导数、反向传播与向量化

Justin Johnson

2017 年 9 月 6 日

1 导数

1.1 标量情形

你大概已经熟悉了标量情形下导数的概念：给定一个函数 $f:\mathbb{R}\to\mathbb{R}$，$f$ 在点 $x\in\mathbb{R}$ 处的导数定义为

$$f'(x)=\lim_{h\to 0}\frac{f(x+h)-f(x)}{h}.$$

导数是刻画变化的一种方式。在标量情形中，函数 f 在点 x 处的导数告诉我们：当输入 x 发生一个很小的变化 ε 时，函数值 f 会变化多少：

$$f(x+\varepsilon)\approx f(x)+\varepsilon f'(x).$$

为了记号简洁，我们通常会给 f 的输出起一个名字。设 $y=f(x)$，并将 y 关于 x 的导数写作 $\frac{\partial y}{\partial x}$。这个记号强调 $\frac{\partial y}{\partial x}$ 是变量 x 与 y 之间的变化率；具体地说，如果 x 改变了 ε，那么 y 大约会改变 $\varepsilon\frac{\partial y}{\partial x}$。

我们可以把这种关系写成

$$x\to x+\Delta x\Longrightarrow y\approx y+\frac{\partial y}{\partial x}\Delta x.$$

这里的意思是：“把 x 从 x 改成 $x+\Delta x$，意味着 y 大约会变成 $y+\Delta x\frac{\partial y}{\partial x}$。”这种记号并不是标准写法，但它很直观，因为它把 x 的变化与 y 的变化之间的关系直接呈现了出来。

链式法则告诉我们如何计算复合函数的导数。在标量情形中，设 $f,g:\mathbb{R}\to\mathbb{R}$，并令 $y=f(x)$、$z=g(y)$；那么也可以写作 $z=(g\circ f)(x)$，或者画成下面的计算图：

$$x\xrightarrow{f}y\xrightarrow{g}z.$$

标量形式的链式法则告诉我们

$$\frac{\partial z}{\partial x}=\frac{\partial z}{\partial y}\frac{\partial y}{\partial x}.$$

这个式子在直觉上也很合理。导数 $\frac{\partial y}{\partial x}$ 和 $\frac{\partial z}{\partial y}$ 分别给出

$$x\to x+\Delta x\Longrightarrow y\approx y+\frac{\partial y}{\partial x}\Delta x,$$

$$y \to y + \Delta y \Longrightarrow z \approx z + \frac{\partial z}{\partial y}\Delta y.$$

把这两条规则组合起来，就能得到 x 对 z 的影响：如果 x 变化了 Δx，那么 y 会变化 $\frac{\partial y}{\partial x}\Delta x$，即

$$\Delta y = \frac{\partial y}{\partial x}\Delta x.$$

如果 y 再变化 Δy，那么 z 就会变化

$$\frac{\partial z}{\partial y}\Delta y = \frac{\partial z}{\partial y}\frac{\partial y}{\partial x}\Delta x,$$

这正是链式法则所表达的内容。

1.2 梯度：向量输入，标量输出

同样的直觉也适用于向量情形。现在设 $f : \mathbb{R}^N \to \mathbb{R}$，它接收一个向量作为输入并输出一个标量。此时，$f$ 在点 $x \in \mathbb{R}^N$ 处的导数称为**梯度**，定义为

$$\nabla_x f(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{\|h\|}.$$

此时梯度 $\nabla_x f(x) \in \mathbb{R}^N$ 是一个向量，但直觉与标量情形相同。若仍设 $y = f(x)$，那么有

$$x \to x + \Delta x \Longrightarrow y \approx y + \frac{\partial y}{\partial x} \cdot \Delta x.$$

相较于标量情形，这个公式稍有变化，因为现在 x、Δx 和 $\frac{\partial y}{\partial x}$ 都是 $\mathbb{R}^N$ 中的向量，而 y 是标量。特别地，把 $\frac{\partial y}{\partial x}$ 与 Δx 相乘时，我们使用点积，它把两个向量合成为一个标量。

这个公式的一个好处是，它赋予了梯度各个分量明确的意义。假设 Δx 是第 i 个基向量，也就是只有第 i 个坐标为 1，其余坐标都为 0。那么点积 $\frac{\partial y}{\partial x} \cdot \Delta x$ 恰好就是 $\frac{\partial y}{\partial x}$ 的第 i 个坐标。因此，梯度的第 i 个分量刻画了：如果沿着第 i 个坐标轴移动 x，那么 y 将近似变化多少。

这也意味着，我们可以把梯度看成由偏导数组成的向量：

$$\frac{\partial y}{\partial x} = \left(\frac{\partial y}{\partial x_1}, \frac{\partial y}{\partial x_2}, \ldots, \frac{\partial y}{\partial x_N} \right),$$

其中 x_i 是向量 x 的第 i 个坐标，它是一个标量，因此每个偏导数 $\frac{\partial y}{\partial x_i}$ 也都是标量。

1.3 雅可比矩阵：向量输入，向量输出

现在设 $f : \mathbb{R}^N \to \mathbb{R}^M$，它接收一个向量作为输入并产生一个向量作为输出。此时，f 在点 x 处的导数也称为**雅可比矩阵（Jacobian）**，它是一个 $M \times N$ 的偏导数矩阵。若仍令 $y = f(x)$，那么可以写成

$$\frac{\partial y}{\partial x} = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_M}{\partial x_1} & \cdots & \frac{\partial y_M}{\partial x_N} \end{pmatrix}.$$

雅可比矩阵描述了 x 的每个元素与 y 的每个元素之间的关系：$\frac{\partial y}{\partial x}$ 的第 (i,j) 个元素等于 $\frac{\partial y_i}{\partial x_j}$，因此它告诉我们当 x_j 发生一个小变化时，y_i 会变化多少。

和前面的情形一样，雅可比矩阵给出了输入变化与输出变化之间的关系：

$$x \to x + \Delta x \Longrightarrow y \approx y + \frac{\partial y}{\partial x}\Delta x.$$

这里 $\frac{\partial y}{\partial x}$ 是一个 $M \times N$ 矩阵，而 Δx 是一个 N 维向量，因此 $\frac{\partial y}{\partial x}\Delta x$ 是一个矩阵与向量的乘法，其结果是一个 M 维向量。

链式法则可以自然地扩展到向量情形。设 $f : \mathbb{R}^N \to \mathbb{R}^M$、$g : \mathbb{R}^M \to \mathbb{R}^K$。令 $x \in \mathbb{R}^N$、$y \in \mathbb{R}^M$、$z \in \mathbb{R}^K$，并满足 $y = f(x)$、$z = g(y)$，那么计算图仍然是

$$x \xrightarrow{f} y \xrightarrow{g} z.$$

链式法则的形式与标量情形相同：

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y}\frac{\partial y}{\partial x}.$$

不过现在每一项都是矩阵：$\frac{\partial z}{\partial y}$ 是一个 $K \times M$ 矩阵，$\frac{\partial y}{\partial x}$ 是一个 $M \times N$ 矩阵，而 $\frac{\partial z}{\partial x}$ 是一个 $K \times N$ 矩阵；这里的乘法就是普通的矩阵乘法。

1.4 广义雅可比：张量输入，张量输出

向量是一维数表，矩阵是二维数表，而张量则是一个 D 维数值网格[1]。

深度学习中的很多运算都以张量为输入，并以张量为输出。例如，一张图像通常被表示为一个三维数值网格，这三个维度分别对应图像的高度、宽度以及颜色通道（红、绿、蓝）。因此，我们需要一种能够兼容一般张量函数的导数定义。

现在设

$$f : \mathbb{R}^{N_1 \times \cdots \times N_{D_x}} \to \mathbb{R}^{M_1 \times \cdots \times M_{D_y}}.$$

于是，f 的输入是一个形状为 $N_1 \times \cdots \times N_{D_x}$ 的 D_x 维张量，输出是一个形状为 $M_1 \times \cdots \times M_{D_y}$ 的 D_y 维张量。若 $y = f(x)$，那么导数 $\frac{\partial y}{\partial x}$ 是一个**广义雅可比**，其形状为

$$(M_1 \times \cdots \times M_{D_y}) \times (N_1 \times \cdots \times N_{D_x}).$$

注意，我们把 $\frac{\partial y}{\partial x}$ 的维度分成了两组：前一组与 y 的维度一致，后一组与 x 的维度一致。按这种分组方式，我们可以把广义雅可比看成矩阵的推广，其中每一“行”的形状与 y 相同，每一“列”的形状与 x 相同。

若令 $i \in \mathbb{Z}^{D_y}$、$j \in \mathbb{Z}^{D_x}$ 是整数索引向量，那么可以写作

$$\left(\frac{\partial y}{\partial x}\right)_{i,j} = \frac{\partial y_i}{\partial x_j}.$$

[1] “tensor” 这个词在不同领域中有不同用法；你可能在物理学或抽象代数里见过它。机器学习中把张量定义为一个 D 维数值网格，这一定义与其他领域中的张量定义密切相关。

在这个等式里，y_i 和 x_j 都是标量，因此 $\frac{\partial y_i}{\partial x_j}$ 也是标量。用这个记号可以看到，和普通雅可比一样，广义雅可比描述了 x 的所有元素与 y 的所有元素之间的相对变化率。

广义雅可比与之前一样，也给出输入变化和输出变化之间的关系：

$$x \to x + \Delta x \Longrightarrow y \approx y + \frac{\partial y}{\partial x}\Delta x.$$

区别在于，此时 Δx 是一个形状为 $N_1 \times \cdots \times N_{D_x}$ 的张量，而 $\frac{\partial y}{\partial x}$ 是一个形状为

$$(M_1 \times \cdots \times M_{D_y}) \times (N_1 \times \cdots \times N_{D_x})$$

的广义矩阵。因此，$\frac{\partial y}{\partial x}\Delta x$ 是一次广义的 “矩阵-向量” 乘法，其结果是一个形状为 $M_1 \times \cdots \times M_{D_y}$ 的张量。

这种广义矩阵-向量乘法遵循与传统矩阵-向量乘法相同的代数规则：

$$\left(\frac{\partial y}{\partial x}\Delta x\right)_i = \sum_j \left(\frac{\partial y}{\partial x}\right)_{i,j} (\Delta x)_j = \left(\frac{\partial y}{\partial x}\right)_{i,:} \cdot \Delta x.$$

唯一的不同在于，这里的索引 i 和 j 不再是标量，而是索引向量。在上式中，$\left(\frac{\partial y}{\partial x}\right)_{i,:}$ 表示广义矩阵 $\frac{\partial y}{\partial x}$ 的第 i “行”，它本身是一个与 x 具有相同形状的张量。我们还约定：两个同形张量之间的点积，就是逐元素相乘后再求和，这与向量的点积完全一致。

对于张量值函数，链式法则的形式也保持不变。设 $y = f(x)$、$z = g(y)$，其中 x 和 y 的形状与上面相同，而 z 的形状为 $K_1 \times \cdots \times K_{D_z}$。那么链式法则仍然写作

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y}\frac{\partial y}{\partial x}.$$

不同之处在于，现在 $\frac{\partial z}{\partial y}$ 是一个形状为

$$(K_1 \times \cdots \times K_{D_z}) \times (M_1 \times \cdots \times M_{D_y})$$

的广义矩阵，而 $\frac{\partial y}{\partial x}$ 是一个形状为

$$(M_1 \times \cdots \times M_{D_y}) \times (N_1 \times \cdots \times N_{D_x})$$

的广义矩阵；它们的乘积 $\frac{\partial z}{\partial y}\frac{\partial y}{\partial x}$ 是一次广义的矩阵-矩阵乘法，结果形状为

$$(K_1 \times \cdots \times K_{D_z}) \times (N_1 \times \cdots \times N_{D_x}).$$

和上面定义的广义矩阵-向量乘法一样，广义矩阵-矩阵乘法也遵循传统矩阵-矩阵乘法相同的代数规则：

$$\left(\frac{\partial z}{\partial x}\right)_{i,j} = \sum_k \left(\frac{\partial z}{\partial y}\right)_{i,k} \left(\frac{\partial y}{\partial x}\right)_{k,j} = \left(\frac{\partial z}{\partial y}\right)_{i,:} \cdot \left(\frac{\partial y}{\partial x}\right)_{:,j}.$$

2 张量上的反向传播

在神经网络的语境下，一层 f 通常是输入张量 x 和权重 w 的函数；该层的输出张量为 $y = f(x, w)$。这一层通常被嵌入在某个大型神经网络中，网络具有一个标量损失 L。

在反向传播过程中，我们假设已经知道 $\frac{\partial L}{\partial y}$，目标是计算 $\frac{\partial L}{\partial x}$ 和 $\frac{\partial L}{\partial w}$。根据链式法则，有

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial x}, \qquad \frac{\partial L}{\partial w} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial w}.$$

因此，一种直接的做法是构造出广义雅可比 $\frac{\partial y}{\partial x}$ 和 $\frac{\partial y}{\partial w}$，然后用广义矩阵乘法来计算 $\frac{\partial L}{\partial x}$ 和 $\frac{\partial L}{\partial w}$。

然而，这种做法有一个问题：雅可比矩阵 $\frac{\partial y}{\partial x}$ 和 $\frac{\partial y}{\partial w}$ 往往大得无法放进内存。

举一个具体例子。假设 f 是一个线性层，它接收一个由 N 个向量组成的小批量输入，每个向量的维度是 D，并输出一个由 N 个向量组成的小批量结果，每个向量的维度是 M。那么，x 是一个形状为 $N \times D$ 的矩阵，w 是一个形状为 $D \times M$ 的矩阵，而 $y = f(x, w) = xw$ 是一个形状为 $N \times M$ 的矩阵。

此时雅可比 $\frac{\partial y}{\partial x}$ 的形状为 $(N \times M) \times (N \times D)$。在一个典型的神经网络里，我们可能有 $N = 64$ 且 $M = D = 4096$；那么 $\frac{\partial y}{\partial x}$ 一共包含

$$64 \cdot 4096 \cdot 64 \cdot 4096$$

个标量值，超过 680 亿个数。若使用 32 位浮点数来存储，这个雅可比矩阵需要 256 GB 内存。因此，显式存储并操作雅可比矩阵几乎完全不可行。

不过事实证明，对于大多数常见的神经网络层，我们可以直接推导出计算乘积

$$\frac{\partial L}{\partial y}\frac{\partial y}{\partial x}$$

的表达式，而无须显式构造雅可比矩阵 $\frac{\partial y}{\partial x}$。更进一步，通常我们甚至不需要先写出雅可比矩阵的显式表达式；很多情况下，只需要先在纸上推导一个很小的例子，再据此归纳出一般公式。

下面以线性层 $f(x, w) = xw$ 为例说明这一点。设 $N = 1$、$D = 2$、$M = 3$。则可以显式写成

$$y = xw = \begin{pmatrix} x_{1,1} & x_{1,2} \end{pmatrix} \begin{pmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \end{pmatrix}$$

$$= \begin{pmatrix} x_{1,1}w_{1,1} + x_{1,2}w_{2,1} & x_{1,1}w_{1,2} + x_{1,2}w_{2,2} & x_{1,1}w_{1,3} + x_{1,2}w_{2,3} \end{pmatrix}.$$

在反向传播中，我们假设已经得到 $\frac{\partial L}{\partial y}$。严格来说，它的形状是 $(1) \times (N \times M)$；但为了书写方便，我们把它看成一个形状为 $N \times M$ 的矩阵。于是这里可以写成

$$\frac{\partial L}{\partial y} = \begin{pmatrix} dy_{1,1} & dy_{1,2} & dy_{1,3} \end{pmatrix}.$$

现在我们的目标是用 x、w 和 $\frac{\partial L}{\partial y}$ 来推导 $\frac{\partial L}{\partial x}$ 的表达式，而不显式构造完整的雅可比矩阵 $\frac{\partial y}{\partial x}$。我们知道 $\frac{\partial L}{\partial x}$ 的形状是 $(1) \times (N \times D)$；但按照梯度的常见表示方式，我们把它视为一个形状为 $N \times D$ 的矩阵。也就是说，

$$\frac{\partial L}{\partial x} = \begin{pmatrix} \frac{\partial L}{\partial x_{1,1}} & \frac{\partial L}{\partial x_{1,2}} \end{pmatrix}.$$

逐个元素来考虑，链式法则给出

$$\frac{\partial L}{\partial x_{1,1}} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial x_{1,1}}, \qquad \frac{\partial L}{\partial x_{1,2}} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial x_{1,2}}.$$

现在计算

$$\frac{\partial y}{\partial x_{1,1}} = \begin{pmatrix} \frac{\partial y_{1,1}}{\partial x_{1,1}} & \frac{\partial y_{1,2}}{\partial x_{1,1}} & \frac{\partial y_{1,3}}{\partial x_{1,1}} \end{pmatrix} = \begin{pmatrix} w_{1,1} & w_{1,2} & w_{1,3} \end{pmatrix},$$

$$\frac{\partial y}{\partial x_{1,2}} = \begin{pmatrix} \frac{\partial y_{1,1}}{\partial x_{1,2}} & \frac{\partial y_{1,2}}{\partial x_{1,2}} & \frac{\partial y_{1,3}}{\partial x_{1,2}} \end{pmatrix} = \begin{pmatrix} w_{2,1} & w_{2,2} & w_{2,3} \end{pmatrix}.$$

于是有

$$\frac{\partial L}{\partial x_{1,1}} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial x_{1,1}} = dy_{1,1}w_{1,1} + dy_{1,2}w_{1,2} + dy_{1,3}w_{1,3},$$

$$\frac{\partial L}{\partial x_{1,2}} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial x_{1,2}} = dy_{1,1}w_{2,1} + dy_{1,2}w_{2,2} + dy_{1,3}w_{2,3}.$$

把它们合在一起，就得到

$$\frac{\partial L}{\partial x} = \begin{pmatrix} \frac{\partial L}{\partial x_{1,1}} & \frac{\partial L}{\partial x_{1,2}} \end{pmatrix}$$

$$= \begin{pmatrix} dy_{1,1}w_{1,1} + dy_{1,2}w_{1,2} + dy_{1,3}w_{1,3} & dy_{1,1}w_{2,1} + dy_{1,2}w_{2,2} + dy_{1,3}w_{2,3} \end{pmatrix}$$

$$= \frac{\partial L}{\partial y} w^T.$$

这个最终结果非常有意思，因为它让我们能够高效地计算 $\frac{\partial L}{\partial x}$，而无须显式构造雅可比矩阵 $\frac{\partial y}{\partial x}$。尽管这里我们只针对 $N = 1$、$D = 2$、$M = 3$ 的特例完成了推导，但这个结论实际上在一般情形下同样成立。

通过类似的思路，我们也可以推导出 $\frac{\partial L}{\partial w}$ 的一个对应表达式，而不需要显式构造雅可比矩阵 $\frac{\partial y}{\partial w}$。你可以把这部分当作练习自行完成。