# 矩阵求导

矩阵求导 matrix vector derivatives

以下用小写字母代表标量，例如 $x$；小写黑体代表向量，例如 $\mathbf{x}$；大写黑体代表矩阵，例如 $\mathbf{X}$。$\mathbb{R}^{n\times m}$ 为所有 n 行 m 列实数矩阵(有时为了令维数更醒目也记作 $\mathbb{R}(n\times m)$)。

## 标量对矩阵求导、矩阵对标量求导

**标量对矩阵(向量)求导、矩阵(向量)对标量求导，求导后结果与原矩阵(向量)同型。**
标量对矩阵(向量)求导:

若函数 $f(\mathbf{X}):\mathbb{R}^{n\times m}\to\mathbb{R}$， $y=f(\mathbf{X})$，则

$$\frac{\partial y}{\partial \mathbf{X}}=\begin{bmatrix} \frac{\partial y}{\partial x_{11}} & \frac{\partial y}{\partial x_{12}} & \cdots & \frac{\partial y}{\partial x_{1m}} \\ \frac{\partial y}{\partial x_{21}} & \frac{\partial y}{\partial x_{22}} & \cdots & \frac{\partial y}{\partial x_{2m}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y}{\partial x_{n1}} & \frac{\partial y}{\partial x_{n2}} & \cdots & \frac{\partial y}{\partial x_{nm}} \end{bmatrix}\in\mathbb{R}^{n\times m} \tag{0.1}$$

矩阵(向量)对标量求导:

若函数 $f(x):\mathbb{R}\to\mathbb{R}^{n\times m}$， $\mathbf{Y}=f(x)$，则

$$\frac{\partial \mathbf{Y}}{\partial x}=\begin{bmatrix} \frac{\partial y_{11}}{\partial x} & \frac{\partial y_{12}}{\partial x} & \cdots & \frac{\partial y_{1m}}{\partial x} \\ \frac{\partial y_{21}}{\partial x} & \frac{\partial y_{22}}{\partial x} & \cdots & \frac{\partial y_{2m}}{\partial x} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_{n1}}{\partial x} & \frac{\partial y_{n2}}{\partial x} & \cdots & \frac{\partial y_{nm}}{\partial x} \end{bmatrix}\in\mathbb{R}^{n\times m} \tag{0.2}$$

## 向量对向量求导

若函数 $f(\mathbf{x}):\mathbb{R}^{m}\to\mathbb{R}^{n}$， $\mathbf{y}=f(\mathbf{x})$，则

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_m} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial x_1} & \frac{\partial y_n}{\partial x_2} & \cdots & \frac{\partial y_n}{\partial x_m} \end{bmatrix} \in \mathbb{R}^{n\times m} \tag{0.3}$$

## 雅各布(Jacobian)矩阵

设函数 $f(\mathbf{x}):\mathbb{R}^m \to \mathbb{R}^n$， $\mathbf{y} = f(\mathbf{x})$，则 $\mathbf{y}$ 对 $\mathbf{x}$ 的雅各布矩阵 $J$ 定义如下:

$$J = \Delta_x \mathbf{y} = \frac{d\mathbf{y}}{d\mathbf{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial x_1} & \cdots & \frac{\partial y_n}{\partial x_m} \end{bmatrix} \in \mathbb{R}^{n\times m} \tag{0.4}$$

其中:

$$J_{i,j} = \frac{\partial y_i}{\partial x_j} \tag{0.5}$$

## 一些常用矩阵求导公式的推导

$$\left(\frac{\partial \mathbf{A}\mathbf{x}}{\partial \mathbf{x}}\right)_{ij} = \frac{\partial (\mathbf{A}\mathbf{x})_i}{\partial x_j} = \frac{\partial \sum_k a_{ik} x_k}{\partial x_j} = \frac{\partial a_{ij} x_j}{\partial x_j} = a_{ij} \tag{0.6}$$

所以

$$\frac{\partial \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} = \mathbf{A} \tag{0.7}$$

$$\frac{\partial [tr(\mathbf{A}\mathbf{X})]}{\partial \mathbf{X}} = \mathbf{A}^{\mathrm{T}} \tag{0.8}$$

# BP 算法

## 神经网络记号

设神经网络层数为 $n$ 层，编号为 1 至 $n$，其中含有 1 个输入层(记为层 1)、$n-2$ 个隐层(记为层 2~层 $n-1$)，1 个输出层(记为层 $n$)。层 $i$ 也记作 $\mathbf{l}^{(i)} \in \mathbb{R}(s_i)$， $s_i$ 为层 $i$ 的结点数量， $l_j^{(i)}$ 为层 $i$ 的结点 $j$ 。$i$ 层的**输入**向量记为

$$\mathbf{z}^{(i)}=[z_1^{(i)},\cdots,z_{s_i}^{(i)}]^{\mathrm{T}}\in\mathbb{R}(s_i\times 1) \tag{1.1}$$

其中 $z_j^{(i)}$ 代表层 i 的 j 结点($l_j^{(i)}$)的输入，**输出**向量记为

$$\mathbf{a}^{(i)}=[a_1^{(i)},\cdots,a_{s_i}^{(i)}]^{\mathrm{T}}\in\mathbb{R}(s_i\times 1) \tag{1.2}$$

输入层(层 1)的 $\mathbf{a}_1=\mathbf{z}_1=\mathbf{x}\in\mathbb{R}(s_1)$，其中 $\mathbf{x}$ 为输入。

层 i 与层 i+1 的结点之间的**权重**矩阵记为

$$\mathbf{W}^{(i)}=\begin{bmatrix} w_{11}^{(i)} & \cdots & w_{1s_i}^{(i)} \\ \vdots & \ddots & \vdots \\ w_{s_{i+1}1}^{(i)} & \cdots & w_{s_{i+1}s_i}^{(i)} \end{bmatrix}\in\mathbb{R}(s_{i+1}\times s_i) \tag{1.3}$$

其中： $w_{jk}^{(i)}$ 为结点 $l_k^{(i)}$ 到 $l_j^{(i+1)}$ 的权重。

其余各层的输入、输出、权重之间的关系如下：

$$\mathbf{z}^{(i)}=\mathbf{W}^{(i-1)}\mathbf{a}^{(i-1)}\in\mathbb{R}(s_i\times s_{i-1})\times\mathbb{R}(s_{i-1}\times 1)=\mathbb{R}(s_i\times 1) \tag{1.4}$$

$$z_j^{(i)}=\sum_k w_{jk}^{(i-1)}a_k^{(i-1)} \tag{1.5}$$

$$\mathbf{a}^{(i)}=f(\mathbf{z}^{(i)}) \tag{1.6}$$

其中 $f(x)=\dfrac{1}{1+e^{-x}}$，称为 sigmoid 激活函数。

神经网络的输出

$$\mathrm{h}_{\mathbf{w}}(\mathbf{x})=\mathbf{a}^{(n)}=\mathrm{sigmoid}\,\mathbf{z}^{(n)}\in\mathbb{R}(s_n\times 1) \tag{1.7}$$

神经网络输入向量 $\mathbf{x}$ 的标签向量记作 $\mathbf{y}\in\mathbb{R}(s_n\times 1)$

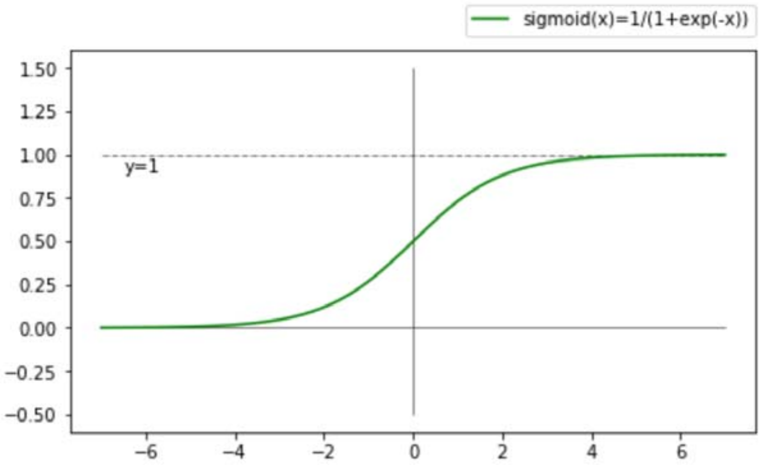
# 激活函数

## sigmoid 激活函数

$$\mathrm{sigmoid}\,x=\frac{1}{1+e^{-x}} \tag{1.8}$$

其导数为：

$$\frac{d(\mathrm{sigmoid}\,x)}{dx}=\mathrm{sigmoid}\,x-(\mathrm{sigmoid}\,x)^2 \tag{1.9}$$




图表 1：sigmoid 激活函数

# 损失函数

损失函数 $\mathrm{Loss}(\mathrm{h}_{\mathbf{w}}(\mathbf{x}),\mathbf{y}):\mathbb{R}(s_n\times 1,s_n\times 1)\to\mathbb{R}^n$ :

$$\mathrm{Loss}(\mathrm{h}_{\mathbf{w}}(\mathbf{x}),\mathbf{y})=-\mathbf{y}^{\mathrm{T}}\ln \mathrm{h}_{\mathbf{w}}(\mathbf{x})-(1-\mathbf{y})^{\mathrm{T}}\ln[1-\mathrm{h}_{\mathbf{w}}(\mathbf{x})] \tag{1.10}$$

注意：根据损失函数的定义，此处需要计算 $\mathbf{y}$ 与 $\mathrm{h}_{\mathbf{w}}(\mathbf{x})$ 的内积(损失函数的值是标量)，因此 $\mathbf{y}$ 使用转置形式。

# 损失函数对权重求导

为了使用梯度下降方法更新权重，需要计算损失函数(简记为 $L$ )对 $\mathbf{w}_{jk}^{(i)}$ 的偏导数:

$$\frac{\partial L}{\partial w_{jk}^{(i)}}=\sum_{p\in(1,s_{i+1})}\frac{\partial L}{\partial z_p^{(i+1)}}\frac{\partial z_p^{(i+1)}}{\partial w_{jk}^{(i)}}=\frac{\partial L}{\partial z_j^{(i+1)}}\frac{\partial z_j^{(i+1)}}{\partial w_{jk}^{(i)}}=\frac{\partial L}{\partial z_j^{(i+1)}}\frac{\partial(w_{jk}^{(i)}a_k^{(i)})}{\partial w_{jk}^{(i)}}=\frac{\partial L}{\partial z_j^{(i+1)}}a_k^{(i)} \tag{1.11}$$

(注意 $L$ 是 $z_j^{(i+1)}$ 的函数，而 $z_j^{(i+1)}$ 是 $\mathbf{w}_{jk}^{(i)}$ 的函数。而当 $p\neq j$ 时，$z_p^{(i+1)}$ 不是 $\mathbf{w}_{jk}^{(i)}$ 的函数，因为 $\mathbf{w}_{jk}^{(i)}$ 不影响 $z_p^{(i+1)}$ 。)

定义误差因子 $\boldsymbol{\delta}^{(i)}$ 如下:

$$\boldsymbol{\delta}^{(i)}=[\delta_1^{(i)},\cdots,\delta_{s_i}^{(i)}]^{\mathrm{T}}=\frac{\partial L}{\partial \mathbf{z}^{(i)}}=[\frac{\partial L}{\partial z_1^{(i)}},\cdots,\frac{\partial L}{\partial z_{s_i}^{(i)}}]^{\mathrm{T}}\in\mathbb{R}(s_i\times 1) \tag{1.12}$$

其中:

$$\delta_j^{(i)}=\frac{\partial L}{\partial z_j^{(i)}} \tag{1.13}$$

则有

$$\frac{\partial L}{\partial w_{jk}^{(i)}}=\frac{\partial L}{\partial z_j^{(i+1)}}a_k^{(i)}=\delta_j^{(i+1)}a_k^{(i)} \tag{1.14}$$

由上式求出的矩阵元素倒推出矩阵公式如下:

$$\frac{\partial L}{\partial \mathbf{W}^{(i)}}=\begin{bmatrix}\frac{\partial L}{\partial w_{11}^{(i)}} & \cdots & \frac{\partial L}{\partial w_{1s_i}^{(i)}}\\ \vdots & \ddots & \vdots \\ \frac{\partial L}{\partial w_{s_{i+1}1}^{(i)}} & \cdots & \frac{\partial L}{\partial w_{s_{i+1}s_i}^{(i)}}\end{bmatrix}=\begin{bmatrix}\delta_1^{(i+1)}a_1^{(i)} & \cdots & \delta_1^{(i+1)}a_{s_i}^{(i)}\\ \vdots & \ddots & \vdots \\ \delta_{s_{i+1}}^{(i+1)}a_1^{(i)} & \cdots & \delta_{s_{i+1}}^{(i+1)}a_{s_i}^{(i)}\end{bmatrix}=\boldsymbol{\delta}^{(i+1)}[\mathbf{a}^{(i)}]^{\mathrm{T}}\in\mathbb{R}(s_{i+1}\times 1)(1\times s_i)=\mathbb{R}(s_{i+1}\times s_i) \tag{1.15}$$

# 计算误差因子

$$\delta_j^{(i)}=\frac{\partial\,\mathrm{Loss}}{\partial z_j^{(i)}}=\sum_k\frac{\partial\,\mathrm{Loss}}{\partial z_k^{(i+1)}}\frac{\partial z_k^{(i+1)}}{\partial z_j^{(i)}}=\sum_k\delta_k^{(i+1)}\frac{\partial z_k^{(i+1)}}{\partial z_j^{(i)}} \tag{1.16}$$

其中: $\frac{\partial z_k^{(i+1)}}{\partial z_j^{(i)}}=\frac{\partial\sum_p w_{kp}^{(i)}a_p^{(i)}}{\partial z_j^{(i)}}=\sum_p\frac{\partial w_{kp}^{(i)}a_p^{(i)}}{\partial z_j^{(i)}}=\frac{\partial w_{kj}^{(i)}a_j^{(i)}}{\partial z_j^{(i)}}=w_{kj}^{(i)}\frac{\partial f(z_j^{(i)})}{\partial z_j^{(i)}}=w_{kj}^{(i)}f'(z_j^{(i)})$, f 为 sigmoid 激活函数。

因此

$$\begin{aligned}\delta_j^{(i)}&=\sum_k\delta_k^{(i+1)}\frac{\partial z_k^{(i+1)}}{\partial z_j^{(i)}}=\sum_k\delta_k^{(i+1)}w_{kj}^{(i)}f'(z_j^{(i)})\\&=f'(z_j^{(i)})\sum_k\delta_k^{(i+1)}w_{kj}^{(i)}=f'(z_j^{(i)})[(\boldsymbol{\delta}^{(i+1)})^{\mathrm{T}}\mathbf{w}_{\cdot j}^{(i)}]\end{aligned} \tag{1.17}$$

矩阵形式如下:

$$\boldsymbol{\delta}^{(i)}=\begin{bmatrix}f'(z_1^{(i)})[(\boldsymbol{\delta}^{(i+1)})^{\mathrm{T}}\mathbf{w}_{\cdot 1}^{(i)}]\\ \vdots \\ f'(z_{s_i}^{(i)})[(\boldsymbol{\delta}^{(i+1)})^{\mathrm{T}}\mathbf{w}_{\cdot s_i}^{(i)}]\end{bmatrix}=f'(\mathbf{z}^{(i)})\circ[(\boldsymbol{\delta}^{(i+1)})^{\mathrm{T}}\mathbf{W}^{(i)}]^{\mathrm{T}} \tag{1.18}$$

由(1.9)可知 $f'(\mathbf{z}^{(i)})=f(\mathbf{z}^{(i)})-f^2(\mathbf{z}^{(i)})$, 因此上式

$$\boldsymbol{\delta}^{(i)}=[f(\mathbf{z}^{(i)})-f^2(\mathbf{z}^{(i)})]\circ[(\boldsymbol{\delta}^{(i+1)})^{\mathrm{T}}\mathbf{W}^{(i)}]^{\mathrm{T}} \tag{1.19}$$

其中∘表示两个同维度向量对应元素相乘, 得到一个同维度的新向量, 例如

$$[a,b,c]\circ[d,e,f]=[ad,be,cf] \tag{1.20}$$

因为: $\mathrm{Loss}(\mathbf{z}^{(n)},\mathbf{y})=-\mathbf{y}^{\mathrm{T}}\ln\mathrm{sigmoid}\,\mathbf{z}^{(n)}-(1-\mathbf{y})^{\mathrm{T}}\ln(1-\mathrm{sigmoid}\,\mathbf{z}^{(n)})$, 有

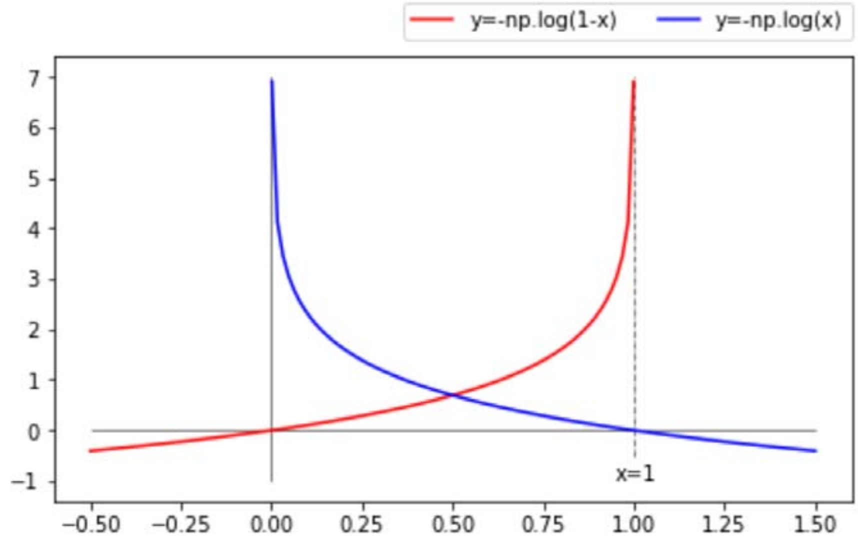
$$\begin{aligned}\frac{d\,\mathrm{Loss}}{dz_i^{(n)}}&=\frac{\sum_k -y_k\ln(\mathrm{sigmoid}\,z_k^{(n)})-\sum_k(1-y_k)\ln(1-\mathrm{sigmoid}\,z_k^{(n)})}{dz_i^{(n)}}\\&=\frac{-y_i\ln\ \mathrm{sigmoid}\,z_i^{(n)}-(1-y_i)\ln(1-\mathrm{sigmoid}\,z_i^{(n)})}{dz_i^{(n)}}\\&=-y_i(1-\mathrm{sigmoid}\,z_i^{(n)})+(1-y_i)\,\mathrm{sigmoid}\,z_i^{(n)}\end{aligned} \tag{1.21}$$

因此由元素推导出相应矩阵公式如下:

$$\boldsymbol{\delta}^{(n)}=\frac{d\,\mathrm{Loss}}{d\mathbf{z}^{(n)}}=-\mathbf{y}\circ(1-\mathrm{sigmoid}\,\mathbf{z}^{(n)})+(1-\mathbf{y})\circ\mathrm{sigmoid}\,\mathbf{z}^{(n)} \tag{1.22}$$

# 交叉熵损失函数原理

$$Loss = -[y\ln h_w(\mathbf{x}) + (1-y)\ln(1-h_w(\mathbf{x})] = \begin{cases} -\ln(1-h_w(\mathbf{x})) & y=0 \\ -\ln(h_w(\mathbf{x}) & y=1 \end{cases}$$




图表 2：交叉熵损失函数原理图(当 $y=0$ 时，$h_w(\mathbf{x})$ 如果趋于 1，或者当 y=1 时，$h_w(\mathbf{x})$ 如果趋于 0，则 $Loss$ 为无限大)