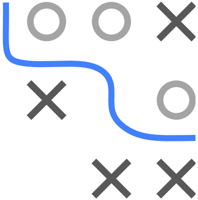
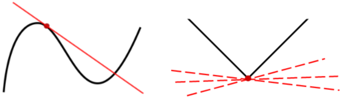
Optimization in Machine Learning



Mathematical Concepts
Differentiation and Derivatives



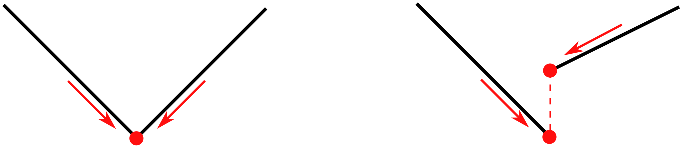
Learning goals

- Definition of smoothness
- Uni- & multivariate differentiation
- Gradient, partial derivatives
- Jacobian matrix
- Hessian matrix

CONTINUITY

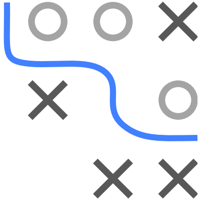
Definition: A function $f : \mathcal{S} \subseteq \mathbb{R} \to \mathbb{R}$ is said to be **continuous** for each inner point $x_0 \in \mathcal{S}$ with $x \in \mathcal{S}$ if the following limit exists:

$$\lim_{x \to x_0} f(x) = f(x_0)$$



Left: Function is continuous everywhere. **Right:** Not continuous at the red point.

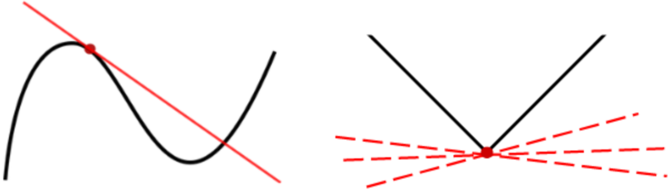
UNIVARIATE DIFFERENTIABILITY



Definition: A function $f : \mathcal{S} \subseteq \mathbb{R} \to \mathbb{R}$ is said to be **differentiable** for each inner point $x \in \mathcal{S}$ if the following limit exists:

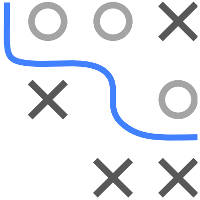
$$f'(x) := \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Intuitively: f can be approxed locally by a lin. fun. with slope $m = f'(x)$.

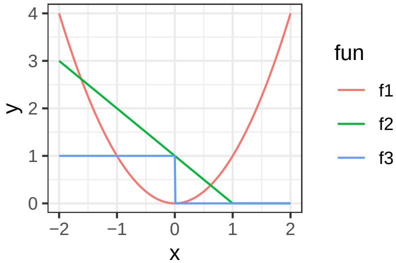


Left: Function is differentiable everywhere. **Right:** Not differentiable at the red point.

SMOOTH VS. NON-SMOOTH



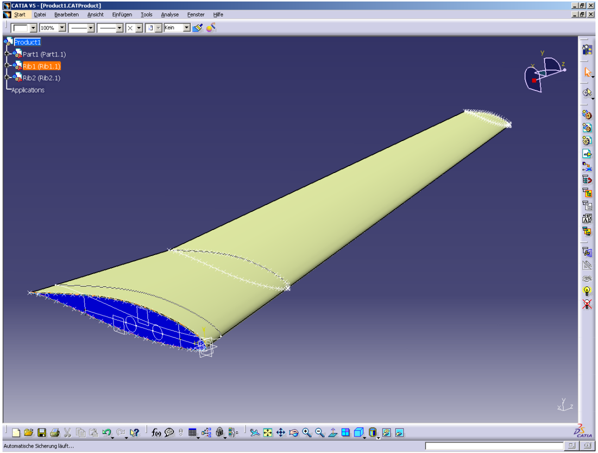
- **Smoothness** of a function $f : \mathcal{S} \to \mathbb{R}$ is measured by the number of its continuous derivatives
- $\mathcal{C}^k$ is class of k-times continuously differentiable functions ($f \in \mathcal{C}^k$ means $f^{(k)}$ exists and is continuous)
- In this lecture, we call f “smooth”, if at least $f \in \mathcal{C}^1$



f_1 is smooth, f_2 is continuous but not differentiable, and f_3 is non-continuous.

SMOOTH VS. NON-SMOOTH

Example: Wing-construction in Computer Aided Design (CAD)

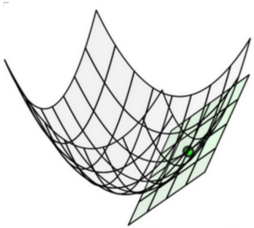


Source: https://ww3.cad.de/foren/ubb/Forum133/HTML/009359.shtml

MULTIVARIATE DIFFERENTIABILITY

Definition: $f : \mathcal{S} \subseteq \mathbb{R}^d \to \mathbb{R}$ is **differentiable** in $\mathbf{x} \in \mathcal{S}$ if there exists a (continuous) linear map $\nabla f(\mathbf{x}) : \mathcal{S} \subseteq \mathbb{R}^d \to \mathbb{R}^d$ with

$$\lim_{\boldsymbol{h} \to 0} \frac{f(\mathbf{x} + \boldsymbol{h}) - f(\mathbf{x}) - \nabla f(\mathbf{x})^T \cdot \boldsymbol{h}}{||\boldsymbol{h}||} = 0$$



Geometrically: The function can be locally approximated by a tangent hyperplane.

Source: `https://github.com/jermwatt/machine_learning_refined`.

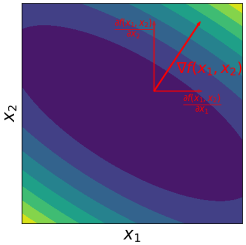
GRADIENT

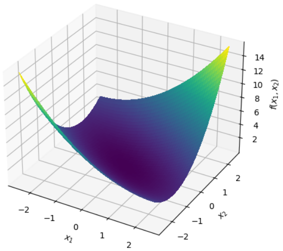
- Linear approximation is given by the **gradient**:

$$\nabla f = \frac{\partial f}{\partial x_1}\boldsymbol{e}_1 + \cdots + \frac{\partial f}{\partial x_d}\boldsymbol{e}_d = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_d}\right)^T$$

- Elements of the gradient are called **partial derivatives**.
- To compute $\partial f/\partial x_j$, regard f as function of x_j only (others fixed)

Example: $f(\mathbf{x}) = \frac{x_1^2}{2} + x_1 x_2 + x_2^2 \Rightarrow \nabla f(\mathbf{x}) = (x_1 + x_2, x_1 + 2x_2)^T$





DIRECTIONAL DERIVATIVE

The **directional derivative** tells how fast $f : \mathcal{S} \to \mathbb{R}$ is changing w.r.t. an arbitrary direction $\boldsymbol{v}$:

$$D_{\boldsymbol{v}} f(\mathbf{x}) := \lim_{h \to 0} \frac{f(\mathbf{x} + h\boldsymbol{v}) - f(\mathbf{x})}{h} = \nabla f(\mathbf{x})^T \cdot \boldsymbol{v}.$$

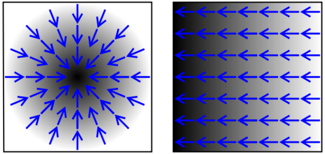
Example: The directional derivative for $\boldsymbol{v} = (1, 1)$ is:

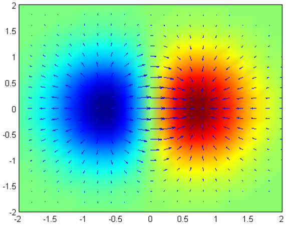
$$D_{\boldsymbol{v}} f(\mathbf{x}) = \nabla f(\mathbf{x})^T \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{\partial f}{\partial x_1} + \frac{\partial f}{\partial x_2}$$

NB: Some people require that $||\boldsymbol{v}|| = 1$. Then, we can identify $D_{\boldsymbol{v}} f(\mathbf{x})$ with the instantaneous rate of change in direction $\boldsymbol{v}$ – and in our example we would have to divide by $\sqrt{2}$.

PROPERTIES OF THE GRADIENT

- **Orthogonal** to level curves/surfaces of a function
- Points in direction of **greatest increase** of f





Proof: Let $\boldsymbol{v}$ be a vector with $\|\boldsymbol{v}\| = 1$ and θ the angle between $\boldsymbol{v}$ and $\nabla f(\mathbf{x})$.

$$D_{\boldsymbol{v}} f(\mathbf{x}) = \nabla f(\mathbf{x})^T \boldsymbol{v} = \|\nabla f(\mathbf{x})\| \, \|\boldsymbol{v}\| \cos(\theta) = \|\nabla f(\mathbf{x})\| \cos(\theta)$$

by the cosine formula for dot products and $\|\boldsymbol{v}\| = 1$. $\cos(\theta)$ is maximal if $\theta = 0$, hence if $\boldsymbol{v}$ and $\nabla f(\mathbf{x})$ point in the same direction.

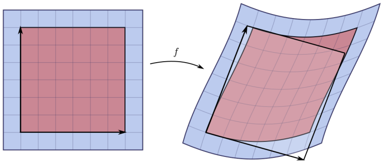
Analogous: Negative gradient $-\nabla f(\mathbf{x})$ points in direction of greatest *de*crease

JACOBIAN MATRIX

For vector-valued function $\mathbf{f} = (f_1, \dots, f_m)^T$, $f_j : \mathcal{S} \to \mathbb{R}$, the **Jacobian** matrix $\mathbf{J}_f : \mathcal{S} \to \mathbb{R}^{m \times d}$ generalizes gradient by placing all ∇f_j in its rows:

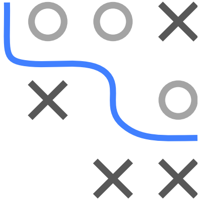
$$\mathbf{J}_f(\mathbf{x}) = \begin{pmatrix} \nabla f_1(\mathbf{x})^T \\ \vdots \\ \nabla f_m(\mathbf{x})^T \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_m(\mathbf{x})}{\partial x_d} \end{pmatrix}$$

- Jacobian gives best linear approximation of distorted volumes



Source: Wikipedia

JACOBIAN DETERMINANT



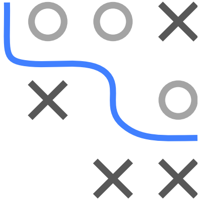
Let $\mathbf{f} \in \mathcal{C}^1$ and $\mathbf{x}_0 \in \mathcal{S}$.

Inverse function theorem: Let $\mathbf{y}_0 = \mathbf{f}(\mathbf{x}_0)$. If $\det(\mathbf{J}_f(\mathbf{x}_0)) \neq 0$, then

1. $\mathbf{f}$ is invertible in a neighborhood of $\mathbf{x}_0$,
2. $\mathbf{f}^{-1} \in \mathcal{C}^1$ with $\mathbf{J}_{f^{-1}}(\mathbf{y}_0) = \mathbf{J}_f(\mathbf{x}_0)^{-1}$.

- $|\det(\mathbf{J}_f(\mathbf{x}_0))|$: factor by which $\mathbf{f}$ expands/shrinks volumes near $\mathbf{x}_0$
- If $\det(\mathbf{J}_f(\mathbf{x}_0)) > 0$, $\mathbf{f}$ preserves orientation near $\mathbf{x}_0$
- If $\det(\mathbf{J}_f(\mathbf{x}_0)) < 0$, $\mathbf{f}$ reverses orientation near $\mathbf{x}_0$

HESSIAN MATRIX



For real-valued function $f : \mathcal{S} \to \mathbb{R}$, the **Hessian** matrix $H : \mathcal{S} \to \mathbb{R}^{d \times d}$ contains all their second derivatives (if they exist):

$$\mathbf{H}(\mathbf{x}) = \nabla^2 \mathbf{f}(\mathbf{x}) = \left(\frac{\partial^2 \mathbf{f}(\mathbf{x})}{\partial x_i \partial x_j} \right)_{i,j=1,\dots,d}$$

Note: Hessian of $\mathbf{f}$ is Jacobian of $\nabla \mathbf{f}$

Example: Let $\mathbf{f}(\mathbf{x}) = \sin(x_1) \cdot \cos(2x_2)$. Then:

$$\mathbf{H}(\mathbf{x}) = \begin{pmatrix} -\cos(2x_2) \cdot \sin(x_1) & -2\cos(x_1) \cdot \sin(2x_2) \\ -2\cos(x_1) \cdot \sin(2x_2) & -4\cos(2x_2) \cdot \sin(x_1) \end{pmatrix}$$

- If $\mathbf{f} \in \mathcal{C}^2$, then H is symmetric
- Many local properties (geometry, convexity, critical points) are encoded by the Hessian and its spectrum ($\to$ later)

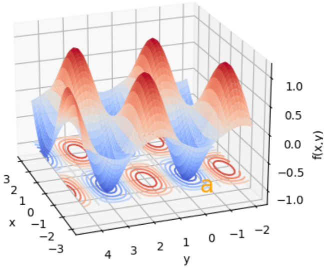
LOCAL CURVATURE BY HESSIAN

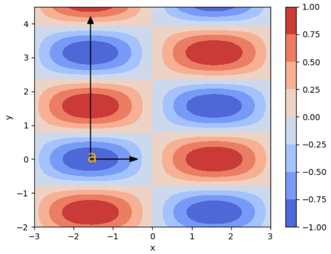
Eigenvector corresponding to largest (resp. smallest) **eigenvalue** of Hessian points in direction of largest (resp. smallest) **curvature**

Example (previous slide)**:** For $\boldsymbol{a} = (-\pi/2, 0)^T$, we have

$$H(\boldsymbol{a}) = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$$

and thus $\lambda_1 = 4$, $\lambda_2 = 1$, $\boldsymbol{v}_1 = (0, 1)^T$, and $\boldsymbol{v}_2 = (1, 0)^T$.





Optimization in Machine Learning

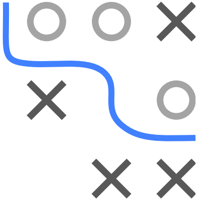
Mathematical Concepts
Matrix Calculus

δ

Learning goals

- Rules of matrix calculus
- Connection of gradient, Jacobian and Hessian

MATRIX-VECTOR-OPERATIONS



- Matrix-Vector-Multiplication

$$\underbrace{\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}}_{\mathbf{b}} = \begin{pmatrix} a_{11}b_1 + \cdots + a_{1n}b_n \\ \vdots \\ a_{m1}b_1 + \cdots + a_{mn}b_n \end{pmatrix}$$

- Transpose $(\mathbf{Ab})^T = \mathbf{b}^T\mathbf{A}^T$, $\left(\mathbf{A}^T\right)^T = \mathbf{A}$
- Symmetry $\mathbf{A}^T = \mathbf{A}$
- Inverse $\mathbf{AA}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$

MATRIX-VECTOR-OPERATIONS / 2

Determinant:

$$det \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{21}a_{12}$$

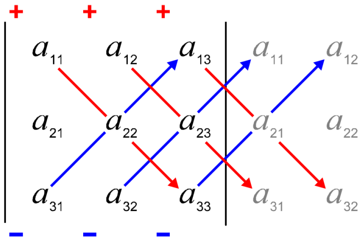


Figure: Only for 3*x*3-matrices, rule of Sarrus
Source: `https://de.wikipedia.org/wiki/Determinante`.

MATRIX-INVERSION

Criterion for invertibility of matrix **A**:

- **A** has to be positive definite

Positive definiteness:

- **A** positive definite, if $\mathbf{x}^T\mathbf{A}\mathbf{x} > 0$ for any non-zero $\mathbf{x} \in \mathbb{R}^n$
- All eigenvalues of **A** are positive and non-zero
- $|\mathbf{A}| > 0$

$\Rightarrow$ the invertibility of a matrix **A** can be tested by calculating its determinant

MATRIX-INVERSION / 2

Inverting a matrix means, we search the entries of $\mathbf{A}^{-1}$ such, that

$$\underbrace{\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} \bar{a}_{11} & \bar{a}_{12} & \bar{a}_{13} \\ \bar{a}_{21} & \bar{a}_{22} & \bar{a}_{23} \\ \bar{a}_{31} & \bar{a}_{32} & \bar{a}_{33} \end{pmatrix}}_{\mathbf{A}^{-1}} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{I}}$$

- We notice: it's all linear operations
- We may exchange rows or premultiply rows with scalars of $\mathbf{A}$ and $\mathbf{I}$ simulataneously without changing the equation
- We may also add rows together without changing the equation

$\Rightarrow$ **elementary row-operations** of the **Gauss-Jordan-algorithm**

MATRIX-INVERSION / 3

Gauss-Jordan-algorithm:

$$\left(\begin{array}{ccc|ccc} a_{11} & a_{12} & a_{13} & 1 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 1 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & 0 & 1 \end{array}\right)$$

Goal: Transform both sides with elementary row-operations such that the left-hand-side becomes the identity matrix. The right-hand-side is then transformed to the inverse matrix:

$$\begin{array}{l} I: \\ II: \\ III: \end{array} \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & \bar{a}_{11} & \bar{a}_{12} & \bar{a}_{13} \\ 0 & 1 & 0 & \bar{a}_{21} & \bar{a}_{22} & \bar{a}_{23} \\ 0 & 0 & 1 & \bar{a}_{31} & \bar{a}_{32} & \bar{a}_{33} \end{array}\right)$$

elementary row-operations:

- Premultiply row *I*, *II*, *III* by a scalar $c \in \mathbb{R}$
- Add row to another row

MATRIX-INVERSION / 4

Example:

- Multiply row *II* with $-\frac{a_{31}}{a_{21}}$ and add it to row *III*

Result:

$$\left(\begin{array}{ccc|ccc} a_{11} & a_{12} & a_{13} & 1 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 1 & 0 \\ \underbrace{a_{31} - \dfrac{a_{31}}{a_{21}} a_{21}}_{=0} & a_{32} - \frac{a_{31}}{a_{21}} a_{22} & a_{33} - \frac{a_{31}}{a_{21}} a_{23} & 0 & -\frac{a_{31}}{a_{21}} & 1 \end{array}\right)$$

Strategy:

- Bring matrix **A** in upper-triangle-form
- Bring upper-triangle-form to diagonal form
- Norm rows to obtain identity matrix **I**

SCOPE

- $\mathcal{X}/\mathcal{Y}$ denote space of **independent**/**dependent** variables
- Identify dependent variable with a **function** $y : \mathcal{X} \to \mathcal{Y}, x \mapsto y(x)$
- Assume y sufficiently smooth
- In matrix calculus, x and y can be **scalars**, **vectors**, or **matrices**:

Type	scalar x	vector $\mathbf{x}$	matrix $\mathbf{X}$
scalar y	$\partial y/\partial x$	$\partial y/\partial \mathbf{x}$	$\partial y/\partial \mathbf{X}$
vector $\mathbf{y}$	$\partial \mathbf{y}/\partial x$	$\partial \mathbf{y}/\partial \mathbf{x}$	–
matrix $\mathbf{Y}$	$\partial \mathbf{Y}/\partial x$	–	–

- We denote vectors/matrices in **bold** lowercase/uppercase letters

NUMERATOR LAYOUT

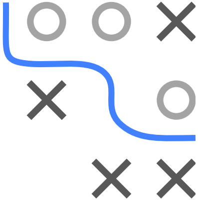
- **Matrix calculus:** collect derivative of each component of dependent variable w.r.t. each component of independent variable
- We use so-called **numerator layout** convention:

$$\frac{\partial y}{\partial \mathbf{x}} = \left(\frac{\partial y}{\partial x_1}, \cdots, \frac{\partial y}{\partial x_d}\right) = \nabla y^T \in \mathbb{R}^{1\times d}$$

$$\frac{\partial \mathbf{y}}{\partial x} = \left(\frac{\partial y_1}{\partial x}, \cdots, \frac{\partial y_m}{\partial x}\right)^T \in \mathbb{R}^{m}$$

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial y_1}{\partial \mathbf{x}} \\ \vdots \\ \frac{\partial y_m}{\partial \mathbf{x}} \end{pmatrix} = \left(\frac{\partial \mathbf{y}}{\partial x_1} \cdots \frac{\partial \mathbf{y}}{\partial x_d}\right) = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_d} \end{pmatrix} = J_{\mathbf{y}} \in \mathbb{R}^{m\times d}$$

SCALAR-BY-VECTOR



Let $\mathbf{x} \in \mathbb{R}^d$, $y, z : \mathbb{R}^d \to \mathbb{R}$ and $\mathbf{A}$ be a matrix.

- If y is a **constant** function: $\frac{\partial y}{\partial \mathbf{x}} = \mathbf{0}^T \in \mathbb{R}^{1 \times d}$
- **Linearity**: $\frac{\partial (a \cdot y + z)}{\partial \mathbf{x}} = a \frac{\partial y}{\partial \mathbf{x}} + \frac{\partial z}{\partial \mathbf{x}} \quad$ (a constant)
- **Product** rule: $\frac{\partial (y \cdot z)}{\partial \mathbf{x}} = y \frac{\partial z}{\partial \mathbf{x}} + \frac{\partial y}{\partial \mathbf{x}} z$
- **Chain** rule: $\frac{\partial g(y)}{\partial \mathbf{x}} = \frac{\partial g(y)}{\partial y} \frac{\partial y}{\partial \mathbf{x}} \quad$ (g scalar-valued function)
- **Second** derivative: $\frac{\partial^2 y}{\partial \mathbf{x} \partial \mathbf{x}^T} = \nabla^2 y^T$ $(= \nabla^2 y$ if $y \in \mathcal{C}^2)$ (Hessian)
- $\frac{\partial (\mathbf{x}^T \mathbf{A} \mathbf{x})}{\partial \mathbf{x}} = \mathbf{x}^T (\mathbf{A} + \mathbf{A}^T)$
- $\frac{\partial (\mathbf{y}^T \mathbf{A} \mathbf{z})}{\partial \mathbf{x}} = \mathbf{y}^T \mathbf{A} \frac{\partial \mathbf{z}}{\partial \mathbf{x}} + \mathbf{z}^T \mathbf{A}^T \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \quad$ ($\mathbf{y}$, $\mathbf{z}$ vector-valued functions of $\mathbf{x}$)

VECTOR-BY-SCALAR

Let $x \in \mathbb{R}$ and $\mathbf{y}, \mathbf{z} : \mathbb{R} \to \mathbb{R}^m$.

- If $\mathbf{y}$ is a **constant** function: $\frac{\partial \mathbf{y}}{\partial x} = \mathbf{0} \in \mathbb{R}^m$
- **Linearity**: $\frac{\partial (a \cdot \mathbf{y} + \mathbf{z})}{\partial x} = a \frac{\partial \mathbf{y}}{\partial x} + \frac{\partial \mathbf{z}}{\partial x}$ (a constant)
- **Chain** rule: $\frac{\partial \mathbf{g}(\mathbf{y})}{\partial x} = \frac{\partial \mathbf{g}(\mathbf{y})}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial x}$ ($\mathbf{g}$ vector-valued function)
- $\frac{\partial (\mathbf{A}\mathbf{y})}{\partial x} = \mathbf{A} \frac{\partial \mathbf{y}}{\partial x}$ ($\mathbf{A}$ matrix)

VECTOR-BY-VECTOR

Let $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{y}, \mathbf{z} : \mathbb{R}^d \to \mathbb{R}^m$.

- If $\mathbf{y}$ is a **constant** function: $\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \mathbf{0} \in \mathbb{R}^{m \times d}$
- $\frac{\partial \mathbf{x}}{\partial \mathbf{x}} = \mathbf{I} \in \mathbb{R}^{d \times d}$
- **Linearity**: $\frac{\partial (a \cdot \mathbf{y} + \mathbf{z})}{\partial \mathbf{x}} = a \frac{\partial \mathbf{y}}{\partial \mathbf{x}} + \frac{\partial \mathbf{z}}{\partial \mathbf{x}}$ (a constant)
- **Chain** rule: $\frac{\partial \mathbf{g}(\mathbf{y})}{\partial \mathbf{x}} = \frac{\partial \mathbf{g}(\mathbf{y})}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{x}}$ ($\mathbf{g}$ vector-valued function)
- $\frac{\partial (\mathbf{A}\mathbf{x})}{\partial \mathbf{x}} = \mathbf{A}$, $\frac{\partial (\mathbf{x}^T \mathbf{B})}{\partial \mathbf{x}} = \mathbf{B}^T$ ($\mathbf{A}, \mathbf{B}$ matrices)

EXAMPLE

Consider $f : \mathbb{R}^2 \to \mathbb{R}$ with

$$f(\mathbf{x}) = \exp\left(-(\mathbf{x}-\mathbf{c})^T \mathbf{A}(\mathbf{x}-\mathbf{c})\right),$$

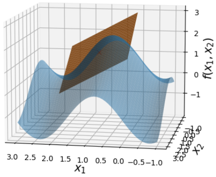
where $\mathbf{c} = (1,1)^T$ and $\mathbf{A} = \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix}$.

Compute $\nabla f(\mathbf{x})$ at $\mathbf{x}^* = \mathbf{0}$:

1. Write $f(\mathbf{x}) = \exp(g(\mathbf{u}(\mathbf{x})))$ with $g(\mathbf{u}) = -\mathbf{u}^T\mathbf{A}\mathbf{u}$ and $\mathbf{u}(\mathbf{x}) = \mathbf{x} - \mathbf{c}$
2. **Chain** rule: $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \exp(g(\mathbf{u}(\mathbf{x})))\frac{\partial g(\mathbf{u})}{\partial \mathbf{u}}\frac{\partial \mathbf{u}(\mathbf{x})}{\partial \mathbf{x}}$
3. $\mathbf{u}^* := \mathbf{u}(\mathbf{x}^*) = (-1,-1)^T$, $g(\mathbf{u}^*) = -3$
4. $\frac{\partial g(\mathbf{u})}{\partial \mathbf{u}} = -2\mathbf{u}^T\mathbf{A}$, $\frac{\partial g(\mathbf{u}^*)}{\partial \mathbf{u}} = (3,3)$
5. **Linearity**: $\frac{\partial \mathbf{u}(\mathbf{x})}{\partial \mathbf{x}} = \frac{\partial(\mathbf{x}-\mathbf{c})}{\partial \mathbf{x}} = \frac{\partial \mathbf{x}}{\partial \mathbf{x}} - \frac{\partial \mathbf{c}}{\partial \mathbf{x}} = \mathbf{I}_2$
6. $\nabla f(\mathbf{x}^*) = \frac{\partial f(\mathbf{x}^*)}{\partial \mathbf{x}}^T = (\exp(-3)\cdot(3,3)\cdot\mathbf{I}_2)^T = \exp(-3)\begin{pmatrix}3\\3\end{pmatrix}$

Optimization in Machine Learning

Mathematical Concepts
Taylor Approximation

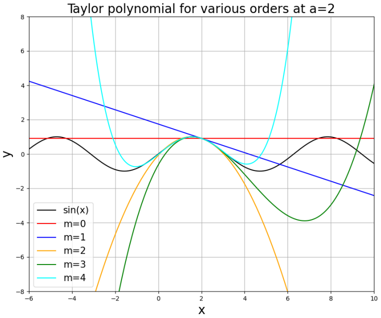


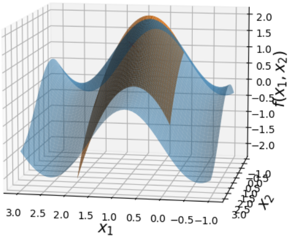
Learning goals

- Taylor’s theorem (univariate)
- Taylor series (univariate)
- Taylor’s theorem (multivariate)
- Taylor series (multivariate)

TAYLOR APPROXIMATIONS

- Mathematically fascinating: **Globally** approximate function by sum of polynomials determined by **local** properties
- Extremely important for **analyzing** optimization algorithms
- Geometry of **linear** and **quadratic** functions very well understood
  $\Longrightarrow$ use them for **approximations**





TAYLOR'S THEOREM (UNIVARIATE)

Taylor's theorem: Let $I \subseteq \mathbb{R}$ be an open interval and $f \in \mathcal{C}^k(I, \mathbb{R})$. For each $a, x \in I$, it holds that

$$f(x) = \underbrace{\sum_{j=0}^{k} \frac{f^{(j)}(a)}{j!}(x-a)^j}_{T_k(x,a)} + R_k(x, a)$$

with the k-th **Taylor polynomial** T_k and a **remainder term**

$$R_k(x, a) = o(|x-a|^k) \quad \text{as } x \to a.$$

- There are explicit formulas for the remainder
- Wording: We "expand f via Taylor around a"

TAYLOR SERIES (UNIVARIATE)

- If $f \in C^\infty$, it *might* be expandable around $a \in I$ as a **Taylor series**

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(a)}{k!}(x-a)^k$$

- If Taylor series converges to f in an interval $I_0 \subseteq I$ centered at a (does not have to), we call f an *analytic function*
- Convergence if $R_k(x, a) \to 0$ as $k \to \infty$ for all $x \in I_0$

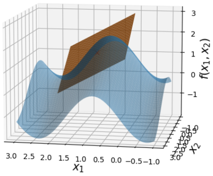
TAYLOR'S THEOREM (MULTIVARIATE)

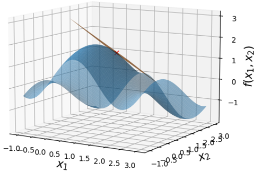
Taylor's theorem (1st order): For $f \in \mathcal{C}^1$, it holds that

$$\mathbf{f}(\mathbf{x}) = \underbrace{\mathbf{f}(\boldsymbol{a}) + \nabla \mathbf{f}(\boldsymbol{a})^T(\mathbf{x} - \boldsymbol{a})}_{T_1(\mathbf{x}, \boldsymbol{a})} + R_1(\mathbf{x}, \boldsymbol{a}).$$

Example: $\mathbf{f}(\mathbf{x}) = \sin(2x_1) + \cos(x_2)$, $\boldsymbol{a} = (1, 1)^T$. Since $\nabla \mathbf{f}(\mathbf{x}) = \begin{pmatrix} 2\cos(2x_1) \\ -\sin(x_2) \end{pmatrix}$,

$$\begin{aligned} \mathbf{f}(\mathbf{x}) = T_1(\mathbf{x}) + R_1(\mathbf{x}, \boldsymbol{a}) &= \mathbf{f}(\boldsymbol{a}) + \nabla \mathbf{f}(\boldsymbol{a})^T(\mathbf{x} - \boldsymbol{a}) + R_1(\mathbf{x}, \boldsymbol{a}) \\ &= \sin(2) + \cos(1) + (2\cos(2), -\sin(1))^T \begin{pmatrix} x_1 - 1 \\ x_2 - 1 \end{pmatrix} + R_1(\mathbf{x}, \boldsymbol{a}) \end{aligned}$$





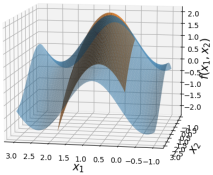
TAYLOR'S THEOREM (MULTIVARIATE) / 2

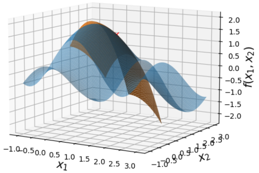
Taylor's theorem (2nd order): If $f \in \mathcal{C}^2$, it holds that

$$f(\mathbf{x}) = \underbrace{f(\boldsymbol{a}) + \nabla f(\boldsymbol{a})^T(\mathbf{x} - \boldsymbol{a}) + \frac{1}{2}(\mathbf{x} - \boldsymbol{a})^T \boldsymbol{H}(\boldsymbol{a})(\mathbf{x} - \boldsymbol{a})}_{T_2(\mathbf{x},\boldsymbol{a})} + R_2(\mathbf{x}, \boldsymbol{a})$$

Example (continued): Since $H(\mathbf{x}) = \begin{pmatrix} -4\sin(2x_1) & 0 \\ 0 & -\cos(x_2) \end{pmatrix}$,

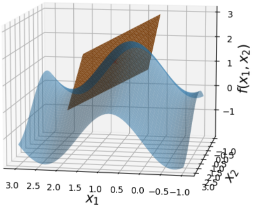
$$f(\mathbf{x}) = T_1(\mathbf{x}, \boldsymbol{a}) + \frac{1}{2}\begin{pmatrix} x_1 - 1 \\ x_2 - 1 \end{pmatrix}^T \begin{pmatrix} -4\sin(2) & 0 \\ 0 & -\cos(1) \end{pmatrix} \begin{pmatrix} x_1 - 1 \\ x_2 - 1 \end{pmatrix} + R_2(\mathbf{x}, \boldsymbol{a})$$

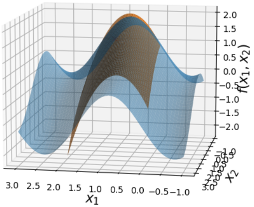




MULTIVARIATE TAYLOR APPROXIMATION

- Higher order k gives a better approximation
- $T_k(\mathbf{x}, \boldsymbol{a})$ is the best k-th order approximation to $f(\mathbf{x})$ near $\boldsymbol{a}$





Consider $T_2(\mathbf{x}, \boldsymbol{a}) = f(\boldsymbol{a}) + \nabla f(\boldsymbol{a})^T(\mathbf{x} - \boldsymbol{a}) + \frac{1}{2}(\mathbf{x} - \boldsymbol{a})^T H(\boldsymbol{a})(\mathbf{x} - \boldsymbol{a})$.
The first/second/third term ensures the values/slopes/curvatures of T_2 and f match at $\boldsymbol{a}$.

TAYLOR'S THEOREM (MULTIVARIATE)

The theorem for general order k requires a more involved notation.

Taylor's theorem (k-th order): If $f \in \mathcal{C}^k$, it holds that

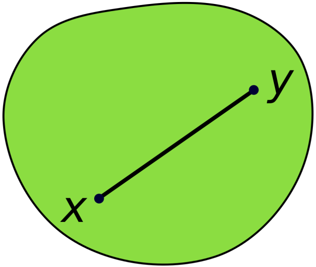
$$f(\mathbf{x}) = \underbrace{\sum_{j=1}^{k} \frac{D^j f(\boldsymbol{a})}{j!} (\mathbf{x} - \boldsymbol{a})^j}_{T_k(\mathbf{x}, \boldsymbol{a})} + R_k(\mathbf{x}, \boldsymbol{a})$$

with $R_k(\mathbf{x}, \boldsymbol{a}) = o(\|\mathbf{x} - \boldsymbol{a}\|^k)$ as $\mathbf{x} \to \boldsymbol{a}$,

$D^j f = \frac{\partial^j f}{\partial^j x_1} \cdots \frac{\partial^j f}{\partial^j x_d}$

Optimization in Machine Learning

Mathematical Concepts
Convexity



Learning goals

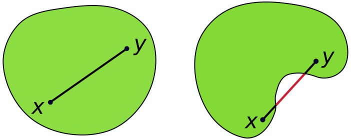
- Convex sets
- Convex functions

CONVEX SETS

A set of $\mathcal{S} \subseteq \mathbb{R}^d$ is **convex**, if for all $\mathbf{x}, \mathbf{y} \in \mathcal{S}$ and all $t \in [0, 1]$ the following holds:

$$\mathbf{x} + t(\mathbf{y} - \mathbf{x}) \in \mathcal{S}$$

Intuitively: Connecting line between any $\mathbf{x}, \boldsymbol{y} \in \mathcal{S}$ lies completely in $\mathcal{S}$.



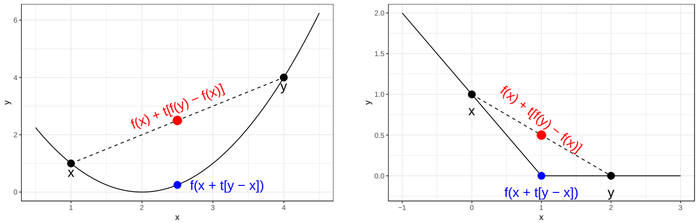
Left: convex set. **Right:** not convex. (Source: Wikipedia)

CONVEX FUNCTIONS

Let $f : \mathcal{S} \to \mathbb{R}$, $\mathcal{S}$ convex. f is **convex** if for all $\mathbf{x}, \mathbf{y} \in \mathcal{S}$ and all $t \in [0, 1]$

$$f(\mathbf{x} + t(\mathbf{y} - \mathbf{x})) \leq f(\mathbf{x}) + t(f(\mathbf{y}) - f(\mathbf{x})).$$

Intuitively: Connecting line lies above function.



Left: Strictly convex function. **Right:** Convex, but not strictly.

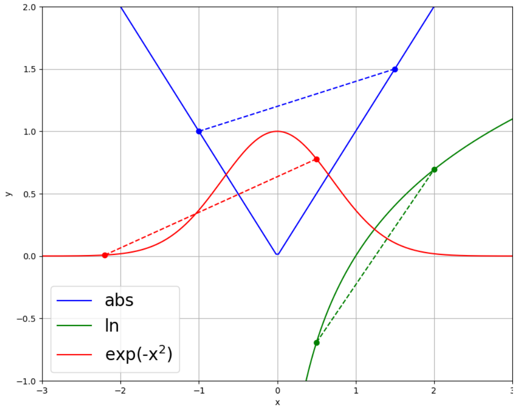
Strictly convex if “$<$” instead of “$\leq$”. **Concave** (strictly) if the inequality holds with “$\geq$” (“$>$”), respectively.

Note: f (strictly) concave $\Leftrightarrow -f$ (strictly) convex.

EXAMPLES

Convex function: $f(x) = |x|$
Concave function: $f(x) = \log(x)$

Neither nor: $f(x) = \exp(-x^2)$ (but log-concave)



OPERATIONS PRESERVING CONVEXITY

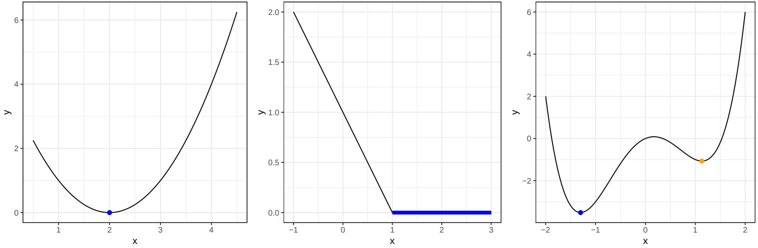
- **Nonnegatively weighted summation:** Weights $w_1, \dots, w_n \geq 0$, convex functions $f_1, \dots, f_n$: $w_1 f_1 + \cdots + w_n f_n$ also convex
  In particular: Sum of convex functions also convex
- **Composition:** g convex, f linear: $h = g \circ f$ also convex
  Proof:

$$
\begin{aligned}
h(\mathbf{x} + t(\mathbf{y} - \mathbf{x})) &= g(f(\mathbf{x} + t(\mathbf{y} - \mathbf{x}))) \\
&= g(f(\mathbf{x}) + t(f(\mathbf{y}) - f(\mathbf{x}))) \\
&\leq g(f(\mathbf{x})) + t(g(f(\mathbf{y})) - g(f(\mathbf{x}))) \\
&= h(\mathbf{x}) + t(h(\mathbf{y}) - h(\mathbf{x}))
\end{aligned}
$$

- **Elementwise maximization:** $f_1, \dots, f_n$ convex functions:
  $g(\mathbf{x}) = \max\{f_1(\mathbf{x}), \dots, f_n(\mathbf{x})\}$ also convex

CONVEX FUNCTIONS IN OPTIMIZATION

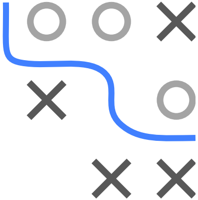
- For a convex function, every local optimum is also a global one
  $\Rightarrow$ No need for involved global optimizers, local ones are enough
- A strictly convex function has at most one optimal point
- Example for strictly convex function without optimum: $\exp$ on $\mathbb{R}$

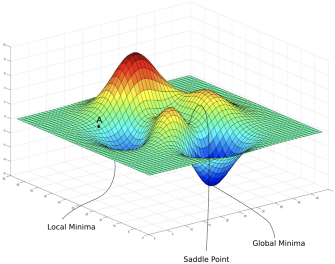


Left: Strictly convex; exactly one local minimum, which is also global. **Middle:** Convex, but not strictly; all local optima are also global ones but not unique. **Right:** Not convex.

Optimization in Machine Learning

Mathematical Concepts
Conditions for optimality





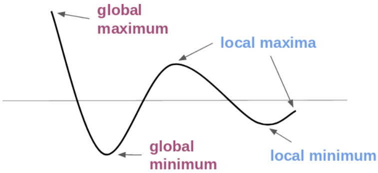
Learning goals

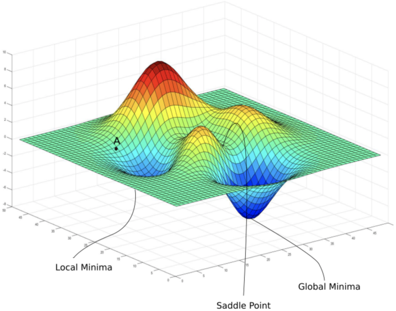
- Local and global optima
- First & second order conditions

DEFINITION LOCAL AND GLOBAL MINIMUM

Given $\mathcal{S} \subseteq \mathbb{R}^d$, $f : \mathcal{S} \to \mathbb{R}$:

- f has **global minimum** in $\mathbf{x}^* \in \mathcal{S}$, if $f(\mathbf{x}^*) \leq f(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{S}$
- f has a **local minimum** in $\mathbf{x}^* \in \mathcal{S}$, if $\epsilon > 0$ exists s.t. $f(\mathbf{x}^*) \leq f(\mathbf{x})$ for all $\mathbf{x} \in B_\epsilon(\mathbf{x}^*)$ ("ϵ"-ball around $\mathbf{x}^*$).





Source (**left**): `https://en.wikipedia.org/wiki/Maxima_and_minima`.

Source (**right**): `https://wngaw.github.io/linear-regression/`.

EXISTENCE OF OPTIMA

We regard the two main cases of $f : \mathcal{S} \to \mathbb{R}$:

- f **continuous**: If $\mathcal{S}$ is **compact**, f attains a minimum and a maximum (extreme value theorem).
- f **discontinuous**: **No general** statement possible about existence of optima.

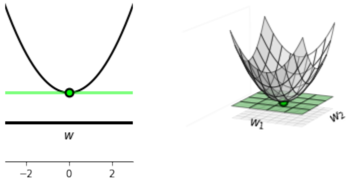
Example: $\mathcal{S} = [0, 1]$ compact, f discontinuous with

$$f(x) = \begin{cases} 1/x & \text{if } x > 0, \\ 0 & \text{if } x = 0. \end{cases}$$

FIRST ORDER CONDITION FOR OPTIMALITY

Observation: At an interior local optimum of $f \in \mathcal{C}^1$, first order Taylor approximation is flat, i.e., first order derivatives are zero.

This condition is therefore **necessary** and called **first order**.



Strictly convex functions (**left:** univariate, **right:** multivariate) with unique local minimum, which is the global one. Tangent (hyperplane) is perfectly flat at the optimum. (Source: Watt, *Machine Learning Refined*, 2020)

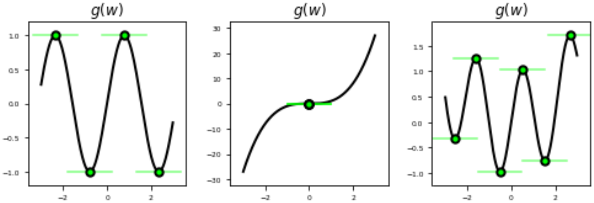
FIRST ORDER CONDITION FOR OPTIMALITY / 2

First order condition: Gradient of f at local optimum $\mathbf{x}^* \in \mathcal{S}$ is zero:

$$\nabla f(\mathbf{x}^*) = (0, \ldots, 0)^T$$

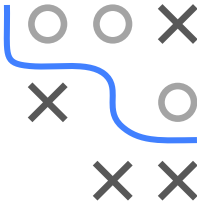
Points with zero first order derivative are called **stationary.**

Condition is **not sufficient**: Not all stationary points are local optima.



Left: Four points fulfill the necessary condition and are indeed optima.
Middle: One point fulfills the necessary condition but is not a local optimum.
Right: Multiple local minima and maxima.
(Source: Watt, 2020, Machine Learning Refined)

SECOND ORDER CONDITION FOR OPTIMALITY



Second order condition: Hessian of $f \in \mathcal{C}^2$ at stationary point $\mathbf{x}^* \in \mathcal{S}$ is positive or negative definite:

$$H(\mathbf{x}^*) \succ 0 \text{ or } H(\mathbf{x}^*) \prec 0$$

Interpretation: Curvature of f at local optimum is either positive in all directions or negative in all directions.

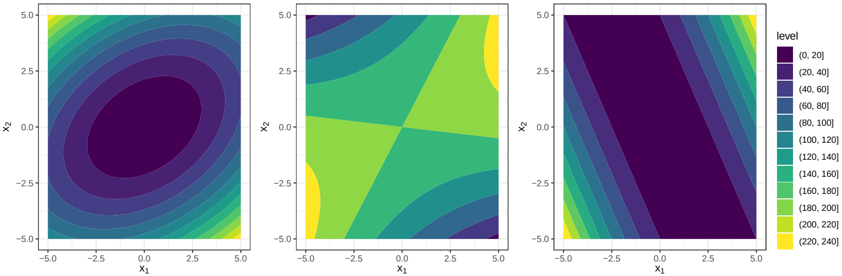
The second order condition is **sufficient** for a stationary point.

Proof: Later.

CONDITIONS FOR OPTIMALITY AND CONVEXITY

Let $f : \mathcal{S} \to \mathbb{R}$ be **convex**. Then:

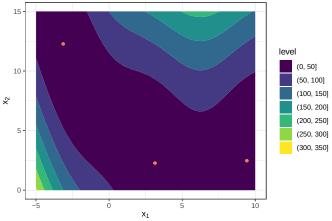
- Any local minimum is **also global** minimum
- If f **strictly convex**, f has **at most one** local minimum which would also be unique global minimum on $\mathcal{S}$

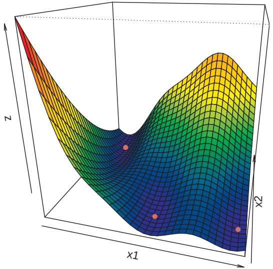


Three quadratic forms. **Left:** $H(\mathbf{x}^*)$ has two positive eigenvalues. **Middle:** $H(\mathbf{x}^*)$ has positive and negative eigenvalue. **Right:** $H(\mathbf{x}^*)$ has positive and a zero eigenvalue.

CONDITIONS FOR OPTIMALITY AND CONVEXITY / 2

Example: Branin function





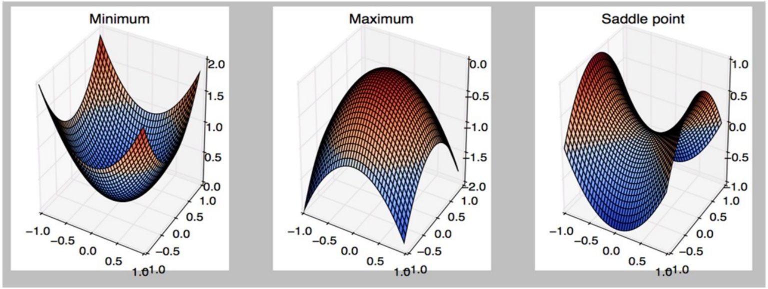
Spectra of Hessians (numerically computed):

	λ_1	λ_2
Left	22.29	0.96
Middle	11.07	1.73
Right	11.33	1.69

CONDITIONS FOR OPTIMALITY AND CONVEXITY / 3

Definition: **Saddle point** at $\mathbf{x}$

- $\mathbf{x}$ stationary (necessary)
- $H(\mathbf{x})$ indefinite, i.e., positive and negative eigenvalues (sufficient)



CONDITIONS FOR OPTIMALITY AND CONVEXITY / 4

Examples:

- $f(x, y) = x^2 - y^2$, $\nabla f(x, y) = (2x, -2y)^T$,
  $$H_f(x, y) = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}$$
  $\implies$ Saddle point at $(0, 0)$ (sufficient condition met)

- $g(x, y) = x^4 - y^4$, $\nabla g(x, y) = (4x^3, -4y^3)^T$,
  $$H_g(x, y) = \begin{pmatrix} 12x^2 & 0 \\ 0 & -12y^2 \end{pmatrix}$$
  $\implies$ Saddle point at $(0, 0)$ (sufficient condition **not** met)