

Principles of Machine Learning

Lecture 3: Calculus Concepts and Optimization Basics

Sharif University of Technology
Dept. of Aerospace Engineering

March 4, 2025



Table of Contents

1 Calculus Concepts

2 Optimization Basics



Overview of Calculus Concepts

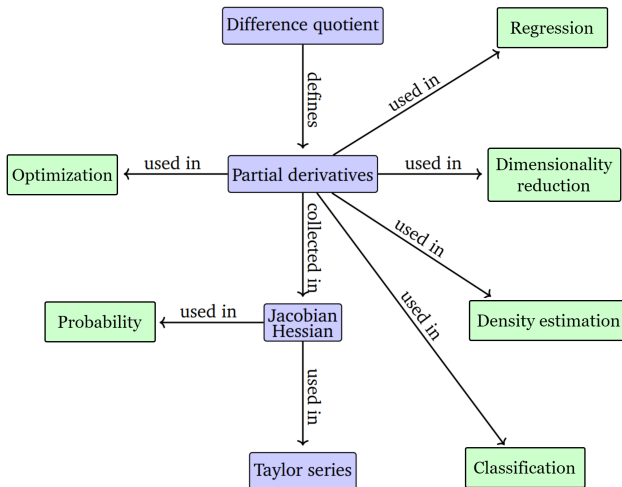


Figure: A mind map of the concepts introduced in this chapter



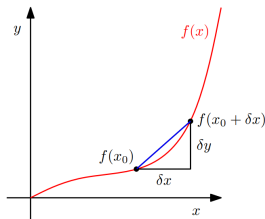
Differentiation of Univariate Functions

Definition (Difference Quotient)

Difference quotient of a univariate function $y = f(x)$, $x, y \in \mathbb{R}$ is defined as

$$\frac{\delta y}{\delta x} := \frac{f(x + \delta x) - f(x)}{\delta x}$$

which computes the slope of the secant line through two points on the graph of f . The difference quotient can also be considered the average slope of f between x and $x + \delta x$ if we assume f to be a linear function.



Definition (Derivative)

Derivative of a univariate function $y = f(x)$, $x, y \in \mathbb{R}$ at x is defined as the limit

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

and the secant becomes a tangent. The derivative of f points in the direction of steepest ascent of f .



Taylor Series

Definition (Taylor Polynomial)

Taylor polynomial of degree n of function $f: \mathbb{R} \rightarrow \mathbb{R}$ at x_0 is defined as

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

where $f^{(k)}(x_0)$ is the k -th derivative of f at x_0 and $\frac{f^{(k)}(x_0)}{k!}$ are the coefficients of the polynomial.

Definition (Taylor Series)

The Taylor series is a representation of a function f as an infinite sum of terms. *Taylor Series* of a smooth function $f \in \mathcal{C}^\infty$, $f: \mathbb{R} \rightarrow \mathbb{R}$ at x_0 is defined as

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

Examples (Maclaurin series expansion of a harmonic function)

Consider the function

$$f(x) = \sin(x) + \cos(x) \in \mathcal{C}^\infty.$$

We seek a Taylor series expansion of f at $x_0 = 0$, which is the Maclaurin series expansion of f . We obtain the following derivatives:

$$f(0) = \sin(0) + \cos(0) = 1$$

$$f'(0) = \cos(0) - \sin(0) = 1$$

$$f''(0) = -\sin(0) - \cos(0) = -1$$

$$f^{(3)}(0) = -\cos(0) + \sin(0) = -1$$

$$f^{(4)}(0) = \sin(0) + \cos(0) = f(0) = 1$$

\vdots

Examples (Contd.)

We can see a pattern here: The coefficients in our Taylor series are only ± 1 , each of which occurs twice before switching to the other one.

Therefore, we have:

$$\begin{aligned}T_{\infty}(x) &= \sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!} x^k \\&= 1 + x - \frac{1}{2!}x^2 - \frac{1}{3!}x^3 + \frac{1}{4!}x^4 + \frac{1}{5!}x^5 - \dots \\&= 1 - \frac{1}{2!}x^2 + \frac{1}{4!}x^4 \mp \dots + x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 \mp \dots \\&= \sum_{k=0}^{\infty} (-1)^k \frac{1}{(2k)!} x^{2k} + \sum_{k=0}^{\infty} (-1)^k \frac{1}{(2k+1)!} x^{2k+1} \\&= \cos(x) + \sin(x)\end{aligned}$$

Differentiation Rules

- Product rule:

$$(f(x)g(x))' = f'(x)g(x) + f(x)g'(x)$$

- Quotient rule:

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

- Sum rule:

$$(f(x) + g(x))' = f'(x) + g'(x)$$

- Chain rule:

$$(g(f(x)))' = (g \circ f)'(x) = g'(f(x))f'(x)$$



Partial Differentiation and Gradients

Definition (Partial Derivative)

For a function $f: \mathbb{R}^n \rightarrow \mathbb{R}$, $\mathbf{x} \rightarrow f(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^n$ of n variables x_1, \dots, x_n we define the *partial derivatives* as

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

and collect them in the row vector

$$\nabla_{\mathbf{x}} f = \text{grad} f = \frac{df}{d\mathbf{x}} = \left[\frac{\partial f(\mathbf{x})}{\partial x_1} \quad \frac{\partial f(\mathbf{x})}{\partial x_2} \quad \dots \quad \frac{\partial f(\mathbf{x})}{\partial x_n} \right] \in \mathbb{R}^{1 \times n}$$

where n is the number of variables and 1 is the dimension of the range of f and $\mathbf{x} = [x_1, \dots, x_n]^T \in \mathbb{R}^n$. $\nabla_{\mathbf{x}} f$ is called the *gradient* of f .

Chain Rule

Consider a function $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ of two variables x_1, x_2 which $x_1(t)$ and $x_2(t)$ are themselves function of t . Thus, the gradient of f wrt t is computed as

$$\frac{df}{dt} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix} \begin{bmatrix} \frac{\partial x_1(t)}{\partial t} \\ \frac{\partial x_2(t)}{\partial t} \end{bmatrix} = \frac{\partial f}{\partial x_1} \frac{\partial x_1}{\partial t} + \frac{\partial f}{\partial x_2} \frac{\partial x_2}{\partial t}$$

where d denotes the gradient and ∂ partial derivatives.

Examples (Chain rule for partial derivatives)

Consider $f(x_1, x_2) = x_1^2 + 2x_2$, where $x_1 = \sin t$ and $x_2 = \cos t$, then

$$\begin{aligned} \frac{df}{dt} &= 2 \sin t \frac{\partial \sin t}{\partial t} + 2 \frac{\partial \cos t}{\partial t} \\ &= 2 \sin t \cos t - 2 \sin t = 2 \sin t (\cos t - 1) \end{aligned}$$

Chain Rule

If $f(x_1, x_2)$ is a function of x_1 and x_2 , where $x_1(s, t)$ and $x_2(s, t)$ are themselves functions of two variables s and t , the chain rule yields the partial derivatives

$$\begin{aligned}\frac{\partial f}{\partial s} &= \frac{\partial f}{\partial x_1} \frac{\partial x_1}{\partial s} + \frac{\partial f}{\partial x_2} \frac{\partial x_2}{\partial s} \\ \frac{\partial f}{\partial t} &= \frac{\partial f}{\partial x_1} \frac{\partial x_1}{\partial t} + \frac{\partial f}{\partial x_2} \frac{\partial x_2}{\partial t}\end{aligned}$$

and the gradient is obtained by the matrix multiplication

$$\frac{df}{d(s, t)} = \frac{\partial f}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial (s, t)} = \underbrace{\begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix}}_{\frac{\partial f}{\partial \mathbf{x}}} \underbrace{\begin{bmatrix} \frac{\partial x_1}{\partial s} & \frac{\partial x_1}{\partial t} \\ \frac{\partial x_2}{\partial s} & \frac{\partial x_2}{\partial t} \end{bmatrix}}_{\frac{\partial \mathbf{x}}{\partial (s, t)}}$$



Gradients of Vector-Valued Functions

Definition (Jacobian)

The collection of all first-order partial derivatives of a vector-valued function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is called the *Jacobian*. The Jacobian \mathbf{J} is an $m \times n$ matrix, which we define and arrange as follows:

$$\mathbf{J} = \frac{d\mathbf{f}(\mathbf{x})}{d\mathbf{x}} = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_m(\mathbf{x})}{\partial x_n} \end{bmatrix}$$



Gradients of Vector-Valued Functions

Examples (Gradient of a Linear Vector-Valued Function)

Consider the function:

$$\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}, \quad \mathbf{f}(\mathbf{x}) \in \mathbb{R}^M, \quad \mathbf{A} \in \mathbb{R}^{M \times N}, \quad \mathbf{x} \in \mathbb{R}^N$$

To compute the gradient $d\mathbf{f}/d\mathbf{x}$ for the function $\mathbf{f} : \mathbb{R}^N \rightarrow \mathbb{R}^M$, we have:

$$f_i(\mathbf{x}) = \sum_{j=1}^N A_{ij}x_j \rightarrow \frac{\partial f_i}{\partial x_j} = A_{ij}$$


We collect the partial derivatives in the Jacobian and obtain the gradient

$$\frac{d\mathbf{f}}{d\mathbf{x}} = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_M(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_M(\mathbf{x})}{\partial x_N} \end{bmatrix} = \begin{bmatrix} A_{11} & \cdots & A_{1N} \\ \vdots & \ddots & \vdots \\ A_{M1} & \cdots & A_{MN} \end{bmatrix} = \mathbf{A} \in \mathbb{R}^{M \times N}$$

Gradient of a Linear Vector-Valued Function

Examples (Gradient of a Least-Squares Loss in a Linear Mode)

Let us consider the linear model

$$\begin{matrix} \textcircled{x_1, y_1} \\ x_2, y_2 \\ \vdots \end{matrix} \quad \hat{y} = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \quad \begin{matrix} \uparrow \\ \text{y} = \Phi \theta \end{matrix}$$


where $\theta \in \mathbb{R}^D$ is a parameter vector, $\Phi \in \mathbb{R}^{N \times D}$ are input features and $y \in \mathbb{R}^N$ are the corresponding observations. We define the functions:

$$\begin{aligned} e_1 &= y_1 - (a_0 + a_1 x_1 + a_2 x_1^2 + a_3 x_1^3) \\ e_2 &= y_2 - (\dots x_2 \dots) \\ e_N &= y_N - (\dots x_N) \end{aligned} \quad L(e) := \|e\|^2 \quad e(\theta) := y - \Phi \theta$$

$$\rightarrow \begin{bmatrix} e_1 \\ \vdots \\ e_N \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} - \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 & x_N^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

where L is called a *least-squares loss* function. Thus, using the chain rule, we have:

$$\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial e} \frac{\partial e}{\partial \theta}$$

Gradient of a Linear Vector-Valued Function

Examples (Contd.)

Considering $\|\mathbf{e}\|^2 = \mathbf{e}^T \mathbf{e}$, we have:

$$L = \mathbf{e}^T \mathbf{e}$$
$$\frac{\partial L}{\partial \mathbf{e}} = \mathbf{e}^T + \mathbf{e}^T = 2\mathbf{e}^T$$

$$\frac{\partial L}{\partial \mathbf{e}} = 2\mathbf{e}^T \in \mathbb{R}^{1 \times N}$$
$$\frac{\partial \mathbf{e}}{\partial \boldsymbol{\theta}} = -\boldsymbol{\Phi} \in \mathbb{R}^{N \times D}$$

$$L = e_1^2 + e_2^2 + \dots$$
$$\frac{\partial L}{\partial \mathbf{e}} = \begin{bmatrix} \frac{\partial L}{\partial e_1} \\ \vdots \\ \frac{\partial L}{\partial e_n} \end{bmatrix} = \begin{bmatrix} 2e_1 \\ \vdots \\ 2e_n \end{bmatrix} = 2 \begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix}$$

Hence:

$$\frac{\partial L}{\partial \boldsymbol{\theta}} = -2\mathbf{e}^T \boldsymbol{\Phi} = -2(\mathbf{y}^T - \boldsymbol{\theta}^T \boldsymbol{\Phi}^T) \boldsymbol{\Phi} \in \mathbb{R}^{1 \times D}$$

Alternative way:

$$L_2(\boldsymbol{\theta}) := \|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\theta}\|^2 = (\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\theta})^T (\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\theta})$$



Gradients of Matrices

Gradient computation of matrix $\mathbf{A}_{m \times n}$ with respect to a vector $\mathbf{x}_{k \times 1}$:

Approach 1: Computing the partial derivatives $\partial \mathbf{A} / \partial x_i$ and collate them in a $m \times n \times k$ Jacobian tensor.

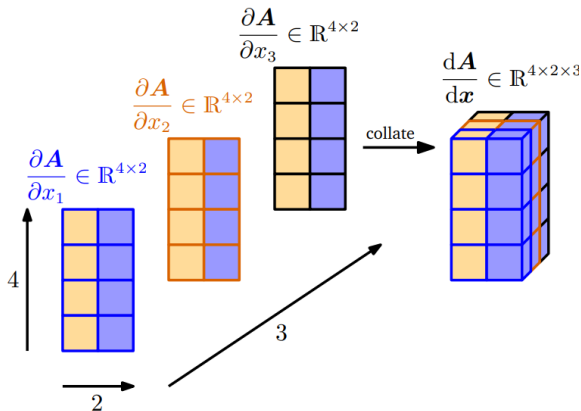


Figure: An illustrative example of Approach 1 with $\mathbf{A} \in \mathbb{R}^{4 \times 2}$ and $\mathbf{x} \in \mathbb{R}^3$



Gradients of Matrices

Approach 2:

- 1 Flattening (reshaping) of the matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ into a vector $\tilde{\mathbf{A}} \in \mathbb{R}^{mn}$
- 2 Computing the gradient $\frac{d\tilde{\mathbf{A}}}{d\mathbf{x}} \in \mathbb{R}^{mn \times k}$
- 3 Reshape this gradient to obtain the Jacobian tensor.

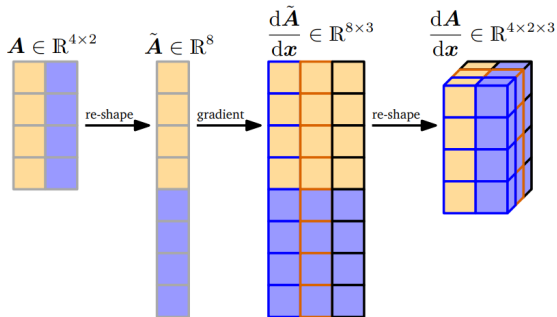


Figure: An illustrative example of Approach 2 with $\mathbf{A} \in \mathbb{R}^{4 \times 2}$ and $\mathbf{x} \in \mathbb{R}^3$



Examples (Gradient of Vectors with Respect to Matrices)

Consider the following function \mathbf{f} and compute $d\mathbf{f}/d\mathbf{A}$.

$$\mathbf{f} = \mathbf{A}\mathbf{x}, \quad \mathbf{f} \in \mathbb{R}^M, \quad \mathbf{A} \in \mathbb{R}^{M \times N}, \quad \mathbf{x} \in \mathbb{R}^N$$

By definition, the gradient is the collection of the partial derivatives:

$$\frac{d\mathbf{f}}{d\mathbf{A}} = \begin{bmatrix} \frac{\partial f_1}{\partial \mathbf{A}} \\ \vdots \\ \frac{\partial f_M}{\partial \mathbf{A}} \end{bmatrix}, \quad \frac{\partial f_i}{\partial \mathbf{A}} \in \mathbb{R}^{1 \times (M \times N)}$$

$$f_i = \sum_{j=1}^N A_{ij}x_j, \quad i = 1, \dots, M$$

$$\frac{\partial f_i}{\partial A_{iq}} = x_q$$

Examples (Contd.)

Thus, partial derivatives of f_i wrt each row of \mathbf{A} is:

$$\frac{\partial f_i}{\partial A_{i,:}} = \mathbf{x}^T \in \mathbb{R}^{1 \times 1 \times N}$$
$$\frac{\partial f_i}{\partial A_{k \neq i,:}} = \mathbf{0}^T \in \mathbb{R}^{1 \times 1 \times N}$$

Hence, by stacking the partial derivatives we get:

$$\frac{\partial f_i}{\partial \mathbf{A}} = [\mathbf{0}^T \quad \dots \quad \mathbf{0}^T \quad \mathbf{x}^T \quad \mathbf{0}^T \quad \dots \quad \mathbf{0}^T] \in \mathbb{R}^{1 \times (M \times N)}$$



Examples (Gradient of Matrices with Respect to Matrices)

Consider a matrix $\mathbf{R} \in \mathbb{R}^{M \times N}$ and function $\mathbf{f} : \mathbb{R}^{M \times N} \rightarrow \mathbb{R}^{N \times N}$ with

$$\mathbf{f}(\mathbf{R}) = \mathbf{R}^T \mathbf{R} := \mathbf{K} \in \mathbb{R}^{N \times N}$$

To compute $d\mathbf{K}/d\mathbf{R}$, we know that

$$\frac{dK_{pq}}{d\mathbf{R}} \in \mathbb{R}^{1 \times M \times N}, \quad p, q = 1, \dots, N$$

where K_{pq} is the (p, q) th entry of $\mathbf{K} = \mathbf{f}(\mathbf{R})$. Thus, we have:

$$K_{pq} = \mathbf{r}_p^T \mathbf{r}_q = \sum_{m=1}^M R_{mp} R_{mq}$$

where \mathbf{r}_i denotes the i th column of \mathbf{R} .

Examples (Contd.)

Entries of the gradient tensor is given by ∂_{pqij}

$$\frac{\partial K_{pq}}{\partial R_{ij}} = \sum_{m=1}^M \frac{\partial}{\partial R_{ij}} R_{mp} R_{mq} = \partial_{pqij} = \begin{cases} R_{iq} & \text{if } j = p, p \neq q \\ R_{ip} & \text{if } j = q, p \neq q \\ 2R_{iq} & \text{if } j = p, p = q \\ 0 & \text{otherwise} \end{cases}$$

where $p, q, j = 1, \dots, N$ and $i = 1, \dots, M$ and the desired gradient has the dimension $(N \times N) \times (M \times N)$.



Useful Identities for Computing Gradients

Useful gradients that are frequently required in machine learning:

- $\frac{\partial}{\partial \mathbf{X}} \mathbf{f}(\mathbf{X})^T = \left(\frac{\partial \mathbf{f}(\mathbf{X})}{\partial \mathbf{X}} \right)^T$
- $\frac{\partial}{\partial \mathbf{X}} \text{tr}(\mathbf{f}(\mathbf{X})) = \text{tr} \left(\frac{\partial \mathbf{f}(\mathbf{X})}{\partial \mathbf{X}} \right)$
- $\frac{\partial}{\partial \mathbf{X}} \det(\mathbf{f}(\mathbf{X})) = \det(\mathbf{f}(\mathbf{X})) \text{tr} \left(\mathbf{f}(\mathbf{X})^{-1} \frac{\partial \mathbf{f}(\mathbf{X})}{\partial \mathbf{X}} \right)$
- $\frac{\partial}{\partial \mathbf{X}} \mathbf{f}(\mathbf{X})^{-1} = -\mathbf{f}(\mathbf{X})^{-1} \frac{\partial \mathbf{f}(\mathbf{X})}{\partial \mathbf{X}} \mathbf{f}(\mathbf{X})^{-1}$
- $\frac{\partial \mathbf{a}^T \mathbf{X}^{-1} \mathbf{b}}{\partial \mathbf{X}} = -(\mathbf{X}^{-1})^T \mathbf{a} \mathbf{b}^T (\mathbf{X}^{-1})^T$



Useful Identities for Computing Gradients

- $\frac{\partial \mathbf{x}^T \mathbf{a}}{\partial \mathbf{x}} = \mathbf{a}^T$
- $\frac{\partial \mathbf{a}^T \mathbf{x}}{\partial \mathbf{x}} = \mathbf{a}^T$
- $\frac{\partial \mathbf{a}^T \mathbf{X} \mathbf{b}}{\partial \mathbf{X}} = \mathbf{a} \mathbf{b}^T$
- $\frac{\partial \mathbf{x}^T \mathbf{B} \mathbf{x}}{\partial \mathbf{x}} = \mathbf{x}^T (\mathbf{B} + \mathbf{B}^T)$
- $\frac{\partial}{\partial \mathbf{s}} (\mathbf{x} - \mathbf{A} \mathbf{s})^T \mathbf{W} (\mathbf{x} - \mathbf{A} \mathbf{s}) = -2 (\mathbf{x} - \mathbf{A} \mathbf{s})^T \mathbf{W} \mathbf{A}$ for symmetric \mathbf{W}



Gradients in a Deep Network

Application of chain rule in multi-layer neural networks to compute gradient of loss function with respect to parameters:

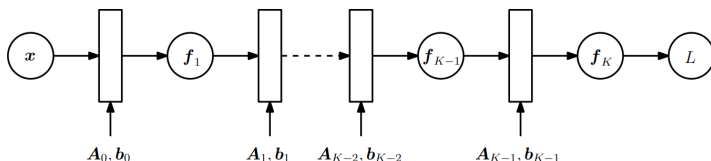


Figure: Forward pass in a multi-layer neural network

$$\mathbf{f}_0 := \mathbf{x}$$

$$\mathbf{f}_i := \sigma_i(\mathbf{A}_{i-1}\mathbf{f}_{i-1} + \mathbf{b}_{i-1}), \quad i = 1, \dots, K$$

We want to find parameters $\theta = \{\mathbf{A}_0, \mathbf{b}_0, \dots, \mathbf{A}_{K-1}, \mathbf{b}_{K-1}\}$, such that the following squared loss is minimized.

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



Gradients in a Deep Network

By defining the parameter $\theta_j = \{\mathbf{A}_j, \mathbf{b}_j\}$ for each layer $j = 0, \dots, K-1$ and using the chain rule, we can determine the partial derivatives of L wrt θ_j as

$$\begin{aligned}\frac{\partial L}{\partial \theta_{K-1}} &= \frac{\partial L}{\partial \mathbf{f}_K} \frac{\partial \mathbf{f}_K}{\partial \theta_{K-1}} \\ \frac{\partial L}{\partial \theta_{K-2}} &= \frac{\partial L}{\partial \mathbf{f}_K} \boxed{\frac{\partial \mathbf{f}_K}{\partial \mathbf{f}_{K-1}} \frac{\partial \mathbf{f}_{K-1}}{\partial \theta_{K-2}}} \\ \frac{\partial L}{\partial \theta_{K-3}} &= \frac{\partial L}{\partial \mathbf{f}_K} \frac{\partial \mathbf{f}_K}{\partial \mathbf{f}_{K-1}} \boxed{\frac{\partial \mathbf{f}_{K-1}}{\partial \mathbf{f}_{K-2}} \frac{\partial \mathbf{f}_{K-2}}{\partial \theta_{K-3}}} \\ \frac{\partial L}{\partial \theta_i} &= \frac{\partial L}{\partial \mathbf{f}_K} \frac{\partial \mathbf{f}_K}{\partial \mathbf{f}_{K-1}} \cdots \boxed{\frac{\partial \mathbf{f}_{i+2}}{\partial \mathbf{f}_{i+1}} \frac{\partial \mathbf{f}_{i+1}}{\partial \theta_i}}\end{aligned}$$

The orange terms are partial derivatives of the output of a layer wrt its inputs, whereas the blue terms are wrt its parameters.



Higher-Order Derivatives

Consider a function $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ of two variables x, y . The notation for higher-order partial derivatives is as follows:

- $\frac{\partial^2 f}{\partial x^2}$ is the second partial derivative of f wrt x .
- $\frac{\partial^n f}{\partial x^n}$ is the n th partial derivative of f wrt x .
- $\frac{\partial^2 f}{\partial y \partial x} = \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial x} \right)$ is the partial derivative obtained by first partial differentiating wrt to x and then with respect to y .
- $\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial y} \right)$ is the partial derivative obtained by first partial differentiating by y and then x .

If $f(x, y)$ is a twice (continuously) differentiable function, then the order of differentiation does not matter.

The Hessian matrix is the collection of all second-order partial derivatives:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

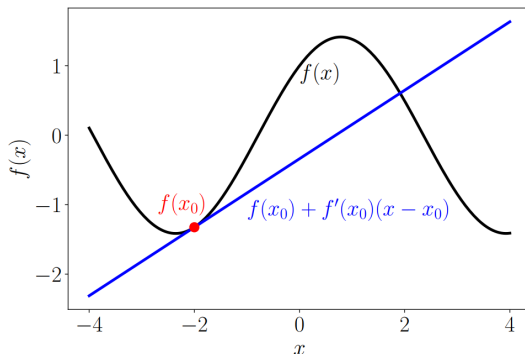


Linearization and Multivariate Taylor Series

The gradient ∇f of a function f is often used for a locally linear approximation f around \mathbf{x}_0 :

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + (\nabla_{\mathbf{x}} f)(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)$$

This approximation is locally accurate, but the farther we move away from \mathbf{x}_0 the worse the approximation gets.



Linearization and Multivariate Taylor Series

Definition (Multivariate Taylor Series)

We consider a function $f: \mathbb{R}^D \rightarrow \mathbb{R}, \mathbf{x} \rightarrow f(\mathbf{x}), \mathbf{x} \in \mathbb{R}^D$ that is smooth at \mathbf{x}_0 . By defining $\delta := \mathbf{x} - \mathbf{x}_0$, the *multivariate Taylor series* of f at \mathbf{x}_0 is defined as

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

where $D_{\mathbf{x}}^k f(\mathbf{x}_0)$ is the k -th (total) derivative of f wrt \mathbf{x} , evaluated at \mathbf{x}_0

Definition (Taylor Polynomial)

The *Taylor polynomial* of degree n of f at \mathbf{x}_0 contains the first $n + 1$ components of the Taylor series and is defined as

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

Linearization and Multivariate Taylor Series

Both $D_{\mathbf{x}}^k f(\mathbf{x}_0)$ and δ^k are k -th order tensors, i.e., k -dimensional arrays. The

k th-order tensor $\delta^k \in \mathbb{R}^{\overbrace{D \times D \times \cdots \times D}^{k \text{ times}}}$ is obtained as a k -fold outer product, denoted by \otimes , of the vector $\delta \in \mathbb{R}^D$. For example:

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

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

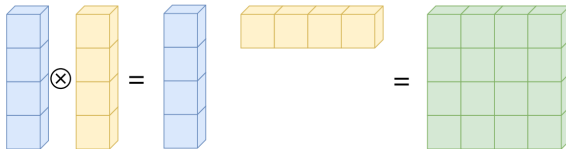


Figure: Given a vector $\delta \in \mathbb{R}^4$, we obtain the outer product $\delta^2 \in \mathbb{R}^{4 \times 4}$ as a matrix



Linearization and Multivariate Taylor Series

In general, we have:

$$D_{\mathbf{x}}^k f(\mathbf{x}_0) \delta^k = \sum_{i_1=1}^D \cdots \sum_{i_k=1}^D D_{\mathbf{x}}^k f(\mathbf{x}_0)[i_1, \dots, i_k] \delta[i_1] \cdots \delta[i_k]$$

The first terms $D_{\mathbf{x}}^k f(\mathbf{x}_0) \delta^k$ of the Taylor series expansion are:

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

$$k = 1 : D_{\mathbf{x}}^1 f(\mathbf{x}_0) \delta^1 = \nabla_{\mathbf{x}} f(\mathbf{x}_0) \delta = \sum_{i=1}^D \nabla_{\mathbf{x}} f(\mathbf{x}_0)[i] \delta[i] \in \mathbb{R}$$

$$k = 2 : D_{\mathbf{x}}^2 f(\mathbf{x}_0) \delta^2 = \text{tr}(\mathbf{H}(\mathbf{x}_0) \delta \delta^T) = \delta^T \mathbf{H}(\mathbf{x}_0) \delta = \sum_{i=1}^D \sum_{j=1}^D H[i, j] \delta[i] \delta[j] \in \mathbb{R}$$

where $\mathbf{H}(\mathbf{x}_0)$ is the Hessian of f evaluated at \mathbf{x}_0 .



Linearization and Multivariate Taylor Series

Examples (Taylor Series Expansion of a Function with Two Variables)

Consider the function

$$f(x, y) = x^2 + 2xy + y^3$$

We want to compute the Taylor series expansion of f at $(x_0, y_0) = (1, 2)$. We start with the constant term and the first-order derivatives, which are given by $f(1, 2) = 13$ and

$$\begin{aligned}\frac{\partial f}{\partial x} &= 2x + 2y \rightarrow \frac{\partial f}{\partial x}(1, 2) = 6 \\ \frac{\partial f}{\partial y} &= 2x + 3y^2 \rightarrow \frac{\partial f}{\partial y}(1, 2) = 14\end{aligned}$$

Therefore, we obtain:

$$D_{x,y}^1 f(1, 2) = \nabla_{x,y} f(1, 2) = \left[\frac{\partial f}{\partial x}(1, 2) \quad \frac{\partial f}{\partial y}(1, 2) \right] = \begin{bmatrix} 6 & 14 \end{bmatrix} \in \mathbb{R}^{1 \times 2}$$

Linearization and Multivariate Taylor Series

Examples (Contd.)

such that

$$\frac{D_{x,y}^1 f(1, 2)}{1!} \delta = \begin{bmatrix} 6 & 14 \end{bmatrix} \begin{bmatrix} x-1 \\ y-2 \end{bmatrix} = 6(x-1) + 14(y-2)$$

The second-order partial derivatives are given by

$$\frac{\partial^2 f}{\partial x^2} = 2 \rightarrow \frac{\partial^2 f}{\partial x^2}(1, 2) = 2$$

$$\frac{\partial^2 f}{\partial y^2} = 6y \rightarrow \frac{\partial^2 f}{\partial y^2}(1, 2) = 12$$

$$\frac{\partial^2 f}{\partial y \partial x} = 2 \rightarrow \frac{\partial^2 f}{\partial y \partial x}(1, 2) = 2$$

$$\frac{\partial^2 f}{\partial x \partial y} = 2 \rightarrow \frac{\partial^2 f}{\partial x \partial y}(1, 2) = 2$$

Linearization and Multivariate Taylor Series

Examples (Contd.)

The Hessian is obtained as

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 2 & 6y \end{bmatrix} \rightarrow \mathbf{H}(1, 2) = \begin{bmatrix} 2 & 2 \\ 2 & 12 \end{bmatrix}$$

Therefore, we have:

$$\begin{aligned} \frac{D_{x,y}^2 f(1, 2)}{2!} \delta^2 &= \frac{1}{2} \delta^T \mathbf{H}(1, 2) \delta \\ &= \frac{1}{2} [x-1 \quad y-2] \begin{bmatrix} 2 & 2 \\ 2 & 12 \end{bmatrix} \begin{bmatrix} x-1 \\ y-2 \end{bmatrix} \\ &= (x-1)^2 + 2(x-1)(y-2) + 6(y-2)^2 \end{aligned}$$

Handwritten red note: $\frac{1}{2} [(x-x_0) \quad (y-y_0)] \sim \begin{bmatrix} x-x_0 \\ y-y_0 \end{bmatrix}$

Linearization and Multivariate Taylor Series

Examples (Contd.)

The third-order derivatives are obtained as

$$D_{x,y}^3 f = \begin{bmatrix} \frac{\partial \mathbf{H}}{\partial x} & \frac{\partial \mathbf{H}}{\partial y} \end{bmatrix} \in \mathbb{R}^{2 \times 2 \times 2},$$

$$D_{x,y}^3 f[:, :, 1] = \frac{\partial \mathbf{H}}{\partial x} = \begin{bmatrix} \frac{\partial^3 f}{\partial x^3} & \frac{\partial^3 f}{\partial x^2 \partial y} \\ \frac{\partial^3 f}{\partial x \partial y \partial x} & \frac{\partial^3 f}{\partial x \partial y^2} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$D_{x,y}^3 f[:, :, 2] = \frac{\partial \mathbf{H}}{\partial y} = \begin{bmatrix} \frac{\partial^3 f}{\partial y \partial x^2} & \frac{\partial^3 f}{\partial y \partial x \partial y} \\ \frac{\partial^3 f}{\partial y^2 \partial x} & \frac{\partial^3 f}{\partial y^3} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 6 \end{bmatrix}$$

Therefore, we have:

$$\frac{D_{x,y}^3 f(1, 2)}{3!} \delta^3 = (y - 2)^3$$



Examples (Contd.)

The Taylor series expansion of f at $(x_0, y_0) = (1, 2)$ is

$$\begin{aligned} f(\mathbf{x}) &= f(1, 2) + D_{x,y}^1 f(1, 2) \delta + \frac{D_{x,y}^2 f(1, 2)}{2!} \delta^2 + \frac{D_{x,y}^3 f(1, 2)}{3!} \delta^3 \\ &= 13 + 6(x - 1) + 14(y - 2) + (x - 1)^2 + 6(y - 2)^2 + 2(x - 1)(y - 2) + \end{aligned}$$



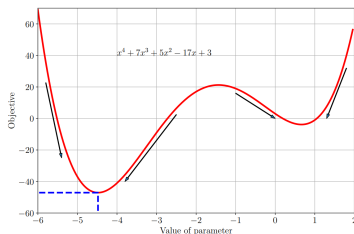
Table of Contents

- 1 Calculus Concepts
- 2 Optimization Basics



Continuous Optimization

- Training a machine learning model boils down to finding a good set of parameters.
- Given an objective function, finding the best value is done using optimization algorithms.
- Finding the best value means moving downhill, opposite to the gradient, to reach the deepest point of the objective function.
- In general, the objective function can have multiple local minima but only one global minimum. For convex functions, any local minimum is also the global minimum.



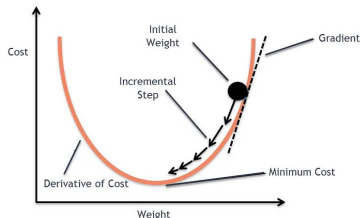
Optimization Using Gradient Descent

We consider the problem of solving for the minimum of a real-valued function

$$\min_{\mathbf{x}} f(\mathbf{x})$$

where $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is an objective function. We assume that our function f is differentiable.

- Gradient descent is a first-order optimization algorithm.
- Gradient descent: move towards the error (local) minimum.
- Compute gradient, which implies getting direction to the cost minimum.
- Take steps proportional to the negative of the gradient of the function at the current point.



Optimization Using Gradient Descent

- Gradient descent exploits the fact that $f(\mathbf{x}_0)$ decreases fastest if one moves from \mathbf{x}_0 in the direction of the negative gradient $-((\nabla f)(\mathbf{x}_0))^T$ of f at \mathbf{x}_0 .
- The gradient descent iterative rule is

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \gamma_i ((\nabla f)(\mathbf{x}_i))^T$$

- We start with an initial guess \mathbf{x}_0 .
- For a suitable step-size (learning rate) γ_i , this algorithm $(f(\mathbf{x}_0) \geq f(\mathbf{x}_1) \geq \dots)$ converges to a local minimum.
- Gradient descent can be relatively slow close to the minimum: Its asymptotic rate of convergence is inferior to many other methods.



Optimization Using Gradient Descent

Examples (Gradient Descent)

Consider a quadratic function of $\mathbf{x} = [x_1 \ x_2]^T$

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \begin{bmatrix} 2 & 1 \\ 1 & 20 \end{bmatrix} \mathbf{x} - [5 \ 3] \mathbf{x}$$

The gradient is

$$\nabla f(\mathbf{x}) = \mathbf{x}^T \begin{bmatrix} 2 & 1 \\ 1 & 20 \end{bmatrix} - [5 \ 3] = [2x_1 + x_2 - 5 \ x_1 + 20x_2 - 3]$$

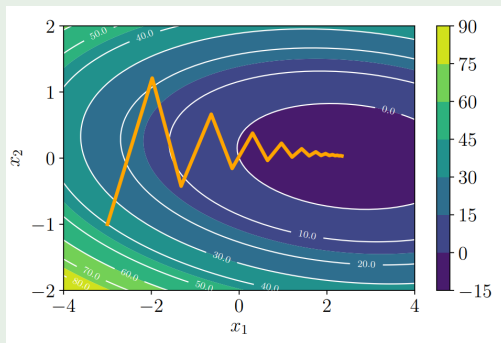
Starting at the initial location $\mathbf{x}_0 = [-3 \ -1]^T$ we iteratively apply the gradient descent rule to obtain a sequence of estimates that converge to the minimum value.



Optimization Using Gradient Descent

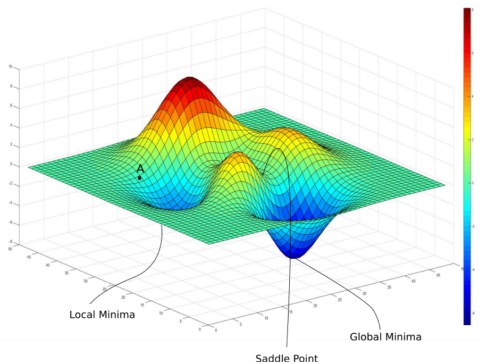
Examples (Contd.)

The gradient points in a direction that is orthogonal to the contour lines of the function we wish to optimize.



Challenges with Gradient Descent

- **Local minimum:** A local minimum is a minimum within some neighborhood that need not be (but may be) a global minimum.
- **Saddle points:** For non-convex functions, having the gradient to be 0 is not good enough. Example: $f(\mathbf{x}) = x_1^2 - x_2^2$ at $\mathbf{x} = (0, 0)$ has zero gradient but it is clearly not a local minimum as $\mathbf{x} = (0, \epsilon)$ has smaller function value.



Challenges with Gradient Descent

- If the step-size is too small, gradient descent can be slow.
- If the step-size is chosen too large, gradient descent can overshoot, fail to converge, or even diverge.

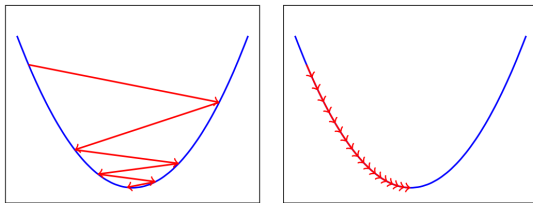


Figure: Effect of step size. Left: Big step size, Right: Small step size

Therefore, an adaptive mechanism should be employed to adjust the step size at each iteration based on changes in the function value.

