf{A}^T \mathbf{A} \mathbf{X}^{-1})] \\
 &= \text{Tr}(d\mathbf{X}^{-T} \mathbf{A}^T \mathbf{A} \mathbf{X}^{-1} + \mathbf{X}^{-T} \mathbf{A}^T \mathbf{A} d\mathbf{X}^{-1}) \\
 &= \text{Tr}(2\mathbf{X}^{-T} \mathbf{A}^T \mathbf{A} d\mathbf{X}^{-1}) \\
 &= \text{Tr}(-2\mathbf{X}^{-T} \mathbf{A}^T \mathbf{A} \mathbf{X}^{-1} d\mathbf{X} \mathbf{X}^{-1}) \\
 &= \text{Tr}(-2\mathbf{X}^{-1} \mathbf{X}^{-T} \mathbf{A}^T \mathbf{A} \mathbf{X}^{-1} d\mathbf{X})
 \end{aligned}$$

故

$$\frac{\partial f}{\partial \mathbf{X}} = -2\mathbf{X}^{-T} \mathbf{A}^T \mathbf{A} \mathbf{X}^{-1} \mathbf{X}^{-T}$$

18.2.6 行列式

行列式也是关于矩阵的一个实值函数，有时我们会面临求 $\frac{\partial |\mathbf{A}|}{\partial \mathbf{A}}$ 。我们首先回顾一下行列式相关的一些概念，假设矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ ，则：

- 余子式 M_{ij} 是矩阵 \mathbf{A} 划去第 i 行 j 列元素组成的矩阵的行列式
- 第 ij 个元素的代数余子式定义为 $A_{ij} = (-1)^{i+j} M_{ij}$
- 如果我们将行列式按第 i 行展开，则有 $|\mathbf{A}| = \sum_j a_{ij} A_{ij}$
- \mathbf{A} 的伴随矩阵被定义为 $\mathbf{A}_{ij}^* = A_{ji}$
- 对于非奇异矩阵 \mathbf{A} 有 $\mathbf{A}^{-1} = \frac{\mathbf{A}^*}{|\mathbf{A}|}$

定理 4

设矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 则有

$$\frac{\partial |\mathbf{A}|}{\partial \mathbf{A}} = (\mathbf{A}^*)^T$$

证明.

为了计算 $\frac{\partial |\mathbf{A}|}{\partial \mathbf{A}}$, 我们利用定义6逐元素进行求导。将行列式按第 i 行展开, 则有

$$\frac{\partial |\mathbf{A}|}{\partial a_{ij}} = \frac{\partial \left(\sum_j a_{ij} A_{ij} \right)}{\partial a_{ij}} = A_{ij}$$

使用定义6来组织元素就有

$$\frac{\partial |\mathbf{A}|}{\partial \mathbf{A}} = (\mathbf{A}^*)^T$$



如果矩阵 \mathbf{A} 非奇异, 则可以进一步推出

$$\frac{\partial |\mathbf{A}|}{\partial \mathbf{A}} = \underbrace{(|\mathbf{A}| \mathbf{A}^{-1})^T}_{= |\mathbf{A}| (\mathbf{A}^{-1})^T}$$

通过上述偏导的结果和定理3, 我们还能够给出对应的微分关系

定理 5

设矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 则有

$$d|\mathbf{A}| = \text{Tr}(\mathbf{A}^* d\mathbf{A})$$

当 \mathbf{A} 可逆时有

$$d|\mathbf{A}| = \text{Tr}(\underbrace{|\mathbf{A}| \mathbf{A}^{-1}}_{= \frac{\partial |\mathbf{A}|}{\partial \mathbf{A}}}) d\mathbf{A}$$

证明.

$$\begin{aligned} d|A| &= \text{Tr} \left(\left(\frac{\partial |A|}{\partial A} \right)^T dA \right) \\ &= \text{Tr} \left(((A^*)^T)^T dA \right) \\ &= \text{Tr}(A^* dA) \end{aligned}$$

当 A 可逆时有

$$d|A| = \text{Tr}(A^* dA) = \text{Tr}(|A| A^{-1} dA)$$



例 15

设矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 是一可逆矩阵。求

$$\frac{\partial |\mathbf{A}^{-1}|}{\partial \mathbf{A}}$$

解

应用定理5有

$$\begin{aligned} d|\mathbf{A}^{-1}| &= \text{Tr}(|\mathbf{A}^{-1}| \mathbf{A} d\mathbf{A}^{-1}) \\ &= \text{Tr}(-|\mathbf{A}^{-1}| \mathbf{A} \mathbf{A}^{-1} d\mathbf{A} \mathbf{A}^{-1}) \\ &= \text{Tr}(-|\mathbf{A}^{-1}| \mathbf{A}^{-1} d\mathbf{A}) \end{aligned}$$

故

$$\frac{\partial |\mathbf{A}^{-1}|}{\partial \mathbf{A}} = -|\mathbf{A}^{-1}| \mathbf{A}^{-T} = -|\mathbf{A}|^{-1} \mathbf{A}^{-T}$$

- 1 18.1 向量和矩阵函数的梯度
- 2 18.2 向量和矩阵函数微分与迹微分法
- 3 18.3 向量值和矩阵值函数的梯度**
- 4 18.4 链式法则与一些有用的梯度公式
- 5 18.5 反向传播与自动微分
- 6 18.6 高阶导数与泰勒展开

18.3.1 向量值函数的梯度: Jacobian 矩阵

我们上面已经讨论函数实值函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ 的偏导和梯度。接下来, 我们将给出向量值函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}^m, (n, m \geq 1)$ 的梯度的概念。

对于一个函数 $\mathbf{f}: \mathbb{R}^n \rightarrow \mathbb{R}^m$ 和一个向量 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$, 那么对应的函数值为

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{pmatrix} \in \mathbb{R}^m$$

这样写能够更好地展示一个向量值函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$, 它就相当于一个函数的向量 $(f_1, f_2, \dots, f_m)^T, f_i: \mathbb{R}^n \rightarrow \mathbb{R}$ 。

因此，应用前面已经讨论过了关于其中任一个 f_i 的微分法则，我们可得向量值函数 \mathbf{f} 关于 x_i 的偏导数：

$$\frac{\partial \mathbf{f}}{\partial x_i} = \begin{pmatrix} \frac{\partial f_1}{\partial x_i} \\ \vdots \\ \frac{\partial f_m}{\partial x_i} \end{pmatrix} = \begin{pmatrix} \lim_{h \rightarrow 0} \frac{f_1(x_1, \dots, x_{i-1}, x_i + h, x_{i+1}, \dots, x_m) - f_1(\mathbf{x})}{h} \\ \vdots \\ \lim_{h \rightarrow 0} \frac{f_m(x_1, \dots, x_{i-1}, x_i + h, x_{i+1}, \dots, x_m) - f_m(\mathbf{x})}{h} \end{pmatrix}$$

在上式中，每一个偏导都是一个列向量。因此，我们按照如下组织得到一个向量值函数的偏导：

$$\frac{d\mathbf{f}(\mathbf{x})}{d\mathbf{x}^T} = \left[\frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_1}, \dots, \frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_n} \right] = \begin{pmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_m(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

定义 9

向量值函数 $\mathbf{f}(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 的所有一阶导数组成的矩阵称为 **Jacobian** 矩阵，它是一个 $m \times n$ 的矩阵，具体定义如下：

$$\mathbf{J} = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}^T} = \begin{pmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_m(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

并且我们定义

$$\frac{\partial \mathbf{f}(\mathbf{x})^T}{\partial \mathbf{x}} = \mathbf{J}^T = \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}^T} \right)^T$$

注意，这里我们没有去定义 $\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}$ 以及 $\frac{\partial \mathbf{f}(\mathbf{x})^T}{\partial \mathbf{x}^T}$ ，所以在后面的讨论中不会出现这两种情况。在计算中也需要注意所计算的形式是否已经被定义。

18.3.2 矩阵值函数的梯度：Jacobian 矩阵

求矩阵关于向量或其它矩阵的梯度，通常会导致一个多维张量。例如，我们计算一个 $m \times n$ 矩阵关于 $p \times q$ 矩阵的梯度，相应的 Jacobian 是 $(p \times q) \times (m \times n)$ ，这是一个四维的张量。

定义 10

函数 $\text{vec} : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^{nm}$ 将一个矩阵按列重排成一个列向量。设 $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m) \in \mathbb{R}^{n \times m}$ 则

$$\text{vec}(\mathbf{A}) = \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_m \end{pmatrix}$$

有了这样一函数之后，我们就可以定义矩阵关于矩阵梯度的 Jacobian 矩阵。

定义 11

设矩阵函数 $F(\mathbf{X}) : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^{q \times p}$ 则其 *Jacobian* 矩阵定义为

$$\mathbf{J} = \frac{\partial \text{vec}(\mathbf{F}(\mathbf{X}))}{\partial \text{vec}(\mathbf{X})^T} = \begin{pmatrix} \frac{\partial f_{11}}{\partial x_{11}} & \frac{\partial f_{11}}{\partial x_{12}} & \cdots & \frac{\partial f_{11}}{\partial x_{nm}} \\ \frac{\partial f_{12}}{\partial x_{11}} & \frac{\partial f_{12}}{\partial x_{12}} & \cdots & \frac{\partial f_{12}}{\partial x_{nm}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{pq}}{\partial x_{11}} & \frac{\partial f_{pq}}{\partial x_{12}} & \cdots & \frac{\partial f_{pq}}{\partial x_{nm}} \end{pmatrix}$$

定义 12

设矩阵 \mathbf{J} 是一 *Jacobian* 矩阵, 则其行列式 $J = |\mathbf{J}|$ 称为 *Jacobian* 行列式。

18.3.3 向量值函数微分

定理 6

设函数 $f(\mathbf{x}) : \mathbb{R}^m \rightarrow \mathbb{R}^n$, $\mathbf{x} \in \mathbb{R}^m$ 则有

$$df = \left(\frac{\partial f^T}{\partial \mathbf{x}} \right)^T d\mathbf{x} = \mathbf{J} d\mathbf{x}$$

证明.

显然, df 有 n 个分量, 所以我们从分量的角度来证明. 考虑第 j 个分量.

$$\begin{aligned}\text{左边}_j &= df_j = \sum_{i=1}^m \frac{\partial f_j}{\partial x_i} dx_i \\ \text{右边}_j &= \left(\left(\frac{\partial f}{\partial \mathbf{x}} \right)^T dx \right)_j \\ &= \sum_{i=1}^m \left(\frac{\partial f}{\partial \mathbf{x}} \right)_{ij} dx_i \\ &= \sum_{i=1}^m \left(\frac{\partial f_j}{\partial x_i} \right) dx_i = \text{左边}_j\end{aligned}$$



注意这个式子在形式上与之前我们推得的定理3是很像的。

利用定理6，仿照求解实值函数梯度的步骤，我们可以简化求解向量对向量的导数。

例 16


考虑向量变换 $\mathbf{x} = \sigma \mathbf{\Lambda}^{-0.5} \mathbf{W}^T \boldsymbol{\eta}$ ， \mathbf{x} 和 $\boldsymbol{\eta}$ 的维数是 d ，其中 σ 是一个实变量， $\mathbf{\Lambda}$ 是一个满秩对角矩阵， \mathbf{W} 是正交矩阵 (即 $\mathbf{W}\mathbf{W}^T = \mathbf{W}^T\mathbf{W} = \mathbf{I}$)，计算 Jacobian 行列式的绝对值。

$$d\mathbf{x} = d(\sigma \mathbf{\Lambda}^{-0.5} \mathbf{W}^T \boldsymbol{\eta}) = \sigma \mathbf{\Lambda}^{-0.5} \mathbf{W}^T d\boldsymbol{\eta}$$


应用定理6我们有

$$\mathbf{J} = \left(\frac{\partial \mathbf{x}^T}{\partial \boldsymbol{\eta}} \right)^T = \sigma \mathbf{\Lambda}^{-0.5} \mathbf{W}^T$$

接着我们利用行列式的性质来计算 Jacobian 行列式 $J = |\mathbf{J}| = \det(\mathbf{J})$ 的绝对值。

$$\begin{aligned}
 |\mathbf{J}| &= |\det(\mathbf{J})| \\
 &= \sqrt{|\det(\mathbf{J})| |\det(\mathbf{J})|} \\
 &= \sqrt{|\det(\mathbf{J})| |\det(\mathbf{J}^T)|} \\
 &= \sqrt{|\det(\mathbf{J}^T \mathbf{J})|} \\
 &= \sqrt{|\det(\mathbf{W} \mathbf{\Lambda}^{-0.5} \sigma \sigma \mathbf{\Lambda}^{-0.5} \mathbf{W}^T)|} \\
 &= \sqrt{|\det(\sigma^2 \mathbf{W} \mathbf{\Lambda}^{-1} \mathbf{W}^T)|}
 \end{aligned}$$


我们令 $\Sigma = \mathbf{W} \mathbf{\Lambda} \mathbf{W}^T$ 我们就能得到一个优美的结果

$$|\mathbf{J}| = |\sigma|^d |\Sigma|^{-1/2}$$


这个结论我们可以应用到多元正态分布的推广中。

定理 7

如果 \mathbf{f} 和 \mathbf{x} 维数相同, 则

$$\left(\frac{\partial \mathbf{f}^T}{\partial \mathbf{x}}\right)^{-1} = \frac{\partial \mathbf{x}^T}{\partial \mathbf{f}}$$

证明.

利用定理6

$$\underline{d\mathbf{f} = \left(\frac{\partial \mathbf{f}^T}{\partial \mathbf{x}}\right)^T d\mathbf{x}} \implies \underline{\left(\left(\frac{\partial \mathbf{f}^T}{\partial \mathbf{x}}\right)^T\right)^{-1} d\mathbf{f} = d\mathbf{x}} \implies d\mathbf{x} = \left(\left(\frac{\partial \mathbf{f}^T}{\partial \mathbf{x}}\right)^{-1}\right)^T d\mathbf{f}$$

所以, 我们就有

$$\frac{\partial \mathbf{x}^T}{\partial \mathbf{f}} = \left(\frac{\partial \mathbf{f}^T}{\partial \mathbf{x}}\right)^{-1}$$



这个结果和标量导数是一致的。这个结论对于变量替换很有用。

- 1 18.1 向量和矩阵函数的梯度
- 2 18.2 向量和矩阵函数微分与迹微分法
- 3 18.3 向量值和矩阵值函数的梯度
- 4 18.4 链式法则与一些有用的梯度公式**
- 5 18.5 反向传播与自动微分
- 6 18.6 高阶导数与泰勒展开

18.4.1 链式法则

回顾对于一元复合函数, 设 $y = f(x)$, $z = g(y)$, 则我们知道

$$\frac{dz}{dx} = \frac{dz}{dy} \frac{dy}{dx}$$

而对于多元复合函数, 设 $z = f(y_1, y_2, \dots, y_n)$, $y_i = g_i(x_1, x_2, \dots, x_m)$, $i = 1, 2, \dots, n$, 则有

$$\frac{\partial z}{\partial x_j} = \sum_{i=1}^n \frac{\partial z}{\partial y_i} \frac{\partial y_i}{\partial x_j} = \sum_{i=1}^n \frac{\partial y_i}{\partial x_j} \frac{\partial z}{\partial y_i}$$

即

$$\frac{\partial z}{\partial x_j} = \left(\frac{\partial z}{\partial y_1}, \frac{\partial z}{\partial y_2}, \dots, \frac{\partial z}{\partial y_n} \right) \begin{pmatrix} \frac{\partial y_1}{\partial x_j} \\ \frac{\partial y_2}{\partial x_j} \\ \vdots \\ \frac{\partial y_n}{\partial x_j} \end{pmatrix} = \left(\frac{\partial y_1}{\partial x_j}, \frac{\partial y_2}{\partial x_j}, \dots, \frac{\partial y_n}{\partial x_j} \right) \begin{pmatrix} \frac{\partial z}{\partial y_1} \\ \frac{\partial z}{\partial y_2} \\ \vdots \\ \frac{\partial z}{\partial y_n} \end{pmatrix}$$

例 17

考虑函数 $z = f(y_1, y_2) = e^{y_1 y_2^2}$, $y_1 = g_1(x) = x \cos x$, $y_2 = g_2(x) = x \sin x$. 那么

$$\begin{aligned}\frac{\partial z}{\partial x} &= \left(\frac{\partial y_1}{\partial x}, \frac{\partial y_2}{\partial x} \right) \begin{pmatrix} \frac{\partial z}{\partial y_1} \\ \frac{\partial z}{\partial y_2} \end{pmatrix} \\ &= (\cos x - x \sin x, \sin x + x \cos x) \begin{pmatrix} y_2^2 e^{y_1 y_2^2} \\ 2y_1 y_2 e^{y_1 y_2^2} \end{pmatrix} \\ &= (y_2^2 (\cos x - x \sin x) + 2y_1 y_2 (\sin x + x \cos x)) e^{y_1 y_2^2}\end{aligned}$$

当我们把 $\mathbf{y} = \mathbf{g}(x)$ 看做一个向量值函数时, 我们就可以将上述例子看做是求复合函数 $z = f(\mathbf{g}(x))$ 关于 x 的导数, 并且可以得到公式

$$\frac{\partial z}{\partial x} = \frac{\partial \mathbf{y}}{\partial x}^T \frac{\partial z}{\partial \mathbf{y}}$$

(多元) 链式法则

一般地，我们可以对多个向量值函数（或标量值函数）复合的函数求偏导，有以下定理：

定理 8

假设我们有 n 个列向量 $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}$ ，它们各自的长度为 l_1, l_2, \dots, l_n ，假设 $\mathbf{x}^{(i)}$ 是 $\mathbf{x}^{(i-1)}$ 的一个函数，则对于所有的 $i = 2, 3, \dots, n$ 有

$$\frac{\partial(\mathbf{x}^{(n)})^T}{\partial \mathbf{x}^{(1)}} = \frac{\partial(\mathbf{x}^{(2)})^T}{\partial \mathbf{x}^{(1)}} \frac{\partial(\mathbf{x}^{(3)})^T}{\partial \mathbf{x}^{(2)}} \cdots \frac{\partial(\mathbf{x}^{(n)})^T}{\partial \mathbf{x}^{(n-1)}}$$

此定理即（多元）链式法则。

证明.

根据向量值函数梯度的定义9和向量值函数微分定理6, 我们应用这个定理在每一对相关向量上, 则有

$$d\mathbf{x}^{(2)} = \left(\frac{\partial(\mathbf{x}^{(2)})^T}{\partial \mathbf{x}^{(1)}} \right)^T d\mathbf{x}^{(1)}, d\mathbf{x}^{(3)} = \left(\frac{\partial(\mathbf{x}^{(3)})^T}{\partial \mathbf{x}^{(2)}} \right)^T d\mathbf{x}^{(2)}, \dots, d\mathbf{x}^{(n)} = \left(\frac{\partial(\mathbf{x}^{(n)})^T}{\partial \mathbf{x}^{(n-1)}} \right)^T d\mathbf{x}^{(n-1)}$$

将它们合并起来则有

$$\begin{aligned} d\mathbf{x}^{(n)} &= \left(\frac{\partial(\mathbf{x}^{(n)})^T}{\partial \mathbf{x}^{(n-1)}} \right)^T \cdots \left(\frac{\partial(\mathbf{x}^{(3)})^T}{\partial \mathbf{x}^{(2)}} \right)^T \left(\frac{\partial(\mathbf{x}^{(2)})^T}{\partial \mathbf{x}^{(1)}} \right)^T d\mathbf{x}^{(1)} \\ &= \left(\frac{\partial(\mathbf{x}^{(2)})^T}{\partial \mathbf{x}^{(1)}} \frac{\partial(\mathbf{x}^{(3)})^T}{\partial \mathbf{x}^{(2)}} \cdots \frac{\partial(\mathbf{x}^{(n)})^T}{\partial \mathbf{x}^{(n-1)}} \right)^T d\mathbf{x}^{(1)} \end{aligned}$$

再次应用定理6可得

$$\frac{\partial(\mathbf{x}^{(n)})^T}{\partial \mathbf{x}^{(1)}} = \frac{\partial(\mathbf{x}^{(2)})^T}{\partial \mathbf{x}^{(1)}} \frac{\partial(\mathbf{x}^{(3)})^T}{\partial \mathbf{x}^{(2)}} \cdots \frac{\partial(\mathbf{x}^{(n)})^T}{\partial \mathbf{x}^{(n-1)}}$$



例 18

考虑线性回归中的优化问题:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^n (\boldsymbol{\theta}^T \mathbf{x}_i - y_i)^2$$

我们将其目标函数改写成 $\|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2$ 并关于 $\boldsymbol{\theta}$ 求梯度, 其中 $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)^T$, $\mathbf{y} = (y_1, y_2, \dots, y_n)^T$, 由链式法则我们有

$$\begin{aligned} & \nabla_{\boldsymbol{\theta}} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 \\ &= \frac{\partial (\mathbf{X}\boldsymbol{\theta} - \mathbf{y})^T}{\partial \boldsymbol{\theta}} \frac{\partial \|z\|_2^2}{\partial z}, \quad \text{其中 } z = \mathbf{X}^T \boldsymbol{\theta} - \mathbf{y} \\ &= \mathbf{X}^T \frac{\partial z^T z}{\partial z} \\ &= 2\mathbf{X}^T z \\ &= 2\mathbf{X}^T \mathbf{X}\boldsymbol{\theta} - \mathbf{X}^T \mathbf{y} \end{aligned}$$

例 19

计算 $(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$ 关于 $\boldsymbol{\mu}$ 的导数, 其中 $\boldsymbol{\Sigma}^{-1}$ 是对称矩阵。由链式法则, 我们有

$$\begin{aligned} & \frac{\partial [(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})]}{\partial \boldsymbol{\mu}} \\ &= \frac{\partial [(\mathbf{x} - \boldsymbol{\mu})^T]}{\partial \boldsymbol{\mu}} \frac{\partial [(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})]}{\partial [\mathbf{x} - \boldsymbol{\mu}]} \\ &= \frac{\partial [(\mathbf{x} - \boldsymbol{\mu})^T]}{\partial \boldsymbol{\mu}} 2\boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \\ &= -I 2\boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \\ &= -2\boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \end{aligned}$$

18.4.2 一些有用的公式

下面给出一些机器学习中经常使用的梯度公式：

•

$$\frac{\partial}{\partial \mathbf{X}} \text{Tr}(f(\mathbf{X})) = \text{Tr}\left(\frac{\partial f(\mathbf{X})}{\partial \mathbf{X}}\right)^T$$

•

$$\frac{\partial}{\partial \mathbf{X}} \det(f(\mathbf{X})) = \det(f(\mathbf{X})) \text{Tr}(f^{-1}(\mathbf{X}) \frac{\partial f(\mathbf{X})}{\partial \mathbf{X}})$$

•

$$\frac{\partial}{\partial \mathbf{X}} f^{-1}(\mathbf{X}) = -f^{-1}(\mathbf{X}) \frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} f^{-1}(\mathbf{X})$$

•

$$\frac{\partial \mathbf{a}^T \mathbf{X}^{-1} \mathbf{b}}{\partial \mathbf{X}} = -\mathbf{X}^{-T} \mathbf{a} \mathbf{b}^T \mathbf{X}^{-T}, \mathbf{X}^{-T} = \mathbf{X}^{-1^T}$$

- $$\frac{\partial \mathbf{a}^T \mathbf{x}}{\partial \mathbf{x}} = \frac{\partial \mathbf{x}^T \mathbf{a}}{\partial \mathbf{x}} = \mathbf{a}$$

- $$\frac{\partial \mathbf{a}^T \mathbf{X} \mathbf{b}}{\partial \mathbf{X}} = \mathbf{a} \mathbf{b}^T$$

- $$\frac{\partial \exp(\mathbf{a}^T \mathbf{X} \mathbf{b})}{\partial \mathbf{X}} = \mathbf{a} \mathbf{b}^T \exp(\mathbf{a}^T \mathbf{X} \mathbf{b})$$

- $$\frac{\partial \mathbf{x}^T \mathbf{A} \mathbf{x}}{\partial \mathbf{x}} = \mathbf{A} \mathbf{x} + \mathbf{A}^T \mathbf{x} = (\mathbf{A} + \mathbf{A}^T) \mathbf{x}$$

特别地, 若 \mathbf{A} 为对称矩阵, 则有 $\frac{\partial \mathbf{x}^T \mathbf{A} \mathbf{x}}{\partial \mathbf{x}} = 2\mathbf{A} \mathbf{x}$ 。

- 1 18.1 向量和矩阵函数的梯度
- 2 18.2 向量和矩阵函数微分与迹微分法
- 3 18.3 向量值和矩阵值函数的梯度
- 4 18.4 链式法则与一些有用的梯度公式
- 5 18.5 反向传播与自动微分**
- 6 18.6 高阶导数与泰勒展开

18.5.1 神经网络中的梯度（计算图）：反向传播

- 在许多机器学习应用中，通过执行梯度下降来找到好的模型参数，这取决于我们可以根据模型参数计算学习目标的梯度。
- 对于给定的目标函数，可以使用微分和链式法则获得模型参数的梯度。

例 20

考虑函数

$$f(x) = \sqrt{x^2 + \exp(x^2)} + \cos(x^2 + \exp(x^2)), \quad (4)$$

应用链式法则，注意微分是线性的，我们可以计算其梯度为：

$$\begin{aligned} \frac{df}{dx} &= \frac{2x + 2x \exp(x^2)}{2\sqrt{x^2 + \exp(x^2)}} - \sin(x^2 + \exp(x^2))(2x + 2x \exp(x^2)) \\ &= 2x \left(\frac{1}{2\sqrt{x^2 + \exp(x^2)}} - \sin(x^2 + \exp(x^2))(1 + \exp(x^2)) \right). \end{aligned}$$

- 用这种显式的方式写出梯度通常是不切实际的，因为它常常导致导数的表达式非常冗长。在实践中，这意味着，梯度的实现可能比计算函数要昂贵得多，这是不必要的开销。
- 对于深层神经网络模型的训练，反向传播算法是计算与模型参数相关的误差函数的梯度的有效方法。

- 考虑深度神经网络中的函数：

$$\mathbf{y} = (f_K \circ f_{K-1} \circ \cdots \circ f_1)(\mathbf{x}) = f_K(f_{K-1}(\cdots (f_1(\mathbf{x})) \cdots))$$

其中 \mathbf{x} 是输入（如图像）， \mathbf{y} 是观察值（如类别标签），每个函数 $f_i, i = 1, \dots, K$ 有自己的参数。在多层神经网络中，我们有第 i 层的函数 $f_i(x_{i-1}) = \sigma(\mathbf{A}_i \mathbf{x}_{i-1} + \mathbf{b}_i)$ ，其中 x_{i-1} 是 $i-1$ 层的输出， σ 是一个激活函数，如 logistic sigmoid $\frac{1}{1+e^{-x}}$ ，tanh 或修正线性单元（ReLU）。

- 为了训练这些模型，我们需要损失函数 L 关于所有模型参数 $\mathbf{A}_j, \mathbf{b}_j, j = 1, \dots, K$ 的梯度，这就要求我们计算 L 关于每层输入的梯度。
- 因为 L 和 \mathbf{y} 是复合函数，所以我们需要使用链式法则。在机器学习中，链式法则在优化层次模型的参数（例如，深度神经网络，最大似然估计）时有重要作用。

例 21

假设有输入 x 和观测 y 以及如下定义的网络结构（如图1所示）：

$$f_0 := x \quad (5)$$

$$f_i := \delta_i(A_i f_{i-1} + b_i), i = 1, \dots, K. \quad (6)$$

我们希望找到 A_j, b_j ，使得平方损失函数

$$L(\theta) = \|y - f_K(\theta, x)\|^2 \quad (7)$$

最小化，其中 $\theta = \{A_1, b_1, \dots, A_K, b_K\}$ 。

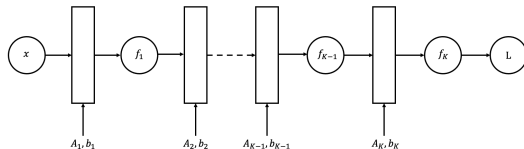


图 1: 多层神经网络中的正向传递，用于计算作为输入 x 和参数 A_i, b_i 的损失函数 L 。

为了获得相对于参数集 θ 的梯度，我们需要 L 关于每层参数 $\theta_j = \{\mathbf{A}_j, \mathbf{b}_j\}$ ($j = 1, 2, \dots, K$) 的偏导数。使用链式法则，有

$$\frac{\partial L^T}{\partial \theta_K} = \frac{\partial f_K^T}{\partial \theta_K} \frac{\partial L^T}{\partial f_K} \quad (8)$$

$$\frac{\partial L^T}{\partial \theta_{K-1}} = \frac{\partial f_{K-1}^T}{\partial \theta_{K-1}} \frac{\partial f_K^T}{\partial f_{K-1}} \frac{\partial L^T}{\partial f_K} \quad (9)$$

$$\frac{\partial L^T}{\partial \theta_{K-2}} = \frac{\partial f_{K-2}^T}{\partial \theta_{K-2}} \frac{\partial f_{K-1}^T}{\partial f_{K-2}} \frac{\partial f_K^T}{\partial f_{K-1}} \frac{\partial L^T}{\partial f_K} \quad (10)$$

$$\vdots$$

$$\frac{\partial L^T}{\partial \theta_i} = \frac{\partial f_i^T}{\partial \theta_i} \frac{\partial f_{i+1}^T}{\partial f_i} \frac{\partial f_{i+2}^T}{\partial f_{i+1}} \dots \frac{\partial f_K^T}{\partial f_{K-1}} \frac{\partial L^T}{\partial f_K} \quad (11)$$

假设我们已经准备好计算偏导数 $\frac{\partial L}{\partial f_{i+1}}$ ，那么大部分计算可以重复使用来计算 $\frac{\partial L}{\partial f_i}$ 。图2显示了梯度通过网络反向传播。

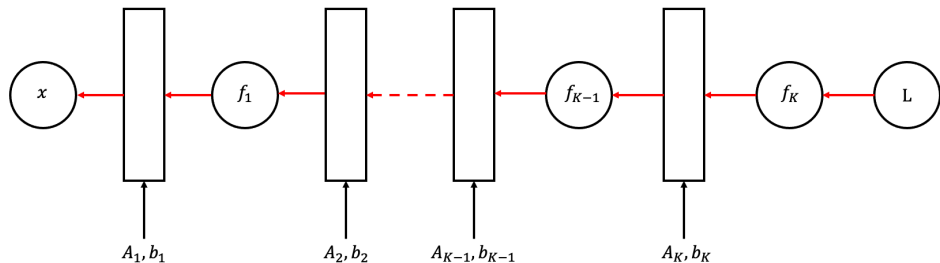


图 2: 三阶张量的 3-模式向量积的原理图

链式法则的计算复杂性分析

注记

1. 在神经网络的每一层上都有许多节点 (神经元), 所以神经网络模型函数是多变量函数, 因此上述使用的链式法则是多元链式法则。
2. 使用链式法则以及反向传递, 除了大部分计算可以重复使用外, 事实上这里还有一个问题就是: 链式法则求梯度公式中都是一些 **Jacobian** 矩阵或梯度向量的乘积, 因此这些矩阵之间、矩阵和向量乘法的哪个顺序 (沿着链向前或向后) 更快?

矩阵矩阵向量式乘积的计算复杂性

假设链式法则中有三个因子乘积 $\mathbf{M}_1 \mathbf{M}_2 \mathbf{w}$ ，两个矩阵和一个向量。我们要先做矩阵乘积 $\mathbf{M}_1 \mathbf{M}_2$ 还是先做矩阵向量乘积 $\mathbf{M}_2 \mathbf{w}$ ？

对于 $N \times N$ 矩阵， $\mathbf{M}_1 \mathbf{M}_2$ 包含 N^3 个独立的乘法，而 $\mathbf{M}_2 \mathbf{w}$ 有 N^2 个独立的乘法。

因此 $(\mathbf{M}_1 \mathbf{M}_2) \mathbf{w}$ 需要 $N^3 + N^2$ 乘法，而 $\mathbf{M}_1 (\mathbf{M}_2 \mathbf{w})$ 仅需要 $N^2 + N^2$ 。

这是一个重要的区别。如果我们在神经网络中有来自 L 个层的 L 个矩阵链，则差异本质上是 N 的一个因子：

- 正向 $((\mathbf{M}_1 \mathbf{M}_2) \mathbf{M}_3) \dots \mathbf{M}_L \mathbf{w}$ 需要 $(L-1)N^3 + N^2$ 个乘法
- 反向 $\mathbf{M}_1 (\mathbf{M}_2 (\dots (\mathbf{M}_L \mathbf{w})))$ 需要 LN^2 个乘法

矩阵矩阵矩阵式乘积的计算复杂性：以什么顺序计算矩阵 ABC 的乘积

正向和反向顺序之间的决定也出现在矩阵乘法中。如果要求我们将 A 乘以 B 乘以 C , 则结合律为乘法顺序提供了两种选择：

- 首先计算 AB 还是 BC ?
- 计算 $(AB)C$ 还是 $A(BC)$?

他们的结果相同, 但单个乘法的数量可能非常不同。假设矩阵 A 为 $m \times n$, B 为 $n \times p$ 以及 C 为 $p \times q$ 。

- 第一种方式 $AB = (m \times n)(n \times p)$ 具有 mnp 个乘法
 $(AB)C = (m \times p)(p \times q)$ 具有 mpq 个乘法
- 第二种方式 $BC = (n \times p)(p \times q)$ 具有 npq 个乘法
 $A(BC) = (m \times n)(n \times q)$ 有 mnq 个乘法

因此我们比较 $mp(n + q)$ 和 $nq(m + p)$, 将两个数除以 $mnpq$ 就会有结论：

当 $\frac{1}{q} + \frac{1}{n} < \frac{1}{m} + \frac{1}{p}$ 时, 则第一种方式更快; 反之, 第二种方式更快。

深度神经网络计算梯度时反向传播的合理性

在深度神经网络中，我们会定义类似如下网络：

$$\mathbf{f}(\mathbf{v}) = \mathbf{A}_L \mathbf{v}_{L-1} = \mathbf{A}_L (\mathbf{R} \mathbf{A}_{L-1} (\dots (\mathbf{R} \mathbf{A}_2 (\mathbf{R} \mathbf{A}_1 \mathbf{v}))))$$

我们的目的是优化其中的参数，所以当我们决定了损失函数 $L(\mathbf{f})$ ，我们所要求的就是 L 关于各参数的梯度，即

$$\frac{\partial L}{\partial \mathbf{A}_i} = \frac{\partial \mathbf{v}_i^T}{\partial \mathbf{A}_i} \frac{\partial \mathbf{v}_{i+1}^T}{\partial \mathbf{v}_i} \cdots \frac{\partial \mathbf{v}_{L-1}^T}{\partial \mathbf{v}_{L-2}} \frac{\partial \mathbf{f}^T}{\partial \mathbf{v}_{L-1}} \frac{\partial L}{\partial \mathbf{f}}$$

等式的右边恰好为若干个矩阵相乘，并且最后乘以了一个向量。根据前面的结论，我们可以知道按照反向计算可以大大减少计算梯度时的计算量。

注： $\frac{\partial \mathbf{v}_i^T}{\partial \mathbf{A}_i}$ 即为 $\frac{\partial \mathbf{v}_i^T}{\partial \text{vec}(\mathbf{A}_i)}$ 。

18.5.2 自动微分

反向传播是数值分析中自动微分的特殊情形。反向传播通过使用中间变量和应用链式法则来计算函数的梯度。自动微分应用一系列基本算术运算，例如加法和乘法以及基本函数，例如 \sin , \cos , \exp , \log 。通过将链式法则应用于这些操作，可以自动计算相当复杂的函数的梯度。

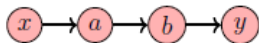


图 3: 一个简单的计算图，显示了数据从 x ，经过中间变量，最终到 y

图3显示了一个简单的计算图，在该计算图中，输入数据 x 经过中间变量 a b 得到输出 y 。如果我们想要去计算梯度 $\frac{dy}{dx}$ ，我们将应用链式法则，最终得到：

$$\frac{dy}{dx} = \frac{dy}{db} \frac{db}{da} \frac{da}{dx} \quad (12)$$

直观地，正向和反向模式在乘法的顺序上是不同，由于矩阵乘法的相关性，我们可以选择如下两种乘法顺序：

$$\frac{dy}{dx} = \left(\frac{dy}{db} \frac{db}{da} \right) \frac{da}{dx} \quad (13)$$

$$\frac{dy}{dx} = \frac{dy}{db} \left(\frac{db}{da} \frac{da}{dx} \right) \quad (14)$$

等式(13)是反向模式，因为梯度通过图向后传播，即与数据流相反。公式(14)是正向模式，其中梯度随着数据从左到右流过图。

在下文中，我们将重点关注反向模式自动微分，即反向传播。在神经网络的背景下，输入维度通常远高于标签维数，反向模式在计算上比正向模式容易得多。

例 22

考虑函数

$$f(x) = \sqrt{x^2 + \exp(x^2)} + \cos(x^2 + \exp(x^2)) \quad (15)$$

如果我们要在计算机上实现一个函数 f ，我们可以通过使用中间变量来节省一些计算：

$$a = x^2 \quad (16)$$

$$b = \exp(a) \quad (17)$$

$$c = a + b \quad (18)$$

$$d = \sqrt{c} \quad (19)$$

$$e = \cos(c) \quad (20)$$

$$f = d + e \quad (21)$$

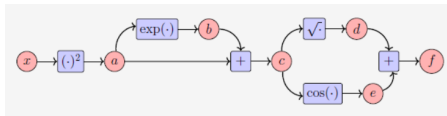


图 4: 输入 x , 函数 f 以及中间变量为 a, b, c, d, e 的计算图.

这与应用链式法则时所发生的思考过程是一样的。注意，上述方程组所需的操作比直接实现例子中定义的函数 $f(x)$ 所需的操作要少。图4中相应的计算图显示了获取函数值 f 所需的数据流和计算。

包含中间变量的方程组可以看作是一个计算图，一种广泛应用于神经网络软件库实现的表示形式。通过回顾初等函数导数的定义，我们可以直接计算中间变量对其相应输入的导数。我们获得：

$$\frac{\partial a}{\partial x} = 2x \quad (22)$$

$$\frac{\partial b}{\partial a} = \exp(a) \quad (23)$$

$$\frac{\partial c}{\partial a} = 1 = \frac{\partial c}{\partial b} \quad (24)$$

$$\frac{\partial d}{\partial c} = \frac{1}{2\sqrt{c}} \quad (25)$$

$$\frac{\partial e}{\partial c} = -\sin(c) \quad (26)$$

$$\frac{\partial f}{\partial d} = 1 = \frac{\partial f}{\partial e} \quad (27)$$

通过看图4中的计算图，我们通过输出的反向传播计算出 $\frac{\partial f}{\partial x}$ ，并且我们可以得到下面的关系：

$$\frac{\partial f}{\partial c} = \frac{\partial f}{\partial d} \frac{\partial d}{\partial c} + \frac{\partial f}{\partial e} \frac{\partial e}{\partial c} \quad (28)$$

$$\frac{\partial f}{\partial b} = \frac{\partial f}{\partial c} \frac{\partial c}{\partial b} \quad (29)$$

$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial b} \frac{\partial b}{\partial a} + \frac{\partial f}{\partial c} \frac{\partial c}{\partial a} \quad (30)$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial a} \frac{\partial a}{\partial x} \quad (31)$$

注意，我们隐含地应用了链式法则来获得 $\frac{\partial f}{\partial x}$ ，通过替换初等函数的导数，我们得到

$$\frac{\partial f}{\partial c} = 1 \cdot \frac{1}{2\sqrt{c}} + 1 \cdot (-\sin(c)) \quad (32)$$

$$\frac{\partial f}{\partial b} = \frac{\partial f}{\partial c} \cdot 1 \quad (33)$$

$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial b} \exp(a) + \frac{\partial f}{\partial c} \cdot 1 \quad (34)$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial a} \cdot 2x \quad (35)$$

通过把上面的每一个导数看作一个变量，我们观察到计算导数所需要的计算与函数本身的计算具有相似的复杂性。这是非常违反直觉的，因为例子中函数导数的数学表达式要比函数 $f(x)$ 本身的数学表达式复杂得多。

例 23

我们考虑一个两层的全连接神经网络：

$$\mathbf{y} = f(\mathbf{x}) = \text{ReLU}(\mathbf{A}_2(\text{ReLU}(\mathbf{A}_1\mathbf{x} + \mathbf{b}_1)) + \mathbf{b}_2)$$

其中

$$\mathbf{A}_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -2 & 1 \end{pmatrix}, \mathbf{A}_2 = \begin{pmatrix} 1 & -2 & 1 \\ 2 & -1 & 0 \end{pmatrix}, \mathbf{b}_1 = \begin{pmatrix} 1 \\ -2 \\ -1 \end{pmatrix}, \mathbf{b}_2 = \begin{pmatrix} -2 \\ -3 \end{pmatrix}$$

假设输入为 $\mathbf{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ ，并且对应的真实输出为 $\mathbf{y}' = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ ，采用平方损失 $L = \frac{1}{2}\|\mathbf{y} - \mathbf{y}'\|_2^2$ 。

试计算函数 L 关于 $\mathbf{b}_1, \mathbf{b}_2$ 的梯度。

解

先计算前项过程：

$$\mathbf{A}_1 \mathbf{x} + \mathbf{b}_1 = \begin{pmatrix} 2 \\ -2 \\ -2 \end{pmatrix}, \mathbf{A}_2(\text{ReLU}(\mathbf{A}_1 \mathbf{x} + \mathbf{b}_1)) + \mathbf{b}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

故 $\mathbf{y} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ 从而 $L = 1$ 。记

$$\mathbf{k} = \text{ReLU}(\mathbf{A}_1 \mathbf{x} + \mathbf{b}_1)$$

然后分别计算

$$\frac{\partial L}{\partial \mathbf{y}} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \frac{\partial \mathbf{y}^T}{\partial \mathbf{b}_2} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \frac{\partial \mathbf{y}^T}{\partial \mathbf{k}} = \begin{pmatrix} 1 & 2 \\ -2 & -1 \\ 1 & 0 \end{pmatrix}, \frac{\partial \mathbf{k}^T}{\partial \mathbf{b}_1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

所以有

$$\frac{\partial L}{\partial \mathbf{b}_1} = \frac{\partial \mathbf{k}^T}{\partial \mathbf{b}_1} \frac{\partial \mathbf{y}^T}{\partial \mathbf{k}} \frac{\partial L}{\partial \mathbf{y}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$
$$\frac{\partial L}{\partial \mathbf{b}_2} = \frac{\partial \mathbf{y}^T}{\partial \mathbf{b}_2} \frac{\partial L}{\partial \mathbf{y}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

- 1 18.1 向量和矩阵函数的梯度
- 2 18.2 向量和矩阵函数微分与迹微分法
- 3 18.3 向量值和矩阵值函数的梯度
- 4 18.4 链式法则与一些有用的梯度公式
- 5 18.5 反向传播与自动微分
- 6 18.6 高阶导数与泰勒展开**

18.6.1 Hessian 矩阵

我们前面已经讨论过了梯度，即一阶导数。有时我们会对高阶导数感兴趣，比如在优化中使用牛顿法时我们需要二阶导数。在一元的情况下，我们可以使用泰勒展开构造多项式来逼近函数，在多元情况下，我们同样可以这么做。

定义 13

设函数 $y = f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ 。 $f(\mathbf{x})$ 的 **Hessian** 矩阵被定义为

$$\mathbf{H} = \frac{\partial}{\partial \mathbf{x}^T} \frac{\partial f}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

记作 $\nabla^2 f$ 。

求函数的 Hessian 矩阵可以用二步法求出：

- (1) 求实值函数 $f(\mathbf{x})$ 关于向量变元 \mathbf{x} 的偏导数，得到实值函数的梯度 $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$ 。
 - (2) 再求梯度 $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$ 相对于 $1 \times n$ 行向量 \mathbf{x}^T 的偏导数，得到梯度的梯度即 Hessian 矩阵。
- 根据以上步骤，容易得到 Hessian 矩阵的下列公式。

- 对于 $n \times 1$ 常数向量 \mathbf{a} ，有

$$\frac{\partial^2 \mathbf{a}^T \mathbf{x}}{\partial \mathbf{x} \partial \mathbf{x}^T} = \mathbf{O}_{n \times n} \quad (36)$$

- 若 \mathbf{A} 是 $n \times n$ 矩阵，则

$$\frac{\partial^2 \mathbf{x}^T \mathbf{A} \mathbf{x}}{\partial \mathbf{x} \partial \mathbf{x}^T} = \mathbf{A} + \mathbf{A}^T \quad (37)$$

- 令 \mathbf{x} 为 $n \times 1$ 向量, \mathbf{a} 为 $m \times 1$ 常数向量, \mathbf{A} 和 \mathbf{B} 分别为 $m \times n$ 和 $m \times m$ 常数矩阵, 且 \mathbf{B} 为对称矩阵, 则

$$\frac{\partial^2 (\mathbf{a} - \mathbf{A}\mathbf{x})^T \mathbf{B} (\mathbf{a} - \mathbf{A}\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T} = 2\mathbf{A}^T \mathbf{B} \mathbf{A} \quad (38)$$

- 若 \mathbf{A} 是一个与向量 \mathbf{x} 无关的矩阵, 而 $\mathbf{y}(\mathbf{x})$ 是与向量 \mathbf{x} 的元素有关的列向量, 则

$$\begin{aligned} \frac{\partial^2 [\mathbf{y}(\mathbf{x})]^T \mathbf{A} \mathbf{y}(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T} &= \frac{\partial [\mathbf{y}(\mathbf{x})]^T}{\partial \mathbf{x}} (\mathbf{A} + \mathbf{A}^T) \frac{\partial \mathbf{y}(\mathbf{x})}{\partial \mathbf{x}^T} + \\ &([\mathbf{y}(\mathbf{x})]^T (\mathbf{A} + \mathbf{A}^T) \otimes \mathbf{I}_n) \frac{\partial}{\partial \mathbf{x}^T} \left[\text{vec} \left(\frac{\partial [\mathbf{y}(\mathbf{x})]^T}{\partial \mathbf{x}} \right) \right] \end{aligned} \quad (39)$$

Hessian 矩阵在机器学习优化中有很多应用。如果 $f(\mathbf{x})$ 是二次 (连续) 可微的函数, 则二阶偏导可交换, 也即二阶偏导与微分的顺序无关, 此时 Hessian 矩阵是对称矩阵。在凸优化的章节中, 我们将会学到在函数的极小点处 Hessian 矩阵为正定矩阵。Hessian 矩阵也被应用于二阶优化算法如牛顿法能够快速的收敛到最优点。

18.6.2 线性化和多元泰勒级数

一个函数的梯度 ∇f 通常可以作为 \mathbf{x}_0 附近的局部线性逼近

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + (\nabla \mathbf{x} f)^T(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)$$

这里 $(\nabla \mathbf{x} f)^T(\mathbf{x}_0)$ 是 f 关于 \mathbf{x} 的梯度在 \mathbf{x}_0 处的取值。即通过一条直线来逼近函数 f , 这种逼近是局部准确的, 但是在更大范围内是有很大的误差的。上式实际上是函数 f 在 \mathbf{x}_0 处泰勒展开的前两项, 它是 $f(\mathbf{x})$ 在 \mathbf{x}_0 处的高阶多元泰勒级数展开的特殊情形。

泰勒展开

定义 14

(多元泰勒展开) 对于多元泰勒展开, 我们考虑函数 $f: \mathbb{R}^D \rightarrow \mathbb{R}, \mathbf{x} \rightarrow f(\mathbf{x}), \mathbf{x} \in \mathbb{R}^D$ 在 \mathbf{x}_0 处光滑。如果我们定义差分向量 $\boldsymbol{\delta} := \mathbf{x} - \mathbf{x}_0$, 那么 f 在 \mathbf{x}_0 处的泰勒展开为

$$f(\mathbf{x}) = \sum_{k=0}^{\infty} \frac{D_{\mathbf{x}}^k f(\mathbf{x}_0)}{k!} \boldsymbol{\delta}^k$$

其中, $D_{\mathbf{x}}^k f(\mathbf{x}_0)$ 是 f 关于 \mathbf{x} 的 k 阶全微分在 \mathbf{x}_0 处的取值。

定义 15

函数 f 在 \mathbf{x}_0 处的 n 阶泰勒多项式被定义为泰勒展开的前 $n+1$ 项:

$$T_n = \sum_{k=0}^n \frac{D_{\mathbf{x}}^k f(\mathbf{x}_0)}{k!} \boldsymbol{\delta}^k$$

注意

当 $D > 1, k > 1$ 时, 我们在上面使用的简写记号 δ^k 并没有在 \mathbb{R}^D 中定义。这里的 $D_x^k f, \delta^k$ 都是 k 阶张量, $\delta^k \in \mathbb{R}^{D \times D \times \dots \times D}$ 是通过张量积 (用符号 \otimes) 得到的。例如

$$\delta^2 = \delta \otimes \delta = \delta \delta^T, \delta^2[i, j] = \delta[i] \delta[j]$$

$$\delta^3 = \delta \otimes \delta \otimes \delta, \delta^3[i, j, k] = \delta[i] \delta[j] \delta[k]$$

在泰勒展开中, 我们得到以下式子

$$D_x^k f(\mathbf{x}_0) \delta^k = \sum_a \cdots \sum_k D_x^k f(\mathbf{x}_0)[a, \dots, k] \delta[a] \cdots \delta[k]$$

其中

$$D_x^k f(\mathbf{x})[i_1, \dots, i_k] = \frac{\partial^k}{\partial x_{i_1} \cdots \partial x_{i_k}} f(\mathbf{x})$$

所以 $D_x^k f(\mathbf{x}_0) \delta^k$ 包含了所有 k 次多项式。

$$k = 0 : D_x^0 f(\mathbf{x}_0) \delta^0 = f(\mathbf{x}_0) \in \mathbb{R}$$

$$k = 1 : D_x^1 f(\mathbf{x}_0) \delta^1 = \nabla_x f(\mathbf{x}_0) \delta = \sum_i \nabla f(\mathbf{x}_0)[i] \delta[i] \mathbb{R}$$

$$k = 2 : D_x^2 f(\mathbf{x}_0) \delta^2 = \delta^T \mathbf{H} \delta = \sum_i \sum_j H[i, j] \delta[i] \delta[j] \in \mathbb{R}$$

$$k = 3 : D_x^3 f(\mathbf{x}_0) \delta^3 = \sum_i \sum_j \sum_k D_x^3 f(\mathbf{x}_0)[i, j, k] \delta[i] \delta[j] \delta[k] \in \mathbb{R}$$

例 24

求函数 $f(\mathbf{x}) = \mathbf{a}^T e^{\mathbf{x}}$ 在 $\mathbf{0}$ 处的 2 阶泰勒多项式。

解

根据泰勒展开我们有

$$T_2 = f(\mathbf{0}) + (\nabla_x f(\mathbf{0}))^T (\mathbf{x} - \mathbf{0}) + \frac{1}{2} (\mathbf{x} - \mathbf{0})^T (\nabla_x^2 f(\mathbf{0})) (\mathbf{x} - \mathbf{0})$$

通过计算可得

$$\nabla_x f(\mathbf{x}) = (\mathbf{a}_1 e^{x_1}, \mathbf{a}_2 e^{x_2}, \dots, \mathbf{a}_n e^{x_n})^T$$

$$\nabla_x^2 f(\mathbf{x}) = \text{diag}(\nabla_x f(\mathbf{x}))$$

所以 $f(\mathbf{0}) = \sum_{i=1}^n \mathbf{a}_i, \nabla_x f(\mathbf{0}) = \mathbf{a}, \nabla_x^2 f(\mathbf{0}) = \text{diag}(\mathbf{a})$

故 $T_2 = \sum_{i=1}^n \mathbf{a}_i + \mathbf{a}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \text{diag}(\mathbf{a}) \mathbf{x} = \sum_{i=1}^n \mathbf{a}_i (1 + x_i + \frac{1}{2} x_i^2)$

本讲小结

一阶导数

- 向量和矩阵函数的梯度、微分
- 向量和矩阵值函数的梯度：
Jacobian 矩阵
- 链式法则
- 反向传播与自动微分

高阶导数

- 二阶偏导数
- Hessian 矩阵
- 多元泰勒级数
- 泰勒多项式

向量和矩阵微分在优化问题求解方法，如梯度下降法、牛顿法中有重要应用。