Tutorial: Multivariate Differentiation

ECE421 – Introduction To Machine Learning (Fall 2022)

Stephan Rabanser University of Toronto & Vector Institute for AI stephan@cs.toronto.edu

1 The Basics: Univariate Differentiation

In machine learning we are often concerned with assessing the rate of change of a function's output(s) with respect to its input(s). In the simple case where said function maps a scalar value to another scalar value, i.e. $f : \mathbb{R} \to \mathbb{R}$, this concept is captured by univariate differentiation.

Formally, we define the derivative of a function f at point x as the limit of a difference quotient:

$$f'(x) = \frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \tag{1}$$

Note that for some functions this limit might not exist for specific choices of x. In that case, the function is called non-differentiable (at x). Conversely, we call a function differentiable if the limit exists for all choices of x in the domain of f.

Informally, the derivative of a function $\frac{df(x)}{dx}$ at a particular input value x measures the slope of the tangent line to the function f at x. Close to x, the derivative $\frac{df(x)}{dx}$ can also be thought of as a linear approximation of f.

1.1 Frequently Used Derivative Rules

The derivative of a function can be computed from the definition by evaluating the difference quotient and computing its limit. In practice, however, it is usually easier to consider a function as a composition of simpler function for which differentiation rules are already readily available. We briefly revise some of the most useful differentiation rules below.

Derivative of Powers If $f(x) = x^a$, then

$$(x^a)' = ax^{a-1}. \tag{2}$$

Derivatives of Popular Functions For the exponential function, the power function, and the logarithmic function, the following differentiation rules apply:

$$(e^x)' = e^x \qquad (a^x)' = a^x \ln(x) \quad \text{for } a > 0 \qquad (\ln(x))' = \frac{1}{x} \quad \text{for } x > 0 \tag{3}$$

Constant Rule If $f(x)$ is constant (i.e., if the there is no functional dependence on x in f), then

$$f'(x) = 0. \tag{4}$$

Sum Rule For all differentiable functions $f : \mathbb{R} \to \mathbb{R}$, $g : \mathbb{R} \to \mathbb{R}$ and scalars $a, b \in \mathbb{R}$:

$$(af(x) + bg(x))' = af'(x) + bg'(x). \tag{5}$$

Product Rule For all differentiable functions $f : \mathbb{R} \to \mathbb{R}$, $g : \mathbb{R} \to \mathbb{R}$:

$$(f(x)g(x))' = f'(x)g(x) + f(x)g'(x). \tag{6}$$

For $f(x) = a$ we have:

$$(ag(x))' = ag'(x). \tag{7}$$

Quotient Rule For all differentiable functions $f : \mathbb{R} \to \mathbb{R}$, $g : \mathbb{R} \to \mathbb{R}$ as long as $g(x) \neq 0$ for any x:

$$\left(\frac{f(x)}{g(x)}\right)' = \frac{f'(x)g(x) - f(x)g'(x)}{g(x)^2} \tag{8}$$

Chain Rule For a differentiable composite function $f(x) = h(g(x))$:

$$(h(g(x)))' = h'(g(x))g'(x) \tag{9}$$

1.2 Identifying Critical Points (Minima, Maxima, Inflection Points)

Critical Points Differentiation techniques are often discussed in the context of curve sketching. In particular, differentiation allows for the identification of critical points such as minima, maxima, and inflection points. Critical points are defined as points where the derivative is 0 (or non-existent).

$$f'(x) \stackrel{?}{=} 0 \implies \text{critical point} \tag{10}$$

Minima, Maxima, and Saddle Points While the first derivative allows us to identify the existence of a critical point, the second derivative allows us to identify whether a critical point is a maximum, a minimum, or a saddle point. At a *maximum* all curvature directions point downwards (concave down); at a *minimum* all curvature directions point upwards (concave up); and at a *inflection point* the curvature points in different directions.

$$f''(x) \stackrel{?}{>} 0 \implies \text{minimum} \qquad f''(x) \stackrel{?}{<} 0 \implies \text{maximum} \qquad f''(x) \stackrel{?}{=} 0 \implies \text{inflection point} \tag{11}$$

Example Usage in Machine Learning

Machine learning often seeks to determine the minimum of a *cost/loss function*. As such, iterative techniques pioneered in numerical computing find widespread usage in various ML algorithms. At its core, most such methods aim at finding critical points where the derivative is 0. Differentiation tools are also used in probabilistic parameter estimation, most notably in *maximum likelihood estimation*, to identify the set of most likely parameters for a data-generating probability distribution.

2 Moving to Higher Dimensions: Multivariate Calculus

We can generalize the concepts from univariate differentiation to higher dimensions by studying multivariate (or multivariable) differentiation.

2.1 Partial derivatives

Assume that we are now dealing with a function $f : \mathbb{R}^n \to \mathbb{R}$, i.e. a function that takes as input a vector $\boldsymbol{x} = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^\top$ and produces a scalar output. We define the partial derivative of f with respect to one of its dimensions x_i as:

$$\frac{\partial}{\partial x_i} f(\boldsymbol{x}) = \frac{\partial}{\partial x_i} f(x_1, \ldots, x_n) = \lim_{h \to 0} \frac{f(x_1, \ldots, x_i + h, \ldots, x_n) - f(x_1, \ldots, x_i, \ldots, x_n)}{h} \tag{12}$$

Note that partial derivatives are denoted using the ∂ ("partial", "del") symbol. This notation conveniently allows us to identify whether f acts on vector-valued inputs. Further note that this definition implies that we only consider the direction x_i as variable and treat all other directions as constant. Hence, partial differentiation can be thought of as univariate differentiation in a particularly chosen dimension.

Gradient Vector We can compute the partial derivative for all dimensions $x_i \in \boldsymbol{x}$ and collect all partial derivatives in a *gradient vector:*

$$\nabla f(x_1, \ldots, x_n) = \begin{bmatrix} \frac{\partial}{\partial x_1} f(x_1, \ldots, x_n) & \cdots & \frac{\partial}{\partial x_n} f(x_1, \ldots, x_n) \end{bmatrix} \tag{13}$$

Example

Consider the function $f(x, y) = x^2 y + 3 \ln(xy)$. Then the partial derivatives and the gradient are computed as follows:

$$\begin{aligned} \frac{\partial}{\partial x} f(x, y) &= 2xy + 3\frac{1}{xy} y \\ &= 2xy + \frac{3}{x} \end{aligned} \tag{14}$$

$$\begin{aligned} \frac{\partial}{\partial y} f(x, y) &= x^2 + 3\frac{1}{xy} x \\ &= x^2 + \frac{3}{y} \end{aligned} \tag{15}$$

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial}{\partial x} f(x, y) & \frac{\partial}{\partial y} f(x, y) \end{bmatrix} = \begin{bmatrix} 2xy + \frac{3}{x} & x^2 + \frac{3}{y} \end{bmatrix} \tag{16}$$

Jacobian Matrix For a differentiable vector valued function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$

$$\boldsymbol{J} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial \boldsymbol{f}}{\partial x_1} & \cdots & \frac{\partial \boldsymbol{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \nabla f_1^\top \\ \vdots \\ \nabla f_m^\top \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1(\boldsymbol{x})}{\partial x_1} & \frac{\partial f_1(\boldsymbol{x})}{\partial x_2} & \cdots & \frac{\partial f_1(\boldsymbol{x})}{\partial x_n} \\ \frac{\partial f_2(\boldsymbol{x})}{\partial x_1} & \frac{\partial f_2(\boldsymbol{x})}{\partial x_2} & \cdots & \frac{\partial f_2(\boldsymbol{x})}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m(\boldsymbol{x})}{\partial x_1} & \frac{\partial f_m(\boldsymbol{x})}{\partial x_2} & \cdots & \frac{\partial f_m(\boldsymbol{x})}{\partial x_n} \end{bmatrix} \tag{17}$$

Example Usage in Machine Learning

Both the gradient and the Jacobian belong to the bread and butter of both classical and modern learning / numerical approximation algorithms. In particular, they are the core ingredients of an algorithm called *gradient descent* which enables us to iteratively search for a local minimum of a differentiable function. For instance, neural network learning relies on gradient descent via the *backpropagation* algorithm.

Example

Consider the multivariate function $\boldsymbol{f}\left(\begin{bmatrix}x\\y\end{bmatrix}\right) = \begin{bmatrix}f_1(x,y)\\f_2(x,y)\end{bmatrix} = \begin{bmatrix}x^2y\\5x+\sin(y)\end{bmatrix}$. Then, the partial derivatives are given by:

$$\begin{aligned}\frac{\partial}{\partial x}f_1(x,y) &= 2xy\\ \frac{\partial}{\partial x}f_2(x,y) &= 5\\ \frac{\partial}{\partial y}f_1(x,y) &= x^2\\ \frac{\partial}{\partial y}f_2(x,y) &= \cos(y)\end{aligned} \tag{18}$$

We collect all partial derivates in the Jacobian matrix:

$$\boldsymbol{J} = \begin{bmatrix}\frac{\partial f_1(x,y)}{\partial x} & \frac{\partial f_1(x,y)}{\partial y}\\ \frac{\partial f_2(x,y)}{\partial x} & \frac{\partial f_2(x,y)}{\partial y}\end{bmatrix} = \begin{bmatrix}2xy & x^2\\ 5 & \cos(y)\end{bmatrix} \tag{19}$$

Hessian Matrix For a twice-differentiable scalar-valued function $f : \mathbb{R}^n \to \mathbb{R}$ the Hessian matrix $\boldsymbol{H}$ is defined as the matrix containing the combinations of all second-order derivatives:

$$\boldsymbol{H} = \begin{bmatrix}\frac{\partial^2 f(\boldsymbol{x})}{\partial x_1^2} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 x_n}\\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2 x_1} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2^2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2 x_n}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n x_1} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n x_2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n^2}\end{bmatrix} \tag{20}$$

Note that $\partial x_i^2 = \partial x_i \partial x_i$.

Example Usage in Machine Learning

The Hessian matrix and its approximations are frequently used to assess the curvature of loss landscapes in neural networks. Similar to the uni-variate second-order derivative of a function, the Hessian enables us to identify saddle points and directions of higher and lower curvature. In particular, the ratio between the largest and the lowest eigenvalue of $\boldsymbol{H}$, i.e. $\frac{\lambda_{\max}}{\lambda_{\min}}$, defines the *condition number*. A large condition number implies slower convergence while a condition number of 1 enables gradient descent to quickly converge in all curvature directions.

Example

Consider the function $f(x, y) = x^3 - 2xy - y^6$. Then the Hessian is computed as follows. The first partial derivatives are given by:

$$\begin{aligned} \frac{\partial}{\partial x} f(x, y) &= 3x^2 - 2y \\ \frac{\partial}{\partial y} f(x, y) &= -2x - 6y^5 \end{aligned} \tag{21}$$

The second partial derivates are given by:

$$\begin{aligned} \frac{\partial}{\partial x \partial x} f(x, y) &= 6x \\ \frac{\partial}{\partial y \partial x} f(x, y) = \frac{\partial}{\partial x \partial y} f(x, y) &= -2 \\ \frac{\partial}{\partial y \partial y} f(x, y) &= -30y^4 \end{aligned} \tag{22}$$

We collect all values in the Hessian matrix:

$$\boldsymbol{H} = \begin{bmatrix} \frac{\partial}{\partial x \partial x} f(x, y) & \frac{\partial}{\partial x \partial y} f(x, y) \\ \frac{\partial}{\partial y \partial x} f(x, y) & \frac{\partial}{\partial y \partial y} f(x, y) \end{bmatrix} = \begin{bmatrix} 6x & -2 \\ -2 & -30y^4 \end{bmatrix} \tag{23}$$

Optionally, we can also explicitly evaluate the Hessian at a certain point, e.g. $(x, y) = (1, 2)$:

$$\boldsymbol{H}_{(1,2)} = \begin{bmatrix} 6 & -2 \\ -2 & -480 \end{bmatrix} \tag{24}$$

2.2 Matrix Differentiation

It is possible to generalize the differentiation concepts applied while deriving the gradient vector and the Jacobian matrix to arbitrary scalar-by-vector, vector-by-scalar, vector-by-vector, scalar-by-matrix, and matrix-by-scalar combinations. Although it is possible to represent higher derivations (such as matrix-by-vector or matrix-by-matrix) in the form of tensors, this is beyond the scope of this course. Note that we will be using the numerator layout convention and hence represent vectors as column vectors.

Scalar by Vector

$$\frac{\partial y}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} & \frac{\partial y}{\partial x_2} & \cdots & \frac{\partial y}{\partial x_n} \end{bmatrix} \tag{25}$$

Vector by Scalar

$$\frac{\partial \boldsymbol{y}}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x} \\ \frac{\partial y_2}{\partial x} \\ \vdots \\ \frac{\partial y_n}{\partial x} \end{bmatrix} \tag{26}$$

Vector by Vector

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \frac{\partial y_m}{\partial x_2} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix} \tag{27}$$

Scalar by Matrix

$$\frac{\partial \boldsymbol{Y}}{\partial x} = \begin{bmatrix} \frac{\partial y_{11}}{\partial x} & \frac{\partial y_{12}}{\partial x} & \cdots & \frac{\partial y_{1n}}{\partial x} \\ \frac{\partial y_{21}}{\partial x} & \frac{\partial y_{22}}{\partial x} & \cdots & \frac{\partial y_{2n}}{\partial x} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_{m1}}{\partial x} & \frac{\partial y_{m2}}{\partial x} & \cdots & \frac{\partial y_{mn}}{\partial x} \end{bmatrix} \tag{28}$$

Matrix by Scalar

$$\frac{\partial x}{\partial \boldsymbol{Y}} = \begin{bmatrix} \frac{\partial x}{\partial y_{11}} & \frac{\partial x}{\partial y_{12}} & \cdots & \frac{\partial x}{\partial y_{1n}} \\ \frac{\partial x}{\partial y_{21}} & \frac{\partial x}{\partial y_{22}} & \cdots & \frac{\partial x}{\partial y_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x}{\partial y_{m1}} & \frac{\partial x}{\partial y_{m2}} & \cdots & \frac{\partial x}{\partial y_{mn}} \end{bmatrix} \tag{29}$$

Additional Resources

Additional matrix differentiation rules are discussed in *Matrix Differentiation* [Barnes, 2006]. Useful matrix differentiation identities can be found in the *Matrix Cookbook* [Petersen et al., 2008].

References

Randal J Barnes. Matrix differentiation. *Springs Journal*, pages 1–9, 2006.

Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.