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 向量和矩阵函数的微分
3. 18.3 向量值函数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 反向传播与自动微分

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一个优化问题：

例 1

- 逻辑回归：

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

- 线性可分支持向量机模型：

$$\min_{\boldsymbol{w}} \frac{1}{2} \|\boldsymbol{w}\|^2$$

$$s.t.\ y_i(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b) - 1 > 0$$

- *PCA*：

$$\min_{\boldsymbol{W}} -Tr(\boldsymbol{W}^T \boldsymbol{X} \boldsymbol{X}^T \boldsymbol{W})$$

$$s.t. \boldsymbol{W}^T \boldsymbol{W} = \boldsymbol{I}$$

例 2

在强化学习中我们需要对一个值函数进行近似，这时我们需要

$$\min_{\boldsymbol{\theta}} E_{\boldsymbol{x}\sim\pi}[\| V^{\pi}(\boldsymbol{x}) - V_{\boldsymbol{\theta}}(\boldsymbol{x})\|_2^2]$$

其中

$$V_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^T\boldsymbol{x}$$

例 3

在深度学习中我们可能会构造一个两层的神经网络

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

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

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

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

- 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 $f:\mathbb{R}\to\mathbb{R}$ 关于 x 的**导数**定义为

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

定义 2

函数 $f:\mathbb{R}\to\mathbb{R}$ 在 x_0 的 n 阶**泰勒多项式**为

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

多元函数偏导数

定义 3

光滑函数 $f: \mathbb{R} \to \mathbb{R}, f \in \mathbb{C}^{\infty}$ 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 $f(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ 关于 $\boldsymbol{x}$ 的 n 个分量的**偏导**为

$$\frac{\partial f}{\partial x_1} = \lim_{h \to 0} \frac{f(x_1 + h, x_2, \ldots, x_n) - f(\boldsymbol{x})}{h}$$

$$\vdots$$

$$\frac{\partial f}{\partial x_n} = \lim_{h \to 0} \frac{f(x_1, x_2, \ldots, x_n + h) - f(\boldsymbol{x})}{h}$$

向量函数梯度

定义 5

若 $\boldsymbol{x} \in \mathbb{R}^n, f(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ 是一实值函数，其中 $\boldsymbol{x} = \left(x_1, x_2, \ldots, x_n\right)^{\mathrm{T}}$，则定义

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

例 4

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

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

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

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

因此梯度为

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

例 5

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

将 $f(\boldsymbol{x})$ 写成分量的形式：

$$f(\boldsymbol{x})=\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}+\boldsymbol{b}=\sum_{i=1}^{n}a_ix_i+b_i$$

那么 $f(\boldsymbol{x})$ 对第 h 个分量的偏导数为

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

从而就有

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

例 6

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

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

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

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

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 $f(\boldsymbol{x})$ 和 $g(\boldsymbol{x})$ 分别是向量 $\boldsymbol{x}$ 的实值函数，c_1 和 c_2 为实常数，则

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

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

$$\frac{\partial f(\boldsymbol{x}) g(\boldsymbol{x})}{\partial \boldsymbol{x}} = g(\boldsymbol{x}) \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} + f(\boldsymbol{x}) \frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}} \tag{2}$$

- 商法则：若 $g(\boldsymbol{x}) \neq 0$，则

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

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

则定义矩阵函数的梯度为

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

例 7

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

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

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

- 注意在向量函数梯度定义5中 $\boldsymbol{x}$ 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 $\boldsymbol{x}^{\mathrm{T}}$ 求偏导的结果

定理 1

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

$$\frac{\partial}{\partial \boldsymbol{x}^{\mathrm{T}}} f = (\frac{\partial}{\partial \boldsymbol{x}} f)^{\mathrm{T}}$$

证明.

通过定义6，有

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

□

例 8

在例5中我们考虑了一个非常简单的多元线性函数 $f(\boldsymbol{x}) = \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{b}$，我们知道

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

利用上述定理我们有

$$\frac{\partial f}{\partial \boldsymbol{x}^{\mathrm{T}}} = (\frac{\partial f}{\partial \boldsymbol{x}})^{\mathrm{T}} = \boldsymbol{a}^{\mathrm{T}}$$

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 $\frac{\partial f}{\partial \boldsymbol{x}}$ 作为桥梁。

例 9

对于一个可分的支持向量机，相应的优化问题为

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$s.t.\ y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) - 1 > 0$$

我们考虑其目标函数的梯度

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

我们逐分量地求其偏导数有

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

所以

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

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 $f(\boldsymbol{A})$ 和 $g(\boldsymbol{A})$ 分别是矩阵 $\boldsymbol{A}$ 的实值函数,c_1 和 c_2 为实常数, 则

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

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

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

- 商法则: 若 $g(\boldsymbol{A}) \neq 0$, 则

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

18.2.1 矩阵的微分

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 $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, 矩阵 $\boldsymbol{A}$ 的**微分**定义为

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

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

$$d\boldsymbol{x} = \begin{pmatrix} dx_1 \\ dx_2 \\ \vdots \\ dx_n \end{pmatrix}; d\boldsymbol{x}^{\mathrm{T}} = \begin{pmatrix} dx_1, dx_2, \ldots, dx_n \end{pmatrix}$$

下面我们将介绍矩阵微分的一些性质，并且我们将向量看做一种特殊的矩阵。

性质 1

矩阵微分有如下性质

- $d(c\boldsymbol{A}) = cd\boldsymbol{A}$ 其中 $\boldsymbol{A} \in \mathbb{R}^{n\times m}$
- $d(\boldsymbol{A}+\boldsymbol{B}) = d\boldsymbol{A} + d\boldsymbol{B}$ 其中 $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times m}$
- $d(\boldsymbol{A}\boldsymbol{B}) = d\boldsymbol{A}\boldsymbol{B} + \boldsymbol{A}d\boldsymbol{B}$ 其中 $\boldsymbol{A} \in \mathbb{R}^{n\times m}, \boldsymbol{B} \in \mathbb{R}^{m\times k}$
- $d\boldsymbol{A}^{\mathrm{T}} = (d\boldsymbol{A})^{\mathrm{T}}$ 其中 $\boldsymbol{A} \in \mathbb{R}^{n\times m}$

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 $\boldsymbol{A}, \boldsymbol{B}$ 的第 ij 个元素分别为 a_{ij}, b_{ij}。

$$
\begin{aligned}
\text{左边}_{ij} &= \mathrm{d}\left(\sum_k a_{ik}b_{kj}\right) \\
&= \sum_k (\mathrm{d}a_{ik}b_{kj} + a_{ik}\mathrm{d}b_{kj}) \\
\text{右边}_{ij} &= (\mathrm{d}\boldsymbol{A}\boldsymbol{B})_{ij} + (\boldsymbol{A}\mathrm{d}\boldsymbol{B})_{ij} \\
&= \sum_k \mathrm{d}a_{ik}b_{kj} + \sum_k a_{ik}\mathrm{d}b_{kj} \\
&= \text{左边}_{ij}
\end{aligned}
$$

□

定理 2

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

$$d\mathit{Tr}(\boldsymbol{A}) = \mathit{Tr}(d\boldsymbol{A})$$

证明.

$$\text{左边} = \mathrm{d}\left(\sum_i a_{ii}\right) = \sum_i \mathrm{d}a_{ii}$$

$$\text{右边} = \mathrm{Tr}\begin{bmatrix} \mathrm{d}a_{11} & \mathrm{d}a_{12} & \dots & \mathrm{d}a_{1n} \\ \mathrm{d}a_{21} & \mathrm{d}a_{22} & \dots & \mathrm{d}a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{d}a_{n1} & \mathrm{d}a_{n2} & \cdots & \mathrm{d}a_{nn} \end{bmatrix} = \sum_i \mathrm{d}a_{ii} = \text{左边}$$

□

18.2.2 矩阵微分与偏导数的联系

我们介绍矩阵微分其最终目标还是为了求解函数的梯度，在我们的定义下也就是偏导数。那么偏导数和梯度有什么联系呢？在接下来的内容中将涉及迹运算的一些性质。这些性质，我们已经在前面的内容中叙述过了，这里仅仅列出这些性质。

性质 2

迹函数性质

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

多元函数的微分和偏导的关系如下

$$\mathrm{d}f(x_1, x_2, \ldots, x_n) = \frac{\partial f}{\partial x_1}\mathrm{d}x_1 + \frac{\partial f}{\partial x_1}\mathrm{d}x_2 + \cdots + \frac{\partial f}{\partial x_1}\mathrm{d}x_n$$

这里 $\mathrm{d}f$ 是一个标量，从分量的角度来看，$\mathrm{d}f$ 就是将 $\frac{\partial f}{\partial \boldsymbol{A}}$ 与 $\mathrm{d}\boldsymbol{A}$ 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 $f: \mathbb{R}^{n\times m}$ 和 $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ 有

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

证明.

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

□

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 $c = \mathrm{Tr}(c)$，这也就意味着对于一个实值函数 $f(\boldsymbol{A})$ 有 $f(\boldsymbol{A}) = \mathrm{Tr}(f(\boldsymbol{A}))$ 从而就有 $\mathrm{d}f(\boldsymbol{A}) = \mathrm{dTr}(f(\boldsymbol{A})) = \mathrm{Tr}(\mathrm{d}f(\boldsymbol{A}))$。
- 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. $\mathrm{d}f = \mathrm{dTr}(f) = \mathrm{Tr}(\mathrm{d}f)$
2. 使用迹函数的性质和矩阵微分的性质来得到如下形式

$$\mathrm{d}f = \mathrm{Tr}\left(\boldsymbol{A}^T \mathrm{d}\boldsymbol{x}\right)$$

3. 应用定理3得到结果

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

例 10

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

$$
\begin{aligned}
df &= dTr\left(\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}\right)\\
&= Tr\left(d\left(\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}\right)\right)\\
&= Tr\left(d\left(\boldsymbol{x}^T\right)\boldsymbol{A}\boldsymbol{x}+\boldsymbol{x}^T d(\boldsymbol{A}\boldsymbol{x})\right)\\
&= Tr\left(d\left(\boldsymbol{x}^T\right)\boldsymbol{A}\boldsymbol{x}+\boldsymbol{x}^T d\boldsymbol{A}\boldsymbol{x}+\boldsymbol{x}^T\boldsymbol{A}d\boldsymbol{x}\right)\\
&= Tr\left(d\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}+\boldsymbol{x}^T\boldsymbol{A}d\boldsymbol{x}\right)\\
&= Tr\left(d\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}\right)+Tr\left(\boldsymbol{x}^T\boldsymbol{A}d\boldsymbol{x}\right)\\
&= Tr\left(\boldsymbol{x}^T\boldsymbol{A}^T d\boldsymbol{x}\right)+Tr\left(\boldsymbol{x}^T\boldsymbol{A}d\boldsymbol{x}\right)\\
&= Tr\left(\boldsymbol{x}^T\boldsymbol{A}^T d\boldsymbol{x}+\boldsymbol{x}^T\boldsymbol{A}d\boldsymbol{x}\right)\\
&= Tr\left(\left(\boldsymbol{x}^T\boldsymbol{A}^T+\boldsymbol{x}^T\boldsymbol{A}\right)d\boldsymbol{x}\right)
\end{aligned}
$$

我们可以得到

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

如果 $\boldsymbol{A}$ 是对称矩阵，我们还可以将其简化为

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

令 $\boldsymbol{A} = \boldsymbol{I}$ 我们则有

$$\frac{\partial \left(\boldsymbol{x}^T \boldsymbol{x}\right)}{\partial \boldsymbol{x}} = 2\boldsymbol{x}$$

例 11

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

$$\min_{\boldsymbol{x}} \boldsymbol{x}^{\mathrm{T}} \boldsymbol{L} \boldsymbol{x}$$
$$s.t. \boldsymbol{x}^{\mathrm{T}} \mathbf{1} = 0$$

中目标函数的梯度为

$$\nabla_{\boldsymbol{x}} \boldsymbol{x}^{\mathrm{T}} \boldsymbol{L} \boldsymbol{x} = 2\boldsymbol{L}\boldsymbol{x}$$

我们再看一个关于矩阵函数的例子。

例 12

在 PCA 中，我们需要求解优化问题

$$\min_{\boldsymbol{W}} -Tr(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{W})$$
$$s.t.\, \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}$$

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

解

利用迹微分法有

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

所以

$$\nabla_{\boldsymbol{W}} - Tr(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{W}) = -2\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}$$

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 13

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

解

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

故

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

18.2.5 逆矩阵

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

$$0=\mathrm{d}\boldsymbol{I}=\mathrm{d}\left(\boldsymbol{X}\boldsymbol{X}^{-1}\right)=\mathrm{d}\boldsymbol{X}\boldsymbol{X}^{-1}+\boldsymbol{X}\mathrm{d}\left(\boldsymbol{X}^{-1}\right)$$

整理可得

$$\mathrm{d}\left(\boldsymbol{X}^{-1}\right)=-\boldsymbol{X}^{-1}\mathrm{d}\boldsymbol{X}\boldsymbol{X}^{-1}$$

这样我们就得到了关于逆矩阵微分的一个结论。

例 14

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

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

解

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

所以

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

例 15

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

解

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

故

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

18.2.6 行列式

行列式也是关于矩阵的一个实值函数，所以我们可以去求 $\frac{\partial |\boldsymbol{A}|}{\partial \boldsymbol{A}}$。我们首先回顾一下行列式相关的一些概念，假设矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$，则:

- 余子式 M_{ij} 是矩阵 $\boldsymbol{A}$ 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 $|\boldsymbol{A}| = \sum_j a_{ij} A_{ij}$
- $\boldsymbol{A}$ 的伴随矩阵被定义为 $\boldsymbol{A}^*_{ij} = A_{ji}$
- 对于非奇异矩阵 $\boldsymbol{A}$ 有 $\boldsymbol{A}^{-1} = \frac{A^*}{|\boldsymbol{A}|}$

定理 4

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

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

证明.

为了计算 $\frac{\partial |\boldsymbol{A}|}{\partial \boldsymbol{A}}$，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

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

使用定义6来组织元素就有

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

□

如果矩阵 $\boldsymbol{A}$ 非奇异，则可以进一步推出

$$\frac{\partial|\boldsymbol{A}|}{\partial \boldsymbol{A}}=\left(|\boldsymbol{A}| \boldsymbol{A}^{-1}\right)^{T}=|\boldsymbol{A}|\left(\boldsymbol{A}^{-1}\right)^{T}$$

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 $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ 则有

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

当 $\boldsymbol{A}$ 可逆时有

$$d|\boldsymbol{A}|=Tr(|\boldsymbol{A}| \boldsymbol{A}^{-1} d \boldsymbol{A})$$

证明.

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

当 $\boldsymbol{A}$ 可逆时有

$$\mathrm{d}|\boldsymbol{A}| = \mathrm{Tr}(\boldsymbol{A}^{*}\mathrm{d}\boldsymbol{A}) = \mathrm{Tr}\left(|\boldsymbol{A}|\boldsymbol{A}^{-1}\mathrm{d}\boldsymbol{A}\right)$$

例 16

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

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

解

应用定理5有

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

故

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

18.3.2 向量值函数和矩阵值函数的梯度：Jacobian 矩阵

定义 9

设 $\boldsymbol{y}=\boldsymbol{f}(\boldsymbol{x})$ 其中 $f\colon \mathbb{R}^n \mapsto \mathbb{R}^m$，我们使用 ***Jacobian* 矩阵**来定义对 $\boldsymbol{y}$ 关于 $\boldsymbol{x}$ 的偏导。

$$\boldsymbol{J}=\frac{\partial \boldsymbol{y}^{\mathrm{T}}}{\partial \boldsymbol{x}}=\begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_2}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_1} \\ \frac{\partial y_1}{\partial x_2} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_m}{\partial x_2} \\ \vdots & \vdots & & \vdots \\ \frac{\partial y_1}{\partial x_n} & \frac{\partial y_2}{\partial x_n} & \cdots & \frac{\partial y_m}{\partial x_n} \end{pmatrix}$$

并且我们定义

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}^{\mathrm{T}}}=\boldsymbol{J}^{\mathrm{T}}=(\frac{\partial \boldsymbol{y}^{\mathrm{T}}}{\partial \boldsymbol{x}})^{\mathrm{T}}$$

注意，这里我们没有去定义 $\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}$ 以及 $\frac{\partial \boldsymbol{y}^{\mathrm{T}}}{\partial \boldsymbol{x}^{\mathrm{T}}}$，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。

Jacobian 矩阵是梯度在向量与向量之间的变化关系上的推广。我们试图将其进一步推广，研究矩阵与向量之间的变化关系或者矩阵之间的变化关系。但该部分内容较为繁琐，我们只介绍其主要思想。

定义 10

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

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

有了这样一函数之后，我们就可以定义两个矩阵之间的 Jacobian 矩阵。

定义 11

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

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

定义 12

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

18.3.3 向量函数微分

在先前的一些部分，我们将矩阵微分和矩阵导数联系在了一起。在使两个量相等时使用了迹作为桥梁。因此我们能够在两者之间选择更方便的一种形式来计算。
在此部分，我们给出了另一个联系向量函数微分和向量对向量导数的定理。这个定理有着一个惊人的简洁的形式。

定理 6

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

$$d\boldsymbol{f} = (\frac{\partial \boldsymbol{f}^{\mathrm{T}}}{\partial \boldsymbol{x}})^{\mathrm{T}} d\boldsymbol{x}$$

证明.

显然，$\mathrm{d}\boldsymbol{f}$ 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。

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

□

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我们可以仿照求解实值函数的步骤类似，我们可以考虑一个步骤来简化求解向量对向量导数。在此，我们利用一个例子来说明这样一个过程。

例 17

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

我们先计算 $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\eta}}$ 这是容易得到的

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

应用定理6我们有

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

因此

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

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

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

如果 $\boldsymbol{x}$ 和 $\boldsymbol{\eta}$ 的维数都是 d 那么我们定义 $\boldsymbol{\Sigma} = \boldsymbol{W}\boldsymbol{\Lambda}\,\boldsymbol{W}^{\mathrm{T}}$。我们就能得到一个优美的结果

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

这个结论我们可以应用到多元正态分布的推广中。

定理 7

当 $\boldsymbol{f}$ 和 $\boldsymbol{x}$ 是相同大小是由定义9

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

证明.

利用定理6

$$\mathrm{d}\boldsymbol{f} = \left(\frac{\partial \boldsymbol{f}^{\mathrm{T}}}{\partial \boldsymbol{x}}\right)^{T} \mathrm{d}\boldsymbol{x} \Longrightarrow \left(\left(\frac{\partial \boldsymbol{f}^{\mathrm{T}}}{\partial \boldsymbol{x}}\right)^{T}\right)^{-1} \mathrm{d}\boldsymbol{f} = \mathrm{d}\boldsymbol{x} \Longrightarrow \mathrm{d}\boldsymbol{x} = \left(\left(\frac{\partial \boldsymbol{f}^{\mathrm{T}}}{\partial \boldsymbol{x}}\right)^{-1}\right)^{T} \mathrm{d}\boldsymbol{f}$$

所以，我们就有

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

□

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。

18.3.4 Hessian 矩阵

定义 13

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

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

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 $f(\boldsymbol{x})$ 关于向量变元 $\boldsymbol{x}$ 的偏导数，得到实值函数的梯度 $\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}$。

(2) 再求梯度 $\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}$ 相对于 $1 \times n$ 行向量 $\boldsymbol{x}^{\mathrm{T}}$ 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 $\boldsymbol{a}$，有

$$\frac{\partial^2 \boldsymbol{a}^{\mathrm{T}} \boldsymbol{x}}{\partial \boldsymbol{x} \partial \boldsymbol{x}^{\mathrm{T}}} = \boldsymbol{O}_{n \times n} \tag{4}$$

- 若 $\boldsymbol{A}$ 是 $n \times n$ 矩阵，则

$$\frac{\partial^2 \boldsymbol{x}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{x}}{\partial \boldsymbol{x} \partial \boldsymbol{x}^{\mathrm{T}}} = \boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}} \tag{5}$$

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

$$\frac{\partial^2 (\boldsymbol{a}-\boldsymbol{A}\boldsymbol{x})^{\mathrm{T}} \boldsymbol{B}(\boldsymbol{a}-\boldsymbol{A}\boldsymbol{x})}{\partial \boldsymbol{x}\partial \boldsymbol{x}^{\mathrm{T}}} = 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{A} \tag{6}$$

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

$$\begin{aligned}\frac{\partial^2 [\boldsymbol{y}(\boldsymbol{x})]^{\mathrm{T}} \boldsymbol{A}\boldsymbol{y}(\boldsymbol{x})}{\partial \boldsymbol{x}\partial \boldsymbol{x}^{\mathrm{T}}} =& \frac{\partial [\boldsymbol{y}(\boldsymbol{x})]^{\mathrm{T}}}{\partial \boldsymbol{x}}(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}})\frac{\partial \boldsymbol{y}(\boldsymbol{x})}{\partial \boldsymbol{x}^{\mathrm{T}}} + \\ & ([\boldsymbol{y}(\boldsymbol{x})]^{\mathrm{T}}(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}})\otimes \boldsymbol{I}_n)\frac{\partial}{\partial \boldsymbol{x}^{\mathrm{T}}}\left[vec(\frac{\partial [\boldsymbol{y}(\boldsymbol{x})]^{\mathrm{T}}}{\partial \boldsymbol{x}})\right]\end{aligned} \tag{7}$$

Hessian 矩阵在机器学习优化中有很多应用由于微分运算在二阶偏导连续的点上可交换，因此 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。

18.4.1 链式法则

在上面的内容中我们讨论了实值函数关于矩阵或者向量的偏导以及矩阵微分。但是上面所举的例子都是之间能给出函数与所关心的矩阵或者向量之间的关系的。在一部分情况下，我们需要通过一些中间变量来传递的方式来展现函数与所关心的矩阵或者向量之间的关系。换句话说，现在我们关心如何对复合函数进行求偏导。

在一元情况下，设 $y = f(x), z = g(y)$ 则我们知道

$$\frac{\mathrm{d}z}{\mathrm{d}x} = \frac{\mathrm{d}z}{\mathrm{d}y}\frac{\mathrm{d}y}{\mathrm{d}x}$$

而在多元情况下 $z = f(y_1, y_2, \ldots, y_n), y_i = g_i(x_1, x_2, \ldots, x_m), i = 1, 2, \ldots, n$ 则

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

上面就是我们熟悉的链式法则，在本节我们并不打算讨论链式法则，链式法则将在之后讨论。

但是我们考虑这样一个依赖关系 $z = f(\boldsymbol{y}), \boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}), z \in \mathbb{R}, \boldsymbol{y}, \boldsymbol{x} \in \mathbb{R}^n$，如果要使用链式法则，则需要考虑对 $\boldsymbol{y}$ 关于 $\boldsymbol{x}$ 求偏导的具体含义。

现在，我们就来考虑对一个有着向量作为应变量的向量函数求偏导。

在很多偏导问题中，$\ln|\cdot|$ 函数是更为广泛地应用的：

$$\frac{\partial \ln|\boldsymbol{A}|}{\partial \boldsymbol{A}} = \frac{1}{|\boldsymbol{A}|}\frac{\partial|\boldsymbol{A}|}{\partial \boldsymbol{A}} = \left(\boldsymbol{A}^{-1}\right)^{T}$$

第一个等号来自于一般的微积分规则中的链式法则 ($\ln|\boldsymbol{A}|$ 和 $|\boldsymbol{A}|$ 都是标量)。同样的我们也能直接给出他的微分

$$\mathrm{d}\ln|\boldsymbol{A}| = \mathrm{Tr}\left[\boldsymbol{A}^{-1}\mathrm{d}\boldsymbol{A}\right]$$

上面我们总结的步骤中，使用微分可以在很多问题中方便的。对于直接求导数，我们可以应用链式法则，这个链式法则类似于一般的微积分中链式法则。但是我们要小心地使用这个链式法则，因为矩阵之间的乘法对维度有要求。

定理 8

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

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

证明.

在定义9和定理6的前提下。我们应用这个定理在每一对相关向量上

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

将他们合并起来则有

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

再次应用定理6则有

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

□

定理 9

考虑如下一个链式依赖：x, z 是两个标量，$\boldsymbol{y}$ 是一个列向量。并且 $z = z(\boldsymbol{y}), y_i = y_i(x), i = 1, 2, \ldots, n$ 应用链式法则，我们则有

$$\frac{\partial z}{\partial x} = \frac{\partial \boldsymbol{y}^{\mathrm{T}}}{\partial x} \frac{\partial z}{\partial \boldsymbol{y}} = \sum_{i=1}^{n} \frac{\partial y_i}{\partial x} \frac{\partial z}{\partial y_i}$$

现在我们来解释一下背后的直觉。x, z 都是标量，所以我们基本上是在计算一般的微积分中的导数。另外，我们有一组中间变量 y_i。$\frac{\partial y_i}{\partial x} \frac{\partial z}{\partial y_i}$ 只是应用标量链式法则的结果 $x \to y_i \to z$。不同中间变量的分离结果是可加的！

例 18

对于线性回归，我们考虑的优化问题为

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

我们将其改写成 $\min_{\boldsymbol{\theta}} \|\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}\|_2^2$，其中 $\boldsymbol{X} = \left(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\right)^{\mathrm{T}}, \boldsymbol{y} = \left(y_1, y_2, \ldots, y_n\right)^{\mathrm{T}}$ 则有

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

例 19

在强化学习中我们需要对一个值函数进行近似时我们考虑优化问题

$$\min_{\boldsymbol{\theta}} E_{\boldsymbol{x}\sim\pi}[\| V^{\pi}(\boldsymbol{x}) - V_{\boldsymbol{\theta}}(\boldsymbol{x})\|_2^2]$$

其中

$$V_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^T\boldsymbol{x}$$

那么就有

$$\begin{aligned}
&\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{x}\sim\pi}[\| V^{\pi}(\boldsymbol{x}) - V_{\boldsymbol{\theta}}(\boldsymbol{x})\|_2^2] \\
=&E_{\boldsymbol{x}\sim\pi}[\nabla_{\boldsymbol{\theta}}\| V^{\pi}(\boldsymbol{x}) - V_{\boldsymbol{\theta}}(\boldsymbol{x})\|_2^2] \\
=&-E_{\boldsymbol{x}\sim\pi}[2(V^{\pi}(\boldsymbol{x}) - V_{\boldsymbol{\theta}}(\boldsymbol{x}))^{\mathrm{T}}\nabla V_{\boldsymbol{\theta}}(\boldsymbol{x})] \\
=&-E_{\boldsymbol{x}\sim\pi}[2(V^{\pi}(\boldsymbol{x}) - V_{\boldsymbol{\theta}}(\boldsymbol{x}))^{\mathrm{T}}\boldsymbol{x}]
\end{aligned}$$

例 20

计算 $(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$ 关于 $\boldsymbol{\mu}$ 的导数。其中 $\boldsymbol{\Sigma}^{-1}$ 是对称矩阵

$$
\begin{aligned}
&\frac{\partial\left[(\boldsymbol{x}-\boldsymbol{\mu})^{T}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]}{\partial\boldsymbol{\mu}}\\
=&\frac{\partial[(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}]}{\partial\boldsymbol{\mu}}\frac{\partial\left[(\boldsymbol{x}-\boldsymbol{\mu})^{T}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]}{\partial[\boldsymbol{x}-\boldsymbol{\mu}]}\\
=&\frac{\partial[(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}]}{\partial\boldsymbol{\mu}}2\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\\
=&-I2\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\\
=&-2\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})
\end{aligned}
$$

18.4.2 一些有用的公式

- $$\frac{\partial}{\partial \boldsymbol{X}} \mathsf{Tr}(f(\boldsymbol{X})) = \mathsf{Tr}(\frac{\partial f(\boldsymbol{X})}{\partial \boldsymbol{X}})^{\mathrm{T}}$$
- $$\frac{\partial}{\partial \boldsymbol{X}} \det(f(\boldsymbol{X})) = \det(f(\boldsymbol{X}))\mathsf{Tr}(f^{-1}(\boldsymbol{X})\frac{\partial f(\boldsymbol{X})}{\partial \boldsymbol{X}})$$
- $$\frac{\partial}{\partial \boldsymbol{X}} f^{-1}(\boldsymbol{X}) = -f^{-1}(\boldsymbol{X})\frac{\partial f(\boldsymbol{X})}{\partial \boldsymbol{X}} f^{-1}(\boldsymbol{X})$$
- $$\frac{\partial \boldsymbol{a}^T \boldsymbol{X}^{-1} \boldsymbol{b}}{\partial \boldsymbol{X}} = -\boldsymbol{X}^{-T} \boldsymbol{a}\boldsymbol{b}^T \boldsymbol{X}^{-T}, \boldsymbol{X}^{-T} = \boldsymbol{X}^{-1\,T}$$

- $$\frac{\partial \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{x}^{\mathrm{T}}\boldsymbol{a}}{\partial \boldsymbol{x}} = \boldsymbol{a}$$
- $$\frac{\partial \boldsymbol{a}\boldsymbol{X}\boldsymbol{b}}{\partial \boldsymbol{X}} = \boldsymbol{a}\boldsymbol{b}^{T}$$
- $$\frac{\partial exp(\boldsymbol{a}^{T}\boldsymbol{X}\boldsymbol{b})}{\partial \boldsymbol{X}} = \boldsymbol{a}\boldsymbol{b}^{T}exp(\boldsymbol{a}^{T}\boldsymbol{X}\boldsymbol{b})$$
- $$\frac{\partial \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{x} = (\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}})\boldsymbol{x}$$

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

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，可以使用微积分和链式规则获得模型参数的梯度。

例 21

考虑这个函数

$$f(x) = \sqrt{x^2 + \exp(x^2)} + \cos(x^2 + \exp(x^2)). \tag{8}$$

应用链式法则，注意微分是线性的，计算梯度。

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

用这种明确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（Kelley，1960；Bryson，1961；Dreyfus，1962；Rumelhart 等人，1986）是计算与模型参数相关的误差函数梯度的有效方法。

18.5.1 神经网络中的梯度（计算图）

例 22

例如，如果我们有输入 $\boldsymbol{x}$ 和观测 $\boldsymbol{y}$，那么网络结构定义为

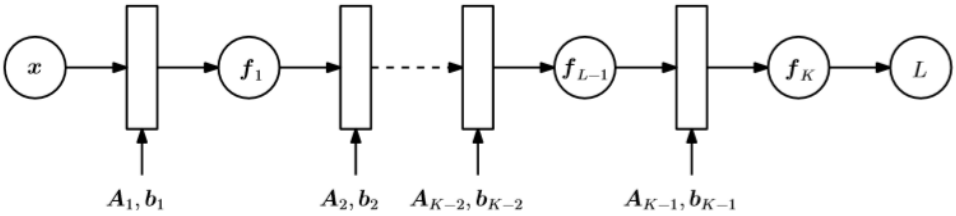


图 1: 多层神经网络中的正向传递，用于计算作为输入 x 和参数 $\boldsymbol{A}_i$，b_i 的损失函数 L.

观察 y 和由网络结构定义

$$\boldsymbol{f}_0 := \boldsymbol{x} \tag{9}$$

$$\boldsymbol{f}_i := \delta_i(\boldsymbol{A}_{i-1}\boldsymbol{f}_{i-1} + \boldsymbol{b}_{i-1}), i = 1, \cdots, K, \tag{10}$$

我们假设使用平方损失函数

$$L(\theta) = \|\boldsymbol{y} - \boldsymbol{f}_K(\theta, x)\|^2 \tag{11}$$

其中 $\theta = \{\boldsymbol{A}_0, \boldsymbol{b}_0, \cdots, \boldsymbol{A}_{K-1}, \boldsymbol{b}_{K-1}\}$，我们需要找到 $\boldsymbol{A}_j, \boldsymbol{b}_j$ 是的平方损失函数最小化。为了获得相对于参数集 θ 的梯度，我们需要 L 关于参数 $\theta_j = \{\boldsymbol{A}_j, \boldsymbol{b}_j\}$ 的每层（$j = 0, \cdots, K-1$）偏导数。

使用链式法则，有

$$\frac{\partial L^{\mathrm{T}}}{\partial f_{K-1}} = \frac{\partial f_K^{\mathrm{T}}}{\partial f_{K-1}} \frac{\partial L^{\mathrm{T}}}{\partial f_K} \tag{12}$$

$$\frac{\partial L^{\mathrm{T}}}{\partial f_{K-2}} = \frac{\partial f_{K-1}^{\mathrm{T}}}{\partial f_{K-2}} \frac{\partial f_K^{\mathrm{T}}}{\partial f_{K-1}} \frac{\partial L^{\mathrm{T}}}{\partial f_K} \tag{13}$$

$$\frac{\partial L^{\mathrm{T}}}{\partial f_{K-3}} = \frac{\partial f_{K-2}^{\mathrm{T}}}{\partial f_{K-3}} \frac{\partial f_{K-1}^{\mathrm{T}}}{\partial f_{K-2}} \frac{\partial f_K^{\mathrm{T}}}{\partial f_{K-1}} \frac{\partial L^{\mathrm{T}}}{\partial f_K} \tag{14}$$

$$\frac{\partial L^{\mathrm{T}}}{\partial f_i} = \frac{\partial f_{i+1}^{\mathrm{T}}}{\partial f_i} \frac{\partial f_{i+2}^{\mathrm{T}}}{\partial f_{i+1}} \cdots \frac{\partial f_K^{\mathrm{T}}}{\partial f_{K-1}} \frac{\partial L^{\mathrm{T}}}{\partial f_K} \tag{15}$$

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显示了渐变通过网络向后传递。

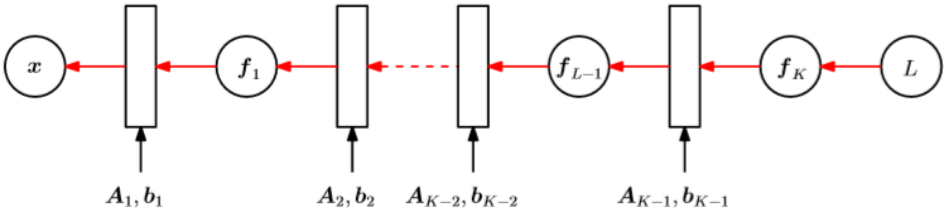


图 2: 三阶张量的 3-模式向量积的原理图

18.5.2 自动微分

反向传播是自动微分的数值分析中的特例。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。

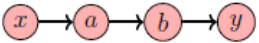


图 3: 一个简单的计算图，显示了数据从 x，经过中间变量，最终到 y

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 $\frac{\mathrm{d}y}{\mathrm{d}x}$ ，我们将应用链式法则，最终得到：

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{\mathrm{d}y}{\mathrm{d}b}\frac{\mathrm{d}b}{\mathrm{d}a}\frac{\mathrm{d}a}{\mathrm{d}x} \tag{16}$$

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，我们可以选择等式(17)或(18)。

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \left(\frac{\mathrm{d}y}{\mathrm{d}b}\frac{\mathrm{d}b}{\mathrm{d}a}\right)\frac{\mathrm{d}a}{\mathrm{d}x} \tag{17}$$

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{\mathrm{d}y}{\mathrm{d}b}\left(\frac{\mathrm{d}b}{\mathrm{d}a}\frac{\mathrm{d}a}{\mathrm{d}x}\right) \tag{18}$$

等式(17)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(18)将是正向模式，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。下面看一个例子：

例 23

从(8)中考虑函数

$$f(x) = \sqrt{x^2 + \exp(x^2)} + \cos(x^2 + \exp(x^2)) \tag{19}$$

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：

$$a = x^2 \tag{20}$$

$$b = \exp(a) \tag{21}$$

$$c = a + b \tag{22}$$

$$d = \sqrt{c} \tag{23}$$

$$e = \cos(c) \tag{24}$$

$$f = d + e \tag{25}$$

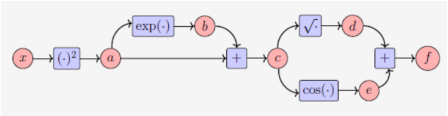


图 4: 输入 x, 函数 f 以及中间变量为 a,b,c,d,e 的计算图.

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(8)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:

$$\frac{\partial a}{\partial x} = 2x \tag{26}$$

$$\frac{\partial b}{\partial a} = \exp(a) \tag{27}$$

$$\frac{\partial c}{\partial a} = 1 = \frac{\partial c}{\partial b} \tag{28}$$

$$\frac{\partial d}{\partial c} = \frac{1}{2\sqrt{c}} \tag{29}$$

$$\frac{\partial e}{\partial c} = -\sin(c) \tag{30}$$

$$\frac{\partial f}{\partial d} = 1 = \frac{\partial f}{\partial e} \tag{31}$$

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：

$$\frac{\partial f}{\partial c} = \frac{\partial f}{\partial d}\frac{\partial d}{\partial c} + \frac{\partial f}{\partial e}\frac{\partial e}{\partial c} \tag{32}$$

$$\frac{\partial f}{\partial b} = \frac{\partial f}{\partial c}\frac{\partial c}{\partial b} \tag{33}$$

$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial b}\frac{\partial b}{\partial a} + \frac{\partial f}{\partial c}\frac{\partial c}{\partial a} \tag{34}$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial a}\frac{\partial a}{\partial x} \tag{35}$$

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

$$\frac{\partial f}{\partial c} = 1 \cdot \frac{1}{2\sqrt{c}} + 1 \cdot (-\sin(c)) \tag{36}$$

$$\frac{\partial f}{\partial b} = \frac{\partial f}{\partial c} \cdot 1 \tag{37}$$

$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial b} exp(a) + \frac{\partial f}{\partial c} \cdot 1 \tag{38}$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial a} \cdot 2x \tag{39}$$

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，因为导数的数学表达式要比函数 $f(x)$ 的数学表达式复杂得多。

反向传播和链式法则

深度学习的根本问题是优化学习函数 F。我们选择权重来最小化损失函数 L。$L(\boldsymbol{x})$ 将计算出的输出 $\boldsymbol{w} = F(\boldsymbol{x}, \boldsymbol{v})$ 与输入 $\boldsymbol{v}$ 的真实分类之间的所有损失 $l(\boldsymbol{w} - true) = l(F(\boldsymbol{x}, \boldsymbol{v}) - true)$ 加起来。我们可以使用微积分求解使 L 最小化的权重。

L 关于权重 $\boldsymbol{x}$ 的偏导数应为零。

要知道所有变量的梯度变化需要当前权重值的导数 (F 梯度的分量)。通过偏导数 $\partial F/\partial \boldsymbol{x}$ 可以计算 $\partial L/\partial \boldsymbol{x}$。从这些信息中, 我们转移到损失较小的新权重。然后我们以新的权重值重新计算 F 和 L 的导数, 然后重复。

反向传播是一种快速计算导数的方法。

使用链式法则：

$$\frac{dF}{dx} = \frac{d}{dx}(F_3(F_2(F_1(x)))) = (\frac{dF_3}{dF_2}(F_2(F_1(x))))(\frac{dF_2}{dF_1}(F_1(x)))(\frac{dF_1}{dx}(x))$$

上述链式法则的适用于一个变量的函数 $F(x)$。但是在神经网络的每一层上都有许多节点 (神经元), 所以我们有的是有许多变量的函数。链式法则仍然适用，除非现在我们在每个步骤都有一个矩阵 (有时是 3 维张量)。这导致两个问题：

1. 什么是多元链式法则？
2. 乘法的哪个顺序 (沿着链的向前或向后) 更快？

我们首先立即回答问题 2

假设链式法则具有三个因子 $\boldsymbol{M}_1\boldsymbol{M}_2\boldsymbol{w}$-两个矩阵和一个向量。我们要先乘矩阵 $\boldsymbol{M}_1\boldsymbol{M}_2$ 还是先乘矩阵 $\boldsymbol{M}_2\boldsymbol{w}$?

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

$(\boldsymbol{M}_1, \boldsymbol{M}_2)\boldsymbol{w}$ 需要 $N^3 + N^2$ 乘法.$\boldsymbol{M}_1(\boldsymbol{M}_2\boldsymbol{w})$ 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:

- 前向 $((\boldsymbol{M}_1\boldsymbol{M}_2)\boldsymbol{M}_3)\ldots\boldsymbol{M}_L)\boldsymbol{w}$ 需要 $(L-1)N^3 + N^2$
- 向后 $\boldsymbol{M}_1(\boldsymbol{M}_2(\ldots(\boldsymbol{M}_w)))$ 需要 LN^2

反向传播已经被发现了很多次, 另一个名字是自动微分 (AD)。这些步骤可以通过两种基本方式进行安排：正向模式和反向模式。正确选择模式可能会在成本上产生很大差异 (成千上万倍)。该选择取决于您是否具有取决于几个输入的许多函数, 还是取决于许多输入的几个函数。

深度学习有一个取决于许多权重的损失函数。

为了最小化函数 $F(\boldsymbol{x})$, 我们需要导数。我们希望每一步的下降方向最陡。对于从 $\boldsymbol{x}_k$ 离开的步骤 k, 该方向由当前点 $\boldsymbol{x}_k$ 处的梯度向量 $\nabla_x F = (\partial F/\partial \boldsymbol{x}_1, \ldots, \partial F/\partial \boldsymbol{x}_N)$ 给出

多变量链式法则

假设向量 $\boldsymbol{v}$ 具有 n 个分量 $\boldsymbol{v}_k$, 是向量 $\boldsymbol{u}$ 具有 m 个分量 $\boldsymbol{u}_j$ 的函数。每个 $\boldsymbol{v}_i$ 的导数在导数矩阵 $\partial \boldsymbol{v}/\partial \boldsymbol{u}$(通常称为雅可比矩阵 $\boldsymbol{J}$) 中表示为 $\boldsymbol{v}_j$ 的形式。它的形状是 $n \times m$(不是 $m \times n$)。类似地, 向量函数 $\boldsymbol{w} = (w_1, \ldots, w_p)$ 的导数矩阵 $d\boldsymbol{w}/d\boldsymbol{v}$ 相对于 $\boldsymbol{v} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n)$ 的分量为 $p \times n$ 矩阵。

$$\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}} = \begin{pmatrix} \frac{\partial w_1}{\partial \boldsymbol{v}_1} & \cdots & \frac{\partial w_1}{\partial \boldsymbol{v}_n} \\ \cdots & & \\ \frac{\partial w_p}{\partial \boldsymbol{v}_1} & \cdots & \frac{\partial w_p}{\partial \boldsymbol{v}_n} \end{pmatrix}, \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{u}} = \begin{pmatrix} \frac{\partial v_1}{\partial \boldsymbol{u}_1} & \cdots & \frac{\partial v_1}{\partial \boldsymbol{u}_m} \\ \cdots & & \\ \frac{\partial v_n}{\partial \boldsymbol{u}_1} & \cdots & \frac{\partial v_n}{\partial \boldsymbol{u}_m} \end{pmatrix}$$

每个 $\boldsymbol{w}_i$ 取决于 $\boldsymbol{v}_j$, 每个 $\boldsymbol{v}_j$ 取决于 $\boldsymbol{u}$。因此, 每个函数 $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_p$ 取决于 $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m$:。链式法则目的是计算导数 $\partial w_i/\partial \boldsymbol{u}_k$。链式法则的形式恰好是一个点积:

$$\frac{\partial w_i}{\partial u_k} = \frac{\partial w_i}{v_1}\frac{\partial v_1}{\partial u_k} + \cdots + \frac{\partial w_i}{v_n}\frac{\partial v_n}{\partial u_k} = \langle (\frac{\partial w_i}{v_1}, \ldots, \frac{\partial w_i}{\partial v_n}), (\frac{\partial v_1}{u_k}, \ldots, \frac{\partial v_n}{\partial u_k}) \rangle$$

故多变量链式法则:

$$\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{u}} = \langle \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}}, \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{u}} \rangle$$

深度学习中的权重 $\boldsymbol{A}$ 实际上是矩阵而不是向量。每个权重都有两个指标, 而不是一个。导数 $\partial w_i/\partial \boldsymbol{A}_{jk}$ 将具有三个索引, 而不是两个。因此, 在实际的链式法则中, 我们有三维张量。幸运的是, 它们是特别简单的张量。它们有很多零元素, 我们仍可以使用链式法则:

$$\frac{\partial w_i}{\partial \boldsymbol{A}_{jk}} = \sum_{t=1}^{n} \frac{\partial w_i}{\partial v_t} \frac{\partial v_t}{\partial A_{jk}}$$

在分别计算每个 $\partial F/\partial t_i$ 时反向传播进行了非常有效的改进。令人难以置信的是, 重新组织计算竟可以产生如此巨大的变化。怀疑者可能会说, 必须为每个步骤计算导数, 然后乘以链式法则。但是重新排序的工作和 N 个导数的计算量远远少于计算一个导数 $\partial F/\partial x_1$ 的 N 倍。

反向传播是后向模式自动微分 (反向模式 AD)。

以什么顺序计算矩阵 $\boldsymbol{ABC}$ 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 $\boldsymbol{A}$ 乘以 $\boldsymbol{B}$ 乘以 $\boldsymbol{C}$, 则结合律为乘法顺序提供了两种选择：

- 首先计算 $\boldsymbol{AB}$ 还是 $\boldsymbol{BC}$?
- 计算 $(\boldsymbol{AB})\boldsymbol{C}$ 还是 $\boldsymbol{A}(\boldsymbol{BC})$?

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 $\boldsymbol{A}$ 为 $m \times n, \boldsymbol{B}$ 为 $n \times p$ 以及 $\boldsymbol{C}$ 为 $p \times q$。

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

因此我们比较 $mp(n+g)$ 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, 第一种方法更快。

这是一个极端情况也是极其重要的情况。假设 $\boldsymbol{c}$ 是列向量：$\boldsymbol{p}$ 乘以 1。因此 $q=1$. 是否应将 $\boldsymbol{Bc}$ 乘以另一个列向量 $(n\times 1)$, 然后乘 $\boldsymbol{A}(\boldsymbol{Bc})$ 来得到输出 $(m\times 1)$? 还是应该先乘 $\boldsymbol{AB}$? 这个问题几乎可以直接回答。正确的 $\boldsymbol{A}(\boldsymbol{Bc})$ 在每个步骤都会产生一个向量。矩阵向量乘法 $\boldsymbol{Bc}$ 需要 np 步。下一个矩阵向量乘法 $\boldsymbol{A}(\boldsymbol{Bc})$ 需要 mn 步。将这些 $np+mn$ 个步骤与从矩阵矩阵乘法 $\boldsymbol{AB}$ 的成本进行比较 (mnp 个步骤！)。正常人就不会这么做。

但是如果 $\boldsymbol{A}$ 是行向量，那么 $(\boldsymbol{AB})\boldsymbol{c}$ 会更好。

这将与深度学习的核心计算相匹配：训练网络 = 优化权重。深度网络的输出 $F(\boldsymbol{v})$ 是一个以 $\boldsymbol{v}$ 开头的链：

$$F(\boldsymbol{u})=\boldsymbol{A}_L\boldsymbol{v}_{L-1}=\boldsymbol{A}_L(R\boldsymbol{A}_{L-1}(\ldots(R\boldsymbol{A}_2(R\boldsymbol{A}_1\boldsymbol{v}))))$$

对于 $\boldsymbol{A}_L\boldsymbol{v}_{L-1}$ 中的矩阵 $\boldsymbol{A}_L$,F 关于矩阵 $\boldsymbol{A}$(和偏置向量 $\boldsymbol{b}$) 的导数最容易求的。$\boldsymbol{Av}$ 关于 $\boldsymbol{A}$ 的导数是被表示为 $\boldsymbol{v}$：$\frac{\partial F_i}{\partial A_{jk}}=\delta_{ij}\boldsymbol{v}_k$。接下来是 $\boldsymbol{A}ReLU(\boldsymbol{A}_{L-1}\boldsymbol{v}_{L-1})$ 关于 $\boldsymbol{A}_{L-1}$ 的导数

我们将解释在直接方法与伴随方法的优化中 (选择合适的权重), 反向模式将如何出现。

学习函数 $F(\boldsymbol{x}, \boldsymbol{v})$ 的偏导 $\partial F/\partial \boldsymbol{x}$

权重 $\boldsymbol{x}$ 由所有矩阵 $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_L$ 和偏差向量 $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_L$ 组成。输入 $\boldsymbol{v} = \boldsymbol{v}_0$ 是训练数据。输出 $\boldsymbol{w} = F(\boldsymbol{x}, V0)$ 出现在 L 层中。因此 $\boldsymbol{w} = \boldsymbol{v}_L$ 是神经网络在隐藏层中 $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{L-1}$ 后的最后一步。

每一层 $\boldsymbol{v}_n$ 来自上一层 $R(\boldsymbol{b}_n + \boldsymbol{A}_n \boldsymbol{v}_{n-1})$。在此,$R$ 是应用于每一个分量的非线性激活函数(通常为 $ReLU$)。

因此, 深度学习将我们从 $\boldsymbol{v} = \boldsymbol{v}_0$ 带到 $\boldsymbol{w} = \boldsymbol{v}_L$, 然后将 $\boldsymbol{w}$ 代入损失函数以测量该样本 $\boldsymbol{v}$ 的误差。这可能是分类误差：0 代替 1, 或者 1 代替 0。最小二乘回归误差 $\|\boldsymbol{g} - \boldsymbol{w}\|_2$(用 $\boldsymbol{w}$ 代替期望的输出 $\boldsymbol{g}$)。通常它是一个交叉熵。总损失 $L(\boldsymbol{x})$ 是所有输入向量上的损失之和。

深度学习的目的是找到使 L 最小的权重。对于梯度下降, 损失为 $L(\boldsymbol{x})$。对于随机梯度下降, 每次迭代后的损失为 $l(\boldsymbol{x})$, 这来自单个输入或一小部分输入。在所有情况下, 我们都需要输出 $\boldsymbol{w}$(最后一层的分量) 关于 $\boldsymbol{x}$ 的导数 $\partial \boldsymbol{w}/\partial \boldsymbol{x}$。

这是深度学习如此昂贵并且在 GPU 上花费如此长时间的原因之一。

$\partial F/\partial \boldsymbol{x}$ 的计算：显式公式

我们从产生最终输出 $\boldsymbol{v}_L = \boldsymbol{w}$ 的最后一个偏置向量 $\boldsymbol{b}_L$ 和权重矩阵 $\boldsymbol{A}_L$ 开始。该层没有非线性, 因此我们删除了层索引 L:

$VL = \boldsymbol{b}_L + \boldsymbol{A}_L \boldsymbol{v}_{L-1}$ 或简单地 $\boldsymbol{w} = \boldsymbol{b} + \boldsymbol{A}\boldsymbol{v}$。

我们的目标是计算 $\boldsymbol{b} + \boldsymbol{A}\boldsymbol{v}$ 所有分量的导数 $\partial \boldsymbol{w}_i/\partial \boldsymbol{b}_j$ 和 $\partial \boldsymbol{w}_i/\partial \boldsymbol{A}_{jk}$。当 j 与 i 不同时, 第 i 个输出 $\boldsymbol{w}_i$ 不受 $\boldsymbol{b}_j$ 或 $\boldsymbol{A}_{jk}$ 的影响。将 $\boldsymbol{A}$ 乘以 $\boldsymbol{v}$, 则 $\boldsymbol{A}$ 的第 j 行乘以 $\boldsymbol{w}_j$ 而不是 $\boldsymbol{w}_i$。我们引入的符号 δ 为 1 或 0:

$$\delta = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

单位矩阵的每一个元素就是 δ_{ij}。

I 的列是 one-hot 向量！导数是 1 或 0 或 $\boldsymbol{v}_k$。

$$\frac{\partial \boldsymbol{w}_i}{\partial \boldsymbol{b}_j} = \delta_{ij}, \frac{\boldsymbol{w}_i}{\partial \boldsymbol{A}_{jk}} = \delta_{ij}\boldsymbol{v}_k$$

例 24

在 $\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} + \begin{pmatrix} a_{11}v_1 + a_{12}v_2 \\ a_{21}v_1 + a_{22}v_2 \end{pmatrix}$ 中有六个 $\boldsymbol{b}$ 和 a。w_1 的梯度为

$$\frac{\partial w_1}{\partial b_1} = 1, \frac{\partial w_1}{\partial b_2} = 0, \frac{\partial w_1}{\partial a_{11}} = \boldsymbol{v}_1, \frac{\partial w_1}{\partial a_{12}} = \boldsymbol{v}_2, \frac{\partial w_1}{\partial a_{21}} = \frac{\partial w_1}{\partial a_{22}} = 0,$$

隐藏层中的梯度

现在假设有一个隐藏层, 所以 $L=2$。输出为 $\boldsymbol{w}=\boldsymbol{v}_L=\boldsymbol{v}_2$, 隐藏层包含 $\boldsymbol{v}_1$, 输入为 $\boldsymbol{v}_0=\boldsymbol{v}$。非线性 R 可能是 $ReLU$。

$$\boldsymbol{v}_1=R(\boldsymbol{b}_1+\boldsymbol{A}_1\boldsymbol{v}_0), w=\boldsymbol{b}_2+\boldsymbol{A}_2\boldsymbol{v}_1=\boldsymbol{b}_2+\boldsymbol{A}_2R(\boldsymbol{b}_1+\boldsymbol{A}_1\boldsymbol{v}_0)$$

公式 $\frac{\partial \boldsymbol{w}_i}{\partial \boldsymbol{b}_j}=\delta_{ij}, \frac{\boldsymbol{w}_i}{\partial \boldsymbol{A}_{jk}}=\delta_{ij}\boldsymbol{v}_k$ 仍然给出了 $\boldsymbol{w}$ 关于最后权重 $\boldsymbol{b}_2$ 和 $\boldsymbol{A}_2$ 的导数。输出端不存在函数 R,$\boldsymbol{v}$ 为 $\boldsymbol{v}_1$。但是 $\boldsymbol{w}$ 关于 $\boldsymbol{b}_i$ 和 $\boldsymbol{A}_1$ 的导数确实包含作用于 $\boldsymbol{b}_1+\boldsymbol{A}_1\boldsymbol{v}_0$ 的非线性函数 R。因此 $\partial \boldsymbol{w}/\partial \boldsymbol{A}_1$ 中的导数需要链式法则 $(\partial f/\partial g)(\partial g/\partial x)$;

链式法则

$$\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{A}_1}=\frac{\partial[\boldsymbol{A}_2R(\boldsymbol{b}_1+\boldsymbol{A}_1\boldsymbol{v}_0)]}{\partial \boldsymbol{A}_1}=\boldsymbol{A}_2R'(\boldsymbol{b}_1+\boldsymbol{A}_1\boldsymbol{v}_0)\frac{\partial(\boldsymbol{b}_1+\boldsymbol{A}_1\boldsymbol{v}_0)}{\partial \boldsymbol{A}_1}$$

该链式法则具有三个因式。从层 $L-2=0$ 处的 $\boldsymbol{v}_0$ 开始, 权重 $\boldsymbol{b}_1$ 和 $\boldsymbol{A}_1$ 将我们引向 $L-1=1$ 层。该步骤的导数与公式 $\frac{\partial \boldsymbol{w}_i}{\partial \boldsymbol{b}_j}=\delta_{ij}$, $\frac{\boldsymbol{w}_i}{\partial \boldsymbol{A}_{jk}}=\delta_{ij}\boldsymbol{v}_k$ 完全相同。但是该部分步骤的输出不是 $\boldsymbol{v}_{L-1}$。要找到该隐藏层, 我们首先必须应用 R。因此, 链式法则包括其导数 R‘。然后, 最后的步骤 (到 $\boldsymbol{w}$) 乘以最后的权重矩阵 $\boldsymbol{A}_2$。

将这些公式扩展到 L 层是可以的。但是因为有池化和批归一化, 自动微分似乎无法克服硬编码。

注意, 上面的公式从 $\boldsymbol{w}$ 倒退到 $\boldsymbol{v}$。自动反向传播也会这样做。“反向模式”是从输出开始的。

导数 $\partial \boldsymbol{w}/\partial \boldsymbol{A}_1$ 的详情

我们需要更仔细地研究上面的公式。它的非线性部分 R' 来自非线性激活函数的导数。通常的选择是斜坡函数 $ReLU(\boldsymbol{x}) = (\boldsymbol{x})_+ = S$ 形 $sigmoid$ 函数的极限情况。

$$dR/dx = \begin{cases} 0 & x < 0 \\ & \\ 1 & x > 0 \end{cases}, R'(\boldsymbol{b}_1 + \boldsymbol{A}\boldsymbol{v}_0) = \begin{cases} 0 & \boldsymbol{b}_1 + \boldsymbol{A}\boldsymbol{v}_0 < 0 \\ & \\ 1 & \boldsymbol{b}_1 + \boldsymbol{A}\boldsymbol{v}_0 > 0 \end{cases}$$

回到上述公式, 将在最后一个隐藏层得到的积 $\boldsymbol{A}_{L-1}$ 和向量 $\boldsymbol{b}_{L-1}$ 写成 $\boldsymbol{A}$ 和 $\boldsymbol{b}$。然后 $ReLU$ 和 $\boldsymbol{A}_L$ 和 $\boldsymbol{b}_L$ 产生最终输出 $\boldsymbol{w} = \boldsymbol{v}_L$。我们的兴趣是 $\partial \boldsymbol{w}/\partial \boldsymbol{A}$, 即 $\boldsymbol{w}$ 对倒数第二个权重矩阵的依赖性。

$$w = \boldsymbol{A}_L(R(\boldsymbol{A}\boldsymbol{v} + \boldsymbol{b})) + \boldsymbol{b}_L, \frac{\partial w}{\partial A} = \boldsymbol{A}_L R'(\boldsymbol{A}\boldsymbol{v} + \boldsymbol{b}) \frac{\partial(\boldsymbol{A}\boldsymbol{v} + \boldsymbol{b})}{\partial \boldsymbol{A}}$$

我们认为 $ReLU$ 函数的对角矩阵 R 逐个分量地作用于 $\boldsymbol{A}\boldsymbol{v}+\boldsymbol{b}$。那么 $R'(\boldsymbol{A}\boldsymbol{v}+\boldsymbol{b})$ 是一个对角矩阵, 其中正分量为 1, 负分量为 0。

$$w=\boldsymbol{A}_L R(\boldsymbol{A}\boldsymbol{v}+\boldsymbol{b}), \frac{\partial w}{\partial A}=\boldsymbol{A}_L J\frac{\partial(\boldsymbol{A}\boldsymbol{v}+\boldsymbol{b})}{\partial \boldsymbol{A}}$$

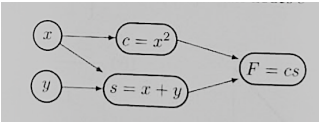
我们知道第三个因式的每个分量 ($\boldsymbol{v}_k$ 或零) 来自于公式 $\frac{\partial \boldsymbol{w}_i}{\partial \boldsymbol{b}_j}=\delta_{ij}$, $\frac{\boldsymbol{w}_i}{\partial \boldsymbol{A}_{jk}}=\delta_{ij}\boldsymbol{v}_k$。

当 Sigmoid 函数 R_a 代替 $ReLU$ 函数时, 对角矩阵 $J=R_a'(\boldsymbol{A}\boldsymbol{v}+\boldsymbol{b})$ 不再包含 1 和 0。现在我们评估在 $\boldsymbol{A}\boldsymbol{v}+\boldsymbol{b}$ 的每个分量处的导数 dR_a/dx。

在实践中。反向传播会自动计算所有 $\boldsymbol{A}$ 和 $\boldsymbol{b}$ 的导数。

计算图

假设 $F(\boldsymbol{x}, \boldsymbol{y})$ 是两个变量 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 的函数。这些输入是计算图中的前两个节点。计算中的一个典型步骤 (图中的一个边) 是算术运算 (加法, 减法, 乘法等) 之一。最终输出是函数 $F(\boldsymbol{x}, \boldsymbol{y})$。我们的示例将是 $F = \boldsymbol{x}_2(\boldsymbol{x} + \boldsymbol{y})$。
这是用中间节点 $c = x^2$ 和 $s = x + y$ 计算 F 的图：



当我们有输入 $\boldsymbol{x}$ 和 $\boldsymbol{y}$, 例如 $\boldsymbol{x} = 2$ 和 $\boldsymbol{y} = 3$ 时, 边导致 $c = 4$ 和 $s = 5$ 且 $F = 20$。

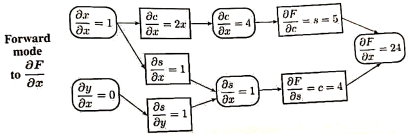
现在, 我们计算图中每个步长的导数。从 $\boldsymbol{x}$ 导数开始。首先, 我们选择前进模式。从输入 $\boldsymbol{x}$ 开始, 向输出函数 $\boldsymbol{x}^2(\boldsymbol{x}+\boldsymbol{y})$ 移动。因此, 第一步将幂运算用于 $c=\boldsymbol{x}^2$, 将和运算用于 $s=\boldsymbol{x}+\boldsymbol{y}$。最后一步将乘积运算应用于 $F=cs$。

$$\frac{\partial c}{\partial x}=2x, \frac{\partial s}{\partial x}=1, \frac{\partial F}{\partial c}=s, \frac{\partial F}{\partial s}=c$$

在图中移动会产生链式法则!

$$\frac{\partial F}{\partial x}=\frac{\partial F}{\partial c}\frac{\partial c}{\partial x}+\frac{\partial F}{\partial s}\frac{\partial s}{\partial x}=(s)(2x)+(c)(1)=(5)(4)+(4)(1)=24$$

结果是计算输出 F 相对于输入 $\boldsymbol{x}$ 的导数。



这会有一个相似的和 $\boldsymbol{y}$ 导数相关的图，这是计算 $\partial F/\partial \boldsymbol{y}$ 的正向模式。链式法则在该图中出现 $\boldsymbol{x}=2$ 和 $\boldsymbol{y}=3$ 且 $c=\boldsymbol{x}^2=2^2$ 和 $s=\boldsymbol{x}+\boldsymbol{y}=2+3, F=cs$ 时使用：

$$\frac{\partial F}{\partial y}=\frac{\partial F}{\partial c}\frac{\partial c}{\partial y}+\frac{\partial F}{\partial s}\frac{\partial s}{\partial y}=(s)(0)+(c)(1)=(5)(0)+(4)(1)=4$$

$\partial F/\partial \boldsymbol{y}$ 的计算图未画出, 但有一点很重要：正向模式要求每个输入 $\boldsymbol{x}_i$ 有一个新图, 以计算偏导数 $\partial F/\partial x_i$。

一个输出的反向模式图

反向模式从输出 F 开始。它计算两个输入的导数。计算在图中向后进行。
这意味着对于 $\boldsymbol{x}$ 导数, 它不会计算正向图中 $\partial \boldsymbol{y}/\partial x = 0$ 部分。也不会计算在正向图中 $\partial r/\partial \boldsymbol{y} = 0$ 的部分。一个大且现实的有 N 个输入的问题将有 N 个前向图, 每个图都有 $N-1$ 个为 0 的部分 (因为 N 个输入是独立的)。相对于所有其他输入 x_j 的 $\boldsymbol{x}_i$ 的导数为 $\partial \boldsymbol{x}_i/\partial \boldsymbol{x}_j = 0$。
用一个输出得到一个反向图来代替来自 N 个输入的 N 个正向图。这是反向模式计算图。它计算 F 关于每个节点权重的导数。它以 $\partial F/\partial F = 1$ 开头, 然后进行反向传播。
计算图执行链式法则以计算导数。反向模式通过跟踪从输出到输入的所有链来计算所有导数 $\partial F/\partial \boldsymbol{x}$。这些链全都显示为图上的路径, 而不是以指数形式出现的许多可能路径的单独链式法则。这是反向模式的成功。

伴随方法

在一类最优化问题中, 出现了矩阵乘法 ABC 最佳运算顺序的相同问题。我们求解一个线性方程 $\boldsymbol{E}\boldsymbol{v}=\boldsymbol{b}$ 构成的平方系统。向量 $\boldsymbol{b}$ 取决于变量 $\boldsymbol{p}=(\boldsymbol{p}_1,\ldots,\boldsymbol{p}_M)$。因此, 解向量 $\boldsymbol{v}=E^{-1}\boldsymbol{b}$ 取决于 $\boldsymbol{p}$。包含导数 $\partial\boldsymbol{v}_i/\partial\boldsymbol{p}_j$ 的矩阵 $\partial\boldsymbol{v}/\partial\boldsymbol{p}$ 将为 $N\times M$ 的矩阵。
我们将最小化 $F(\boldsymbol{v})$。向量 $\boldsymbol{v}$ 取决于变量 $\boldsymbol{p}$。因此, 我们需要一个链式法则, 将导数 $\partial F/\partial\boldsymbol{v}_i$ 乘以 $\partial\boldsymbol{v}_i/\partial\boldsymbol{p}$, 下面展示一下这如何变成三个矩阵的乘积, 并且乘法顺序是决定性的。三组导数控制了 F 如何取决于输入变量 $\boldsymbol{p}_i$

- $\boldsymbol{A}=\partial F/\partial\boldsymbol{v}_i$, F 关于 $\boldsymbol{v}_1,\ldots,\boldsymbol{v}_N$ 的导数
- $\boldsymbol{A}=\partial\boldsymbol{v}_i/\partial\boldsymbol{b}_k$, 每个 $\boldsymbol{v}_i$ 对每个 $\boldsymbol{b}_k$ 的导数
- $\boldsymbol{A}=\partial\boldsymbol{b}_k/\partial\boldsymbol{p}_j$, 每个 $\boldsymbol{b}_k$ 关于每个 $\boldsymbol{p}_j$ 的的导数;

为了得到 $\partial \boldsymbol{v}_i/\partial p_j$, 我们计算方程 $E\boldsymbol{v} = \boldsymbol{b}$ 对 $\boldsymbol{p}_j$ 取的导数

$$E\frac{\partial \boldsymbol{v}}{\partial p_j} = \frac{\partial \boldsymbol{b}}{\partial p_j}, j = 1, \ldots, M$$

故，$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{p}} = E^{-1}\frac{\partial \boldsymbol{b}}{\partial \boldsymbol{p}}$ 这似乎我们有 M 个大小为 N 的线性方程。随着我们寻找使 $F(\boldsymbol{v})$ 最小的 $\boldsymbol{p}$ 的选择, 这些线性系统将难以求解。矩阵 $\partial \boldsymbol{v}/\partial \boldsymbol{p}$ 包含 $\boldsymbol{v}_i, \ldots, \boldsymbol{v}_N$ 关于变量 $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_M$ 的导数。

现在假设代价函数 $F(\boldsymbol{v}) = \boldsymbol{c}^T\boldsymbol{v}$ 是线性的 (所以 $\partial F/\partial \boldsymbol{v} = c^T$)。那么最优化实际上需要的是 $F(\boldsymbol{v})$ 相对于 $\boldsymbol{p}$ 的梯度。导数 $\partial F/\partial \boldsymbol{v}$ 就是向量 c^T。

$$\frac{\partial F}{\partial p} = \frac{\partial F}{\partial \boldsymbol{v}} \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{p}} = \boldsymbol{c}^{\mathrm{T}} E^{-1} \frac{\partial \boldsymbol{b}}{\partial \boldsymbol{p}}$$

这是一个关键的式子。我们应该如何计算乘积: 行向量 c^T 乘以 $N \times N$ 矩阵 E 乘以 $N \times M$ 矩阵 $\partial \boldsymbol{b}/\partial \boldsymbol{p}$?

这个问题几乎可以直接回答。我们不想将两个矩阵相乘。因此, 我们毕竟不是在计算 $\partial \boldsymbol{v}/\partial \boldsymbol{p}$。相反, 好的第一步是计算 $c^T E^{-1}$。这将产生一个行向量 λ^{T}。换句话说, 我们求解伴随方程 $E^{\mathrm{T}}\lambda = c$:

伴随方程 $E^{\mathrm{T}}\lambda = c, \lambda^{\mathrm{T}} E = c^{\mathrm{T}}, \lambda^{\mathrm{T}} = c^{\mathrm{T}} E^{-1}$

将 λ^{T} 代入公式中的 $c^T E^{-1}$。最后一步将行向量乘以向量 $\boldsymbol{b}$ 的导数 (梯度):

代价函数 F 的梯度:

$$\frac{\partial F}{\partial \boldsymbol{p}} = \lambda^{\mathrm{T}} \frac{\partial b}{\partial p}$$

所以最佳顺序为 $(AB)C$, 因为第一个因子 A 实际上是行向量 λ^{T}。

这个伴随方法的例子是求 $E\boldsymbol{x} = \boldsymbol{b}$。右侧 $\boldsymbol{b}$ 取决于参数 $\boldsymbol{p}$。因此, 解 $x = E^{-1}\boldsymbol{b}$ 取决于 $\boldsymbol{p}$。那么代价函数 $F(\boldsymbol{x}) = c^T\boldsymbol{x}$ 取决于 $\boldsymbol{p}$。

伴随方程 $\boldsymbol{A}^{\mathrm{T}}\lambda = c$ 找到了将最后两个步骤有效组合的向量。“伴随”与“转置”具有相同的含义, 我们也可以将其应用于微分方程。变量 $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_M$ 可能出现在矩阵 E 中, 或者出现在特征值问题或微分方程中。

这里的重点是强调和加强反向传播的关键思想：反向模式可以以更快的方式对微分计算进行排序。

深层的伴随性和敏感性

进入探究深度学习的问题。参数 $\boldsymbol{x}$ 为 $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ 且在 L 层处的输出 $\boldsymbol{w}\,(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M)$ 时，$\partial \boldsymbol{w}/\partial \boldsymbol{x}_j$ 为多少？从输入 $\boldsymbol{v}_0$ 开始经过 L 步后，可以看到输出 $\boldsymbol{w} = \boldsymbol{v}_L$。我们把第 n 步写成 $\boldsymbol{v}_n = F_n(\boldsymbol{v}_{n-1}, x_n)$ 其中 F_n 取决于权重 (参数)$\boldsymbol{x}_n$。

如果每个步骤都可以使用相同的参数 $\boldsymbol{x}$，这是一个递归关系。深度学习对每个新层都有新的参数，这赋予了它“学习能力”，这和普通递归不同。实际上，典型的递归只是差分方程 $d\boldsymbol{v}/dt = f(\boldsymbol{v}, \boldsymbol{x}, t)$ 的有限差分模拟。

同样在这种情况下，我们希望获得期望的输出 $\boldsymbol{v}(T)$。我们选择参数来使我们的输出接近期望的输出。问题是计算导数 $J = \partial \boldsymbol{v}_N/\partial \boldsymbol{v}_M$ 的矩阵。我们必须对上面的等式应用链式法则，从 N 一直回到 0：

$$\boldsymbol{v}_N = F_N(\boldsymbol{v}_{N-1}, \boldsymbol{x}_N) = F_N(F_{N-1}(\boldsymbol{v}_{N-2}, \boldsymbol{x}_{N-1}), \boldsymbol{x}_N)$$

取其关于 $\boldsymbol{x}_{N-1}$ 的导数。查看最后两层：

$$\frac{\partial \boldsymbol{v}_N}{\partial \boldsymbol{x}_{N-1}} = \frac{\partial F_N}{\partial \boldsymbol{v}_{N-1}} \frac{\partial \boldsymbol{v}_{N-1}}{\partial \boldsymbol{x}_{N-1}}, \frac{\partial \boldsymbol{v}_N}{\partial \boldsymbol{x}_{N-2}} = \frac{\partial F_N}{\partial \boldsymbol{v}_{N-1}} \frac{\partial \boldsymbol{v}_{N-1}}{\partial \boldsymbol{x}_{N-2}} = \frac{\partial F_N}{\partial \boldsymbol{v}_{N-1}} \frac{\partial \boldsymbol{v}_{N-1}}{\partial \boldsymbol{x}_{N-2}} \frac{\partial \boldsymbol{v}_{N-2}}{\partial \boldsymbol{x}_{N-2}}$$

最后一个表达式是三元乘积 $\boldsymbol{ABC}$, 所以需要决定计算从 AB 开始还是从 BC 开始？优化的伴随方法和反向传播的反向模式都建议：从 $\boldsymbol{AB}$ 开始。

对于深度学习, 递归关系位于网络的各个层之间。

例 25

假设有如下的神经网络：第一层为卷积层（$2 \times 2 \times 2$

- 第一个输出通道的卷积为

$$\begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

- 第二个输出通道的卷积为

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix}$$

第二层为池化层：每个输出通道后都连接池化层，都采用 2×2 的最大池化。

假设输入为

$$\begin{pmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

对应的输出为 $(130, 180)^{\mathrm{T}}$，采用平方损失 $\frac{1}{5}\|y - \hat{y}\|_2^2$，求卷积层第一个输出通道中第一个卷

解

先计算前向过程。

卷积层的输出结果：

第一个通道输出为 $c^{(1)}=\begin{pmatrix}57 & 73\\ 105 & 121\end{pmatrix}$ 第二个通道输出为 $c^{(2)}=\begin{pmatrix}78 & 102\\ 150 & 174\end{pmatrix}$

池化层输出结果为 $p=\begin{pmatrix}121\\ 174\end{pmatrix}$

计算损失 $l=\frac{1}{2}(9^2+6^2)=\frac{117}{2}$。

记

$$k^{(1)}=\begin{pmatrix}0 & 1\\ 2 & 3\end{pmatrix}, k^{(2)}=\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}$$

我们要求的是

$$\frac{\partial l}{\partial k^{(1)}}$$

然后计算反向传播过程。

$$\frac{\partial l}{\partial p_1} = \frac{\partial}{\partial p_1}\frac{1}{2}((p_1-130)^2+(p_2-180)^2) = p_1 - 130 = -9$$

$$\frac{\partial l}{\partial c^{(1)}} = \frac{\partial p_1}{\partial c^{(1)}}\frac{\partial l}{\partial p_1} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \times (-9) = \begin{pmatrix} 0 & 0 \\ 0 & -9 \end{pmatrix}$$

$$\frac{\partial l}{\partial k^{(1)}} = \frac{\partial c_{22}^{(1)}}{\partial k^{(1)}}\frac{\partial l}{\partial c_{22}^{(1)}} = \begin{pmatrix} 4 & 5 \\ 7 & 8 \end{pmatrix} \times (-9) = \begin{pmatrix} -36 & -45 \\ -63 & -72 \end{pmatrix}$$