Fundamentals of Mathematics for Machine Learning

Oxford ML Summer School
8 May 2023

Rasul Tutunov
Juliusz Ziomek

Plan of the Talk

Linear algebra

Vector Calculus

Probability Theory

Loss functions in ML

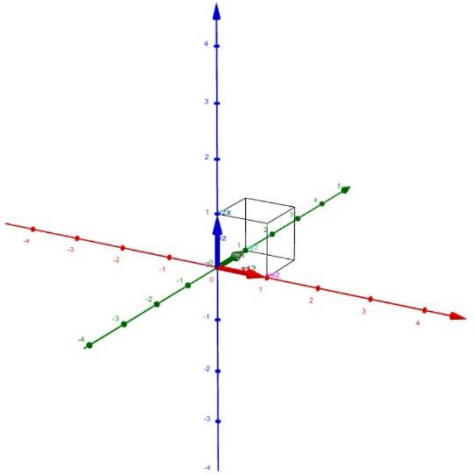
Collection of 511 American Mathematical Textbooks, 1760–1850

Courtesy of Robin Halwas

Linear Algebra

Vectors in Euclidian Spaces.

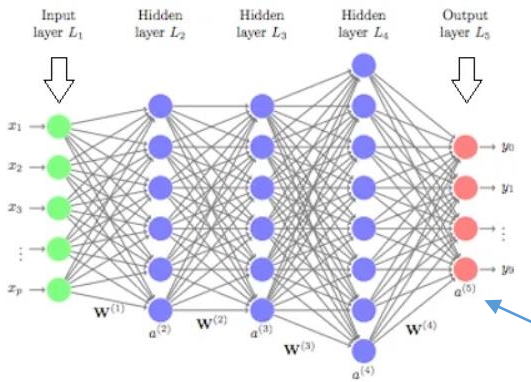
- Definition, properties, and examples.
- Length, inner products, norms, metrics.
- Linear Independence.



Matrices

- Definition, properties, and examples.
- Trace, inner products, norms, metrics.
- Symmetric matrices, positive semidefinite matrices.
- Eigenvalues, eigenvalue decomposition.
- Cholesky Decomposition.

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \\ m \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & \dots & n \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \end{array}$$



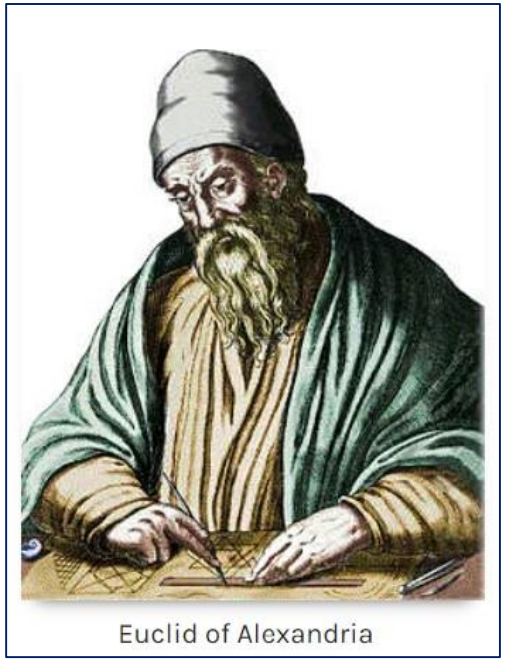
Courtesy of Vishal Yadav

Euclidian Space

Euclidian Space:

A space in any finite number of dimensions *n*, in which points are designated by ***coordinates*** (one for each dimension) and the distance between two points is given by a ***distance formula***.

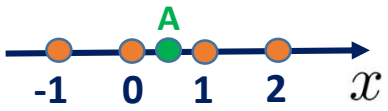
Typical notation: $\mathbb{R}^n$


Euclid of Alexandria

Coordinates of Points

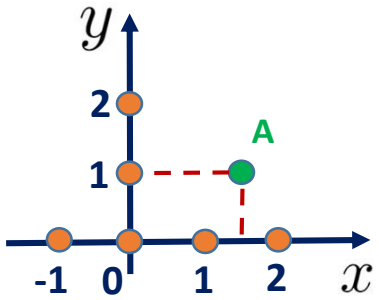
An **ordered collection** of *n* numbers, identifying the location of points in space with respect to predefined reference lines (called **coordinate axes**)

Examples:



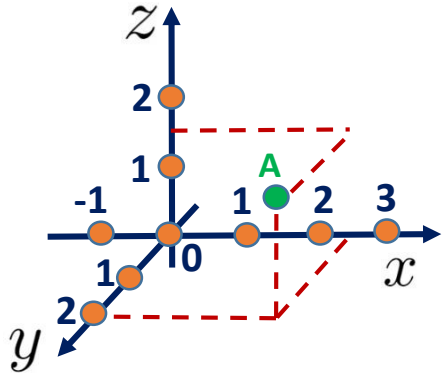
A = 0.5

$\mathbb{R}^1$



A = (1.5, 1)

$\mathbb{R}^2$



A = (2.5, 2, 1.5)

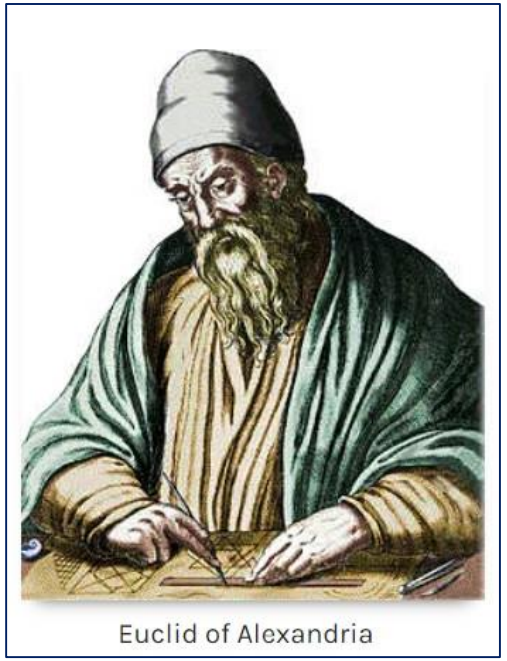
$\mathbb{R}^3$

Euclidian Space

Euclidian Space:

A space in any finite number of dimensions *n*, in which points are designated by ***coordinates*** (one for each dimension) and the distance between two points is given by a ***distance formula***.

Typical notation: $\mathbb{R}^n$

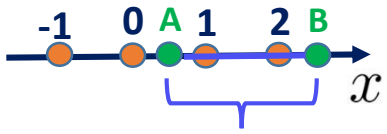

Euclid of Alexandria

Distance Formula

Algebraic expression that gives the distances between pairs of points in terms of their coordinates

Examples:

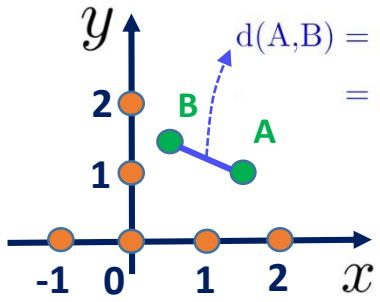
A = 0.5 B = 2.5



$$d(A,B) = |A - B| = 2$$

$\mathbb{R}^1$

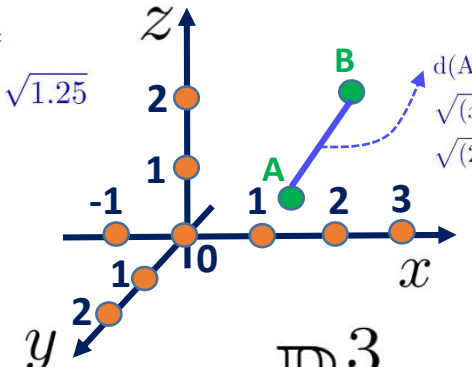
A = (1.5, 1) B = (0.5, 1.5)



$$d(A,B) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} = \sqrt{(1.5 - 0.5)^2 + (1 - 1.5)^2} = \sqrt{1.25}$$

$\mathbb{R}^2$

A = (2.5, 2, 1.5) B = (3, 1, 2)



$$d(A,B) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2} = \sqrt{(2.5 - 3)^2 + (2 - 1)^2 + (1.5 - 2)^2} = \sqrt{1.5}$$

$\mathbb{R}^3$

Euclidian Spaces

In *n*-dimensional Euclidian space $\mathbb{R}^n$, each point is designated by *n* coordinates: $\mathrm{A} = (x_1^{(A)}, x_2^{(A)}, \ldots, x_n^{(A)})$

Distance (Euclidian) between two points $\mathrm{A} = (x_1^{(A)}, x_2^{(A)}, \ldots, x_n^{(A)})$ and $\mathrm{B} = (x_1^{(B)}, x_2^{(B)}, \ldots, x_n^{(B)})$ is defined as:

$$\mathrm{d(A,B)} = \sqrt{\sum_{i=1}^{n} \left(x_i^{(A)} - x_i^{(B)} \right)^2}$$

Remark:

Distance function $\mathrm{d(A,B)}$ in Euclidian space, can be defined in many ways. Actually any function $\rho(\mathrm{A,B})$ satisfying the following **metric axioms** is a valid distance:

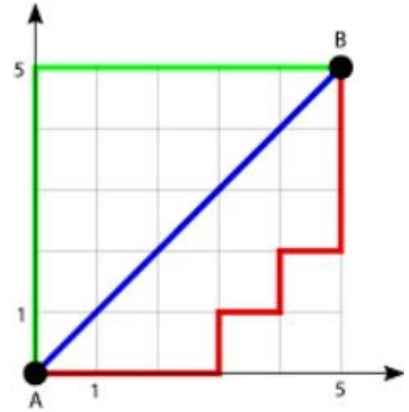
Example:

- For any distinct points A and B it is positive: $\rho(\mathrm{A,B}) > 0$
- For any point A: $\rho(\mathrm{A},$

Manhattan distance:

$$\mathrm{d(A,B)} = \sum_{i=1}^{n} \left| x_i^{(A)} - x_i^{(B)} \right| \mathrm{A})$$

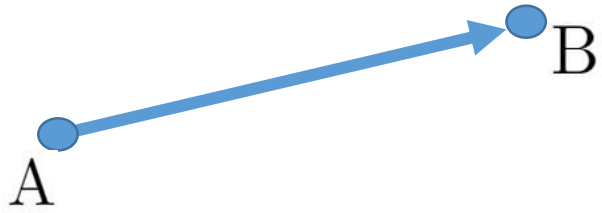
- For any points A and B it i
- For any points A, B and C we have triangular inequality: $\rho(\mathrm{A,C}) \leq \rho(\mathrm{A}$



Vectors in Euclidian Spaces

Vector in Euclidian space $\mathbb{R}^n$ is a geometric object that has ***magnitude*** and ***direction***. It characterized by two points:

the beginning point $\mathrm{A} = (a_1, \ldots, a_n)$ and the end point $\mathrm{B} = (b_1, \ldots, b_n)$



$$\overrightarrow{\mathrm{AB}} = \begin{bmatrix} b_1 - a_1 \\ b_2 - a_2 \\ \vdots \\ b_n - a_n \end{bmatrix}$$

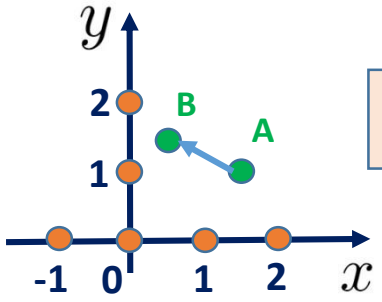
components/coordinates of $\overrightarrow{\mathrm{AB}}$

Magnitude of vector $\overrightarrow{\mathrm{AB}}$ is its length. Typically, is defined via distance measure between points A and B

Examples:

A = (1.5, 1) B = (0.5, 1.5)

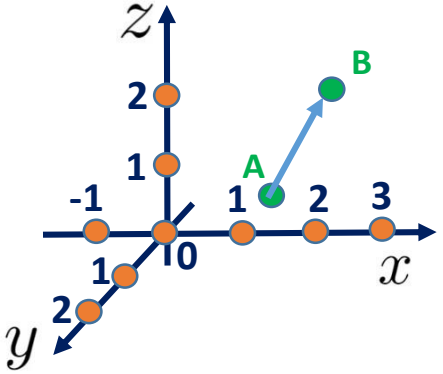
$\mathbb{R}^2$



$$\overrightarrow{\mathrm{AB}} = \begin{bmatrix} 0.5 - 1.5 \\ 1.5 - 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0.5 \end{bmatrix}$$

A = (2.5, 2, 1.5) B = (3, 1, 2)

$\mathbb{R}^3$



$$\overrightarrow{\mathrm{AB}} = \begin{bmatrix} 3 - 2.5 \\ 1 - 2 \\ 2 - 1.5 \end{bmatrix} = \begin{bmatrix} 0.5 \\ -1 \\ 0.5 \end{bmatrix}$$

Vectors in Euclidian Spaces

Typically, the beginning point A of vector $\overrightarrow{\mathrm{AB}}$ is chosen to be $\mathrm{A} = (0, \ldots, 0)$. Hence:

$$\overrightarrow{\mathrm{AB}} = \overrightarrow{\mathrm{B}} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}}_{\text{column vector representation}} = \underbrace{[b_1, b_2, \ldots, b_n]}_{\text{row vector representation}}^{\mathsf{T}}$$

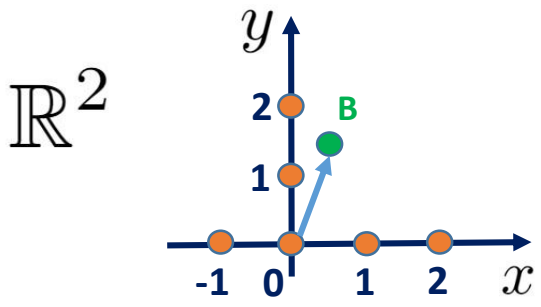
Transpose operation, allows to write row vector as column vector

Typical notation for vectors uses bold lowercase letters:

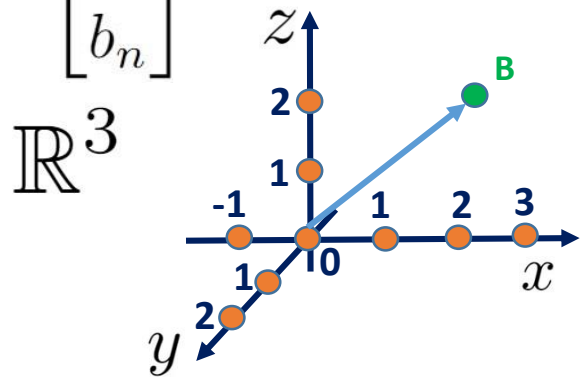
$$\boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Examples:

B = (0.5, 1.5)



B = (3, 1, 2)



Important Operations on Vectors

Linear Operations:

$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}, \quad \alpha\boldsymbol{v} + \beta\boldsymbol{u} = \begin{bmatrix} \alpha v_1 + \beta u_1 \\ \alpha v_2 + \beta u_2 \\ \vdots \\ \alpha v_n + \beta u_n \end{bmatrix}$$

scalars
(any real numbers)

both $\boldsymbol{v}$ and $\boldsymbol{u}$ must be the same dimensionality

Geometric interpretation:

Examples:

$$\boldsymbol{v} = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} 3 \\ 9 \\ 15 \end{bmatrix}, \quad 2\boldsymbol{v} + 0.5\boldsymbol{u} = \begin{bmatrix} 2 \cdot 1 + 0.5 \cdot 3 \\ 2 \cdot 2 + 0.5 \cdot 9 \\ 2 \cdot 5 + 0.5 \cdot 15 \end{bmatrix} = \begin{bmatrix} 3.5 \\ 8.5 \\ 17.5 \end{bmatrix}$$

$\boldsymbol{u}$

$\beta\boldsymbol{u}$

$0 < \beta < 1$

Important Operations on Vectors

Inner product:

$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}, \quad < u, v >= \boldsymbol{v}^\mathsf{T}\boldsymbol{u} = \underbrace{v_1u_1 + v_2u_2 + \ldots v_nu_n}$$

Result of this product
is **NOT** a vector, but a scalar.

both $\boldsymbol{v}$ and $\boldsymbol{u}$ must be the same dimensionality

Application in ML:

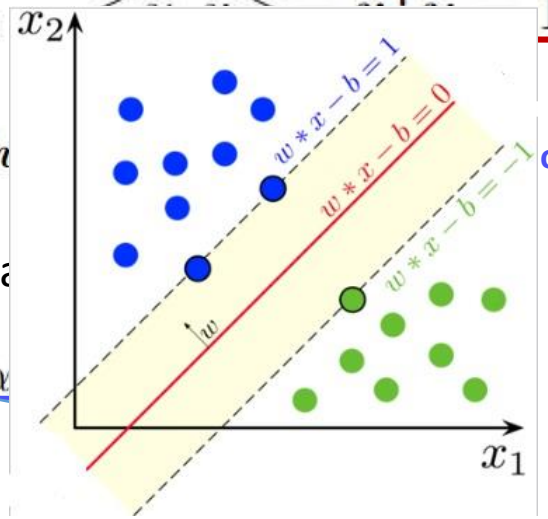
Basic properties:

Maximal margin hyperplane for support vector machines has form: $\boldsymbol{w}^\mathsf{T}\boldsymbol{x} - b = 0$

$$\boldsymbol{v} = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} 3 \\ -1 \\ 0.5 \end{bmatrix}, \quad 1 \cdot 3 + 2 \cdot (-1) + 5 \cdot 0.5 = 3.5$$

parameters

- Multiplication on itself: $\boldsymbol{v}^\mathsf{T}\boldsymbol{v}$ of the vector
- Orthogonality of vectors $\boldsymbol{u}$ a
- Linearity: $(\alpha\boldsymbol{v} + \beta\boldsymbol{u})^\mathsf{T}\boldsymbol{w} = \alpha$



Important Operations on Vectors

Component wise-product:

$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}, \quad \boldsymbol{v} \odot \boldsymbol{u} = \begin{bmatrix} v_1 \cdot u_1 \\ v_2 \cdot u_2 \\ \vdots \\ v_n \cdot u_n \end{bmatrix}$$

both $\boldsymbol{v}$ and $\boldsymbol{u}$ must be the same dimensionality

Result of this product is a vector.

Application in ML:

Examples:

ADAM update equations:

$$\boldsymbol{v} = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}, \quad \boldsymbol{u} \begin{bmatrix} \vdots \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \begin{bmatrix} 1 \cdot 3 \\ \cdot (-1)) \\ 5 \cdot 0.2 \end{bmatrix} = \begin{bmatrix} 3 \\ -2 \\ 1 \end{bmatrix}$$

$$\boldsymbol{m}_t = \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{g}_t$$

$$\boldsymbol{v}_t = \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)\boldsymbol{g}_t \odot \boldsymbol{g}_t$$

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \alpha \frac{\boldsymbol{m}_t}{\sqrt{\boldsymbol{v}_t} + \epsilon \mathbf{1}}$$

Component-wise product and division.

Important Operations on Vectors

Kronecker-product:

$$\boldsymbol{v} = \underbrace{\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}}_{\in \mathbb{R}^n}, \quad \boldsymbol{u} = \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}}_{\in \mathbb{R}^m}, \quad \boldsymbol{v} \otimes \boldsymbol{u} = \begin{bmatrix} v_1\boldsymbol{u} \\ v_2\boldsymbol{u} \\ \vdots \\ v_n\boldsymbol{u} \end{bmatrix} = \underbrace{\begin{bmatrix} v_1 \cdot u_1 \\ v_1 \cdot u_2 \\ \vdots \\ v_1 \cdot u_m \\ v_2 \cdot u_1 \\ v_2 \cdot u_2 \\ \vdots \\ v_2 \cdot u_m \\ \vdots \\ v_n \cdot u_1 \\ v_n \cdot u_2 \\ \vdots \\ v_n \cdot u_m \end{bmatrix}}_{\text{Result of this product is a vector in } \mathbb{R}^{nm}}$$

Vectors $\boldsymbol{v}$ and $\boldsymbol{u}$ can have different dimensionality

Not symmetric: $\boldsymbol{v} \otimes \boldsymbol{u} \neq \boldsymbol{u} \otimes \boldsymbol{v}$

Example:

$$\boldsymbol{v} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} 3 \\ -5 \\ 20 \end{bmatrix},$$

$$\boldsymbol{v} \otimes \boldsymbol{u} = \begin{bmatrix} 1 \cdot 3 \\ 1 \cdot (-5)) \\ 1 \cdot 20 \\ 2 \cdot 3 \\ 2 \cdot (-5)) \\ 2 \cdot 20 \end{bmatrix} = \begin{bmatrix} 3 \\ -5 \\ 20 \\ 6 \\ -10 \\ 40 \end{bmatrix}$$

Vector Norms

Norm of vector $\boldsymbol{v}$ is any univariate function $||\cdot||$, satisfying norm axioms:

- definiteness: $||\boldsymbol{v}|| = 0 \rightarrow \boldsymbol{v} = \mathbf{0}$
- absolute homogeneity : $||\alpha\boldsymbol{v}|| = |\alpha|||\boldsymbol{v}||$ for any vectors $\boldsymbol{v}$ and scalar α
- subadditivity or triangular inequality: $||\boldsymbol{u} + \boldsymbol{v}|| \leq ||\boldsymbol{u}|| + ||\boldsymbol{v}||$ for any vectors $\boldsymbol{v}$ and $\boldsymbol{u}$
- Non-negativity: $||\boldsymbol{v}|| \geq 0$ for any vector $\boldsymbol{v}$

Application in ML:

Train ML model: $\min_{\boldsymbol{w}} \mathcal{L}_{\text{loss}}(\boldsymbol{w}, \mathcal{D}) + ||\boldsymbol{w}||$

Regularisation term

Some Vector Norms and some Not Vector Norms

- Euclidian (or L_2 -norm):

$$||\boldsymbol{v}||_2 = \sqrt{v_1^2 + v_2^2 + \ldots + v_n^2}$$

$$\boldsymbol{v} = [1, 3, 5]^\mathsf{T} \quad , ||\boldsymbol{v}||_2 = \sqrt{1^2 + 3^2 + 5^2} = \sqrt{1 + 9 + 25} = \sqrt{35}.$$

- L_1 -norm:

$$||\boldsymbol{v}||_1 = |v_1| + \ldots + |v_n|$$

$$\boldsymbol{u} = [1, -3, 4]^\mathsf{T} \quad , ||\boldsymbol{u}||_1 = |1| + |-3| + |4| = 1 + 3 + 4 = 8$$

- Infinity norm (or L_∞ -norm):

$$||\boldsymbol{v}||_\infty = \max\{|v_1|, |v_2|, \ldots, |v_n|\}$$

$$\boldsymbol{w} = [-20, -3, 8]^\mathsf{T}, \quad ||\boldsymbol{w}||_\infty \max\{|-20|, |-3|, |8|\} = \\ = \max\{20, 3, 8\} = 20$$

- Sparsity norm (or L_0 -norm):

$$||\boldsymbol{v}||_0 = |\{i \in [1, \ldots, n] \text{ such that } v_i \neq 0\}|$$

$$\boldsymbol{h} = [0, -3, 0, 4]^\mathsf{T} \quad , ||\boldsymbol{h}||_0 = 2$$

One of them, actually is not a norm… Which one and why?

Vector Norms (Equivalence)

For the same vector $\boldsymbol{v}$ different norms gives different values….

However, for any two norm $||\cdot||_p$ and $||\cdot||_q$ there are constants $0 < a \leq b$ such that:

$$a||\boldsymbol{v}||_p \leq ||\boldsymbol{v}||_q \leq b||\boldsymbol{v}||_p \quad \text{for any vector } \boldsymbol{v}$$

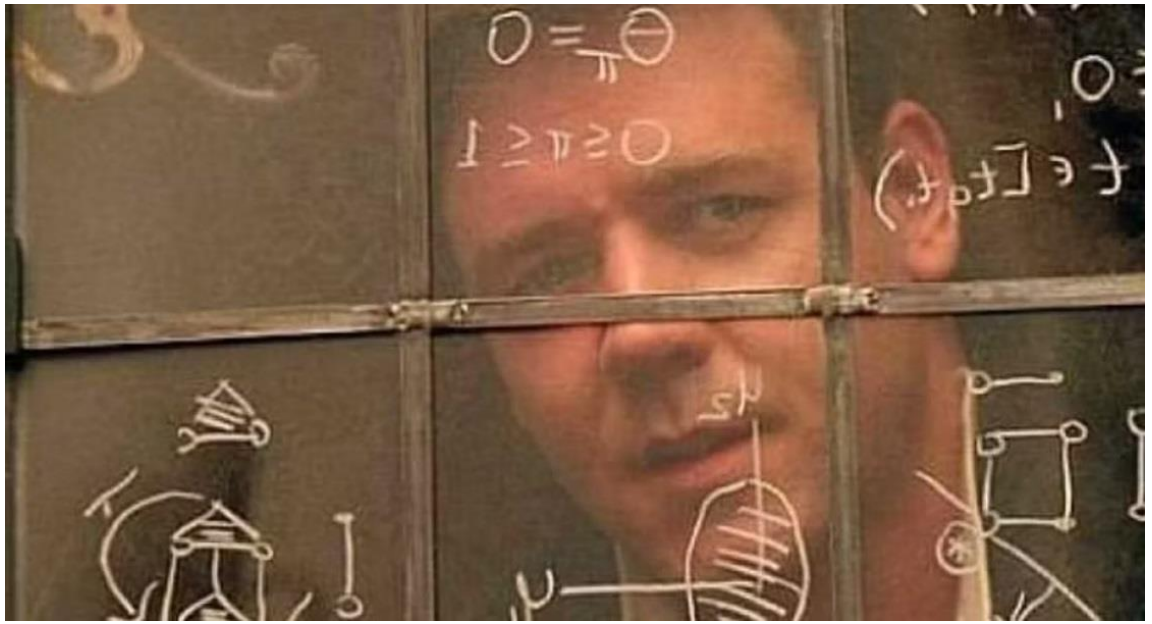
Equivalence of norms

Exercise:

Prove the following statements:

- If $||\cdot||$ is a valid norm, then $||u - v||$ is a valid distance function between points $\boldsymbol{u}$ and $\boldsymbol{v}$
- For any vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ any scalar product function:

$$|<\boldsymbol{u}, \boldsymbol{v}>| \leq ||\boldsymbol{u}||_2||\boldsymbol{v}||_2$$



Linear Independence

Collection of vectors $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_K\}$ is called ***linear independent*** if:

$$\alpha_1 \boldsymbol{v}_1 + \alpha_2 \boldsymbol{v}_2 + \ldots + \alpha_K \boldsymbol{v}_K = \mathbf{0} \quad \textbf{implies} \quad \alpha_1 = 0, \alpha_2 = 0, \ldots \alpha_K = 0$$

Example:

$$\boldsymbol{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \boldsymbol{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \boldsymbol{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \quad \ldots, \quad \boldsymbol{e}_n = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

Standard Basis in $\mathbb{R}^n$

are linearly independent

any vector in $\mathbb{R}^n$ can be written as their linear combination:

$$\begin{bmatrix} 2.7 \\ -13 \\ 5 \\ \frac{2}{7} \end{bmatrix} = 2.7 \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\boldsymbol{e}_1} - 13 \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}}_{\boldsymbol{e}_2} + 5 \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{\boldsymbol{e}_3} + \frac{2}{7} \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{\boldsymbol{e}_4}$$

$$\boldsymbol{v}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{v}_2 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad \boldsymbol{v}_3 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$\boldsymbol{v}_1 + \boldsymbol{v}_3 - 2\boldsymbol{v}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 \\ 1 \end{bmatrix} + 2 \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 - 1 + 0 \\ 1 + 1 - 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

are NOT linearly independent

Now, we are moving to the next topic….

matrices…

Reality

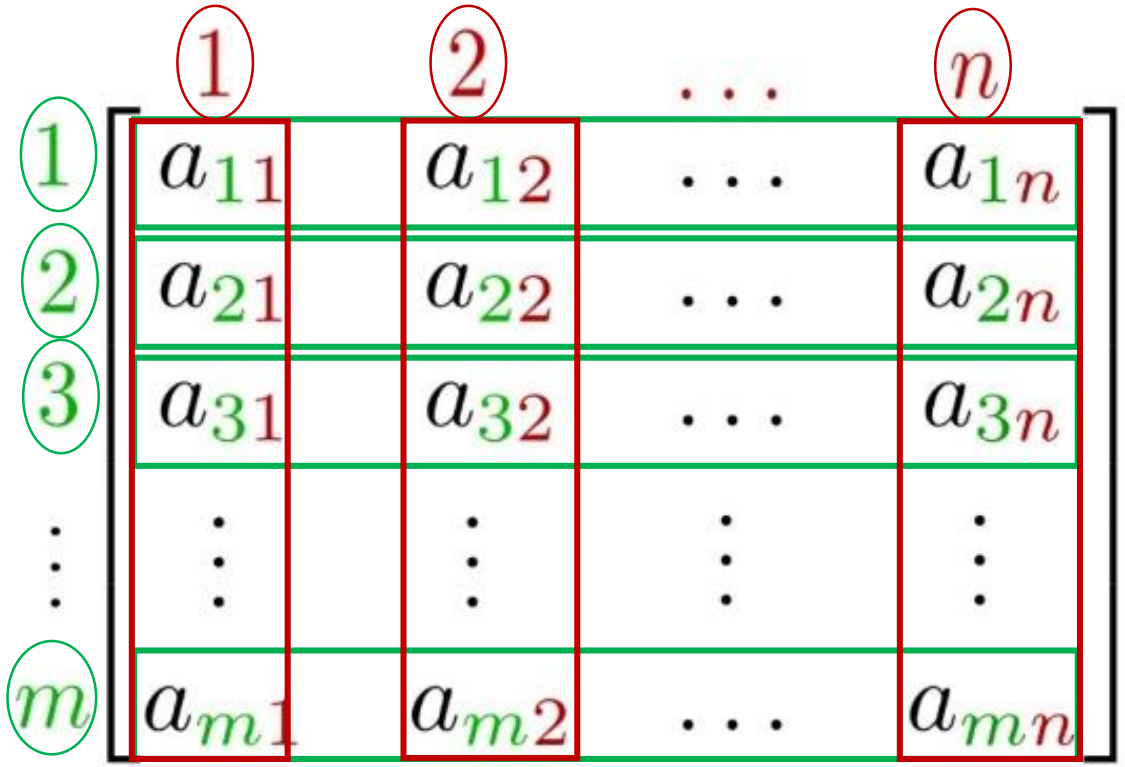
Given

$$A = \begin{bmatrix} 2 & 3 & 0 \\ 4 & 3 & 7 \end{bmatrix} \qquad B = \begin{bmatrix} 5 & 7 \\ 6 & 4 \end{bmatrix}$$

Find AB and BA.

Matrices

Matrix of size *m* by *n* of real numbers has the following form:



Examples:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 2 \\ 3 & 4 \\ 8 & -0.65 \\ \frac{1}{3} & 5 \end{bmatrix} \in \mathbb{R}^{4\times 2}, \quad \boldsymbol{B} = \begin{bmatrix} 0 & 2 & 3 & 7 \\ 8 & 4 & 0.4 & -5 \\ 2 & -0.65 & \frac{3}{7} & -\log 3 \end{bmatrix} \in \mathbb{R}^{3\times 4} \quad \boldsymbol{C} = \begin{bmatrix} 0 & 2 & 6 & \frac{2}{3} \\ 3 & 4 & -0.75 & 43 \\ 8 & -0.65 & \log 7 & 1 \\ \frac{1}{3} & 5 & 0 & -54 \end{bmatrix} \in \mathbb{R}^{4\times 4}$$

If *m* = *n* matric is called square matrix

Matrix Operations (Transpose)

Transpose operation on matrix swaps rows with columns:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n} \quad \Rightarrow \quad \boldsymbol{A}^{\mathsf{T}} = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \cdots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix} \in \mathbb{R}^{n \times m}$$

Matrix $\boldsymbol{A}$ is called ***symmetric*** if: $\boldsymbol{A}^{\mathsf{T}} = \boldsymbol{A}$

Example:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 5 & 0 \\ 5 & -2 & 9 \\ 0 & 9 & 91 \end{bmatrix} \rightarrow \boldsymbol{A}^{\mathsf{T}} = \begin{bmatrix} 1 & 5 & 0 \\ 5 & -2 & 9 \\ 0 & 9 & 91 \end{bmatrix} \Longrightarrow \boldsymbol{A} = \boldsymbol{A}^{\mathsf{T}} \in \mathbb{R}^{4 \times 3}$$

Symmetric:

$$\boldsymbol{B} = \begin{bmatrix} 1 & 5 & 0 \\ -7 & 2.3 & 9 \\ -12 & 9 & 91 \end{bmatrix} \rightarrow \boldsymbol{B}^{\mathsf{T}} = \begin{bmatrix} 1 & -7 & -12 \\ 5 & 2.3 & 9 \\ 0 & 9 & 91 \end{bmatrix} \Longrightarrow \boldsymbol{B} \neq \boldsymbol{B}^{\mathsf{T}} \times 3$$

NOT symmetric:

Square matrix remains square after transposition

Matrix Operations (Linear Operations)

Linear operation for matrices are element-wise:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad \boldsymbol{B} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \dots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{bmatrix}$$

for any $\alpha, \beta \in \mathbb{R}$

$$\alpha\boldsymbol{A} + \beta\boldsymbol{B} = \begin{bmatrix} \alpha a_{11} + \beta b_{11} & \alpha a_{12} + \beta b_{12} & \dots & \alpha a_{1n} + \beta b_{1n} \\ \alpha a_{21} + \beta b_{21} & \alpha a_{22} + \beta b_{22} & \dots & \alpha a_{2n} + \beta b_{2n} \\ \vdots & \vdots & \dots & \vdots \\ \alpha a_{m1} + \beta b_{m1} & \alpha a_{m2} + \beta b_{m2} & \dots & \alpha a_{mn} + \beta b_{mn} \end{bmatrix}$$

both $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m \times n}$ must be the same dimensionality

Examples:

$$\boldsymbol{A} = \begin{bmatrix} 2 & 3 & 5 \\ -1 & 2 & 3 \\ -4 & 10 & 7 \end{bmatrix} \quad \boldsymbol{B} = \begin{bmatrix} 10 & 100 & 1000 \\ 1 & -3 & 5 \\ 5 & -5 & 3 \end{bmatrix}$$

$$2\boldsymbol{A} - 0.5\boldsymbol{B} = \begin{bmatrix} 2 \cdot 2 - 0.5 \cdot 10 & 2 \cdot 3 - 0.5 \cdot 100 & 2 \cdot 5 - 0.5 \cdot 1000 \\ 2 \cdot (-1) - 0.5 \cdot 1 & 2 \cdot 2 - 0.5 \cdot (-3) & 2 \cdot 3 - 0.5 \cdot 5 \\ 2 \cdot (-4) - 0.5 \cdot 5 & 2 \cdot 10 - 0.5 \cdot (-5) & 2 \cdot 7 - 0.5 \cdot 3 \end{bmatrix} = \begin{bmatrix} -1 & -44 & -490 \\ -2.5 & 5.5 & 3.5 \\ -10.5 & 22.5 & 12.5 \end{bmatrix}$$

Matrix Operations (Multiplication)

For two matrices $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n \times p}$:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \dots & a_{in} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad \text{and} \quad \boldsymbol{B} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1j} & \dots & b_{1p} \\ b_{21} & b_{22} & \dots & b_{2j} & \dots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nj} & \dots & b_{np} \end{bmatrix}$$

Number of columns in $\boldsymbol{A}$ must be equal to number of rows in $\boldsymbol{B}$

$$\boldsymbol{AB} = \boldsymbol{C} = \underbrace{\begin{bmatrix} c_{11} & c_{12} & \dots & c_{1j} & \dots & c_{1p} \\ c_{21} & c_{22} & \dots & c_{2j} & \dots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{i1} & c_{i2} & \dots & c_{ij} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \dots & c_{mj} & \dots & c_{mp} \end{bmatrix}}_{\in \mathbb{R}^{m \times p}}$$

where $c_{ij} = \underbrace{\begin{bmatrix} a_{i1} & a_{i2} & \dots a_{in} \end{bmatrix} \begin{bmatrix} b_{1j} \\ b_{2j} \\ \vdots \\ b_{nj} \end{bmatrix}}_{\text{inner product}} = \sum_{k=1}^{n} a_{ik} b_{nk}$

Matrix Operations (Multiplication)

Examples:

$$A = \begin{bmatrix} 1 & 2 \\ -3 & 4 \\ 5 & 0.5 \end{bmatrix} \qquad B = \begin{bmatrix} -1 & 0 & 0 & -1 \\ 1 & 1 & -2 & 0 \end{bmatrix}$$

Notice, number of columns in A is 2 and number of rows in B is 2

$$C = \begin{bmatrix} 1\cdot(-1)+2\cdot 1 & 1\cdot 0+2\cdot 1 & 1\cdot 0+2\cdot(-2) & 1\cdot(-1)+2\cdot 0 \\ (-3)\cdot(-1)+4\cdot 1 & (-3)\cdot 0+4\cdot 1 & (-3)\cdot 0+4\cdot(-2) & (-3)\cdot(-1)+4\cdot 0 \\ 5\cdot(-1)+0.5\cdot 1 & 5\cdot 0+0.5\cdot 1 & 5\cdot 0+0.5\cdot(-2) & 5\cdot(-1)+0.5\cdot 0 \end{bmatrix} = \begin{bmatrix} 1 & 2 & -4 & -1 \\ 7 & 4 & -8 & 4 \\ -4.5 & 0.5 & -1 & -5 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 2 & 0 \\ -3 & 4 & 8 \\ 5 & 0.5 & -3 \\ 0 & -1 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} 10 \\ 100 \\ 100 \end{bmatrix} \qquad Ab = \begin{bmatrix} 1\cdot 10+2\cdot 100+0\cdot 1000 \\ (-3)\cdot 10+4\cdot 100+8\cdot 1000 \\ 5\cdot 10+0.5\cdot 100+(-3))\cdot 1000 \\ 0\cdot 10+(-1)\cdot 100+1\cdot 1000 \end{bmatrix} = \begin{bmatrix} 210 \\ 8370 \\ -2800 \\ 900 \end{bmatrix}$$

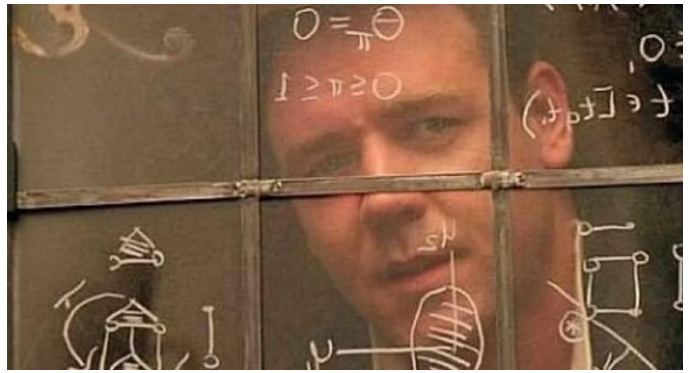
Matrix vector multiplication

Exercise:

Give example of two square matrices A, B such that $AB \neq BA$

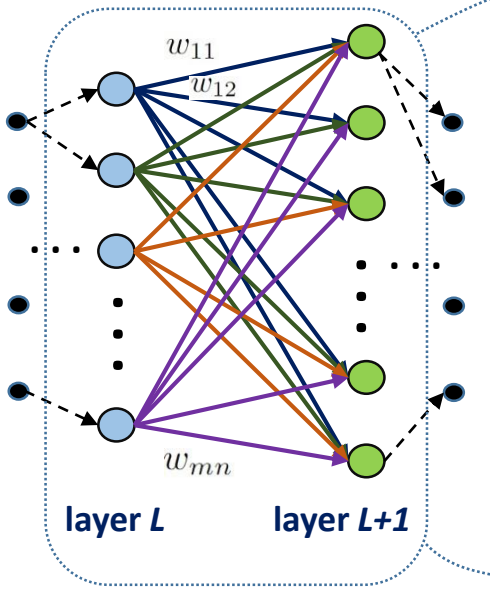
Show that for any square matrix: $AI = IA$ where $I = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \in \mathbb{R}^{n\times n}$

identity matrix



Matrix Operations (Multiplication)

ML application:



Layers in neural NN are connected via weights:

$$\boldsymbol{X}_{\text{input}} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_m \end{bmatrix}$$

$x_1 w_{11} + x_2 w_{21} + \ldots + x_m w_{m1}$

$x_1 w_{12} + x_2 w_{22} + \ldots + x_m w_{m2}$

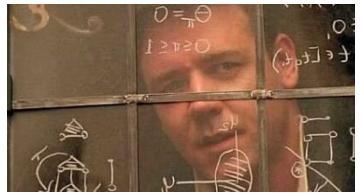
$x_1 w_{13} + x_2 w_{23} + \ldots + x_m w_{m3}$

$x_1 w_{1n} + x_2 w_{2n} + \ldots + x_m w_{mn}$

Results mathematical expressions:

$$x_1 w_{11} + x_2 w_{21} + x_3 w_{31} + \ldots + x_m w_{m1}$$
$$x_1 w_{12} + x_2 w_{22} + x_3 w_{32} + \ldots + x_m w_{m2}$$
$$x_1 w_{13} + x_2 w_{23} + x_3 w_{33} + \ldots + x_m w_{m3}$$
$$\vdots$$
$$x_1 w_{1n} + x_2 w_{2n} + x_3 w_{3n} \ldots + x_m w_{mn}$$

Exercise:



$$= \begin{bmatrix} w_{11} & w_{12} & w_{13} & \ldots & w_{1n} \\ w_{21} & w_{22} & w_{23} & \ldots & w_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{m1} & w_{m2} & w_{m3} & \ldots & w_{mn} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_m \end{bmatrix} = \boldsymbol{W}^{\mathsf{T}} \boldsymbol{X}_{\text{input}}$$

Matrix Multiplication (Compexity)

How computational expensive is matrix multiplication?

$$\boldsymbol{A} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}}_{\in \mathbb{R}^{n \times n}} \quad \boldsymbol{B} = \underbrace{\begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix}}_{\in \mathbb{R}^{n \times n}} \Rightarrow \boldsymbol{C} = \boldsymbol{AB} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \dots & c_{nn} \end{bmatrix}$$

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

Each entry requires *n* multiplications and *n* additions

There are total n^2 entries in matrix $\boldsymbol{C}$ ⇒ Total number of arithmetic operations: $2n^3 = \mathcal{O}(n^3)$

Best Current Theoretical Algorithm:

A Refined Laser Method and Faster Matrix Multiplication

Josh Alman* Virginia Vassilevska Williams†

October 13, 2020

$$\mathcal{O}(n^w), \quad w \leq 2.37286$$

previous best: $w \leq 2.37287$

But… it is galactic algorithm

Widely Used Practical Algorithm

Published: August 1969

Gaussian elimination is not optimal

Volker Strassen

Numerische Mathematik **13**, 354–356 (1969) | Cite this article

$$\mathcal{O}(n^{\log_2 7})$$

Cool Recent Advances

Discovering faster matrix multiplication algorithms with reinforcement learning

Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J. R. Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, David Silver, Demis Hassabis & Pushmeet Kohli

Nature **610**, 47–53 (2022) | Cite this article

$$\mathcal{O}(n^{\log_4 47})$$

Inner Products of Matrices

Given two matrices $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n \times m}$ can we compute inner product $< \boldsymbol{A}, \boldsymbol{B} >$ between them?

Method 1: Via the Trace

HELL YEAH

Trace of the square matrix $\boldsymbol{G} \in \mathbb{R}^{k \times k}$:

$$\boldsymbol{G} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1k} \\ g_{21} & g_{22} & \cdots & g_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kk} \end{bmatrix} \Rightarrow \operatorname{Tr}(\boldsymbol{G}) = \sum_{i=1}^{k} g_{ii}$$

Example:

$$\boldsymbol{G} = \begin{bmatrix} 1 & 3 & 4 \\ 41 & -3 & 80 \\ -0.9 & 65 & 21 \end{bmatrix} \Rightarrow \operatorname{Tr}(\boldsymbol{G}) = 1 + (-3) + 21 = 19$$

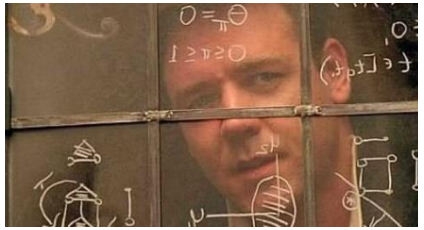
The inner product now can be defined as: $< \boldsymbol{A}, \boldsymbol{B} >= \operatorname{Tr}(\boldsymbol{AB}) = \sum_{i=1}^{m} c_{ii} = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ik} b_{ki}$

Example:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 3 \\ 0 & -2 \\ 9 & -5 \\ 5 & -0.5 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0 & 1 & 6 & 2 \\ -1 & 5 & -2.5 & 1 \end{bmatrix} \Rightarrow \boldsymbol{AB} = \begin{bmatrix} 3 & 16 & -1.5 & 5 \\ 2 & -10 & 5 & -2 \\ 95 & -16 & 66.5 & 13 \\ 0.5 & 2.5 & 31.25 & 9.5 \end{bmatrix}$$

$$\Rightarrow \operatorname{Tr}(\boldsymbol{AB}) = 3 + (-10) + 66.5 + 9.5 = 69$$

Exercise:



Show symmetric property: $\operatorname{Tr}(\boldsymbol{AB}) = \operatorname{Tr}(\boldsymbol{BA})$

Hint:

Use definition of trace and try reordering of terms

Inner Products of Matrices

Given two matrices $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n \times m}$ can we compute inner product $< \boldsymbol{A}, \boldsymbol{B} >$ between them?

Method 2: Via Matrix Vectorization:

Any matrix $\boldsymbol{G} \in \mathbb{R}^{n \times m}$ can be represented using vector notation:

$$\boldsymbol{G} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1m} \\ g_{21} & g_{22} & \cdots & g_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{nm} \end{bmatrix}$$

$$\text{vec}(\boldsymbol{G}) = \begin{bmatrix} g_{11} \\ g_{12} \\ \cdots \\ g_{1m} \\ g_{21} \\ g_{22} \\ \cdots \\ g_{2m} \\ \vdots \\ g_{n1} \\ g_{n2} \\ \cdots \\ g_{nm} \end{bmatrix}$$

concatenating rows :

OR

$$\text{vec}(\boldsymbol{G}) = \begin{bmatrix} g_{11} \\ g_{21} \\ \cdots \\ g_{n1} \\ g_{12} \\ g_{22} \\ \cdots \\ g_{n2} \\ \vdots \\ g_{1m} \\ g_{2m} \\ \cdots \\ g_{nm} \end{bmatrix}$$

concatenating columns :

Example:

$$\boldsymbol{G} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

$$\text{vec}(\boldsymbol{G}) = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} \qquad \text{vec}(\boldsymbol{G}) = \begin{bmatrix} 1 \\ 3 \\ 5 \\ 2 \\ 4 \\ 6 \end{bmatrix}$$

Now, we can use any inner product between vectors:

$$< \boldsymbol{A}, \boldsymbol{B} > = < \text{vec}(\boldsymbol{A}), \text{vec}(\boldsymbol{B}) >$$

Matrix Inverse

Square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is called inverse to another square matrix $\boldsymbol{B} \in \mathbb{R}^{n \times n}$ if:

$$\boldsymbol{AB} = \boldsymbol{BA} = \boldsymbol{I}$$

$$\boldsymbol{I} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}}_{\text{Identity matrix:}} \in \mathbb{R}^{n \times n}$$

- **Only square matrices might have inverse.**
- **Not all square matrices have inverse:**

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$ **- is not invertible:**

Example:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 1 \\ 4 & 4 & 5 \\ 6 & 7 & 7 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} -7 & -7 & 6 \\ 2 & 1 & -1 \\ 4 & 5 & -4 \end{bmatrix} \Rightarrow \boldsymbol{AB} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Matrix Inverse Computation:

Gaussian Elimination Algorithm

LU Decomposition Algorihms

$$\mathcal{O}(n^3) \Rightarrow$$

In general, big matrices **are expensive** to invert…

But, if matrix has **special structure**:

- Symmetric diagonal dominant
- Sparse lower/upper triangular

⋮

can be inverted in a much faster way

Solving Linear Equations

System of ***m*** linear equations with ***n*** variables have general form:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots a_{2n}x_n = b_2 \\ a_{31}x_1 + a_{32}x_2 + \ldots a_{3n}x_n = b_3 \\ \ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots a_{mn}x_n = b_m \end{cases}$$

Unknown variables :

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n$$

Known variables :

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n} \qquad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \in \mathbb{R}^m$$

In a ***matrix vector form*** can be written compactly:

$$\boldsymbol{Ax} = \boldsymbol{b}$$

Example:

$$\begin{cases} 2x_1 - 3x_2 + 5x_3 = 2 \\ 6x_1 + x_2 - 7x_3 = -3 \\ 5x_1 - 8x_2 + 0.5x_3 = 0 \end{cases}$$

$$\boldsymbol{A} = \begin{bmatrix} 2 & -3 & 5 \\ 6 & 1 & -7 \\ 5 & -8 & 0.5 \end{bmatrix} \qquad \boldsymbol{b} = \begin{bmatrix} 2 \\ -3 \\ 0 \end{bmatrix}$$

Solving Linear Equations

Given a system of linear equations: $$\boldsymbol{Ax} = \boldsymbol{b}$$

Can be reduced to matrix inversion problem

(by multiplying both sides on $\boldsymbol{A}^\mathsf{T}$): $$\underbrace{\boldsymbol{A}}_{\in \mathbb{R}^{m\times n}}\boldsymbol{x} = \boldsymbol{b} \iff \underbrace{\boldsymbol{A}^\mathsf{T}\boldsymbol{A}}_{\text{square matrix:}\in \mathbb{R}^{n\times n}}\boldsymbol{x} = \boldsymbol{A}^\mathsf{T}\boldsymbol{b} \Rightarrow \boldsymbol{x}^* = \underbrace{\left(\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\right)^{-1}}_{\text{Inverse of } \boldsymbol{A}^\mathsf{T}\boldsymbol{A}}\boldsymbol{A}^\mathsf{T}\boldsymbol{b}$$

Sometimes, instead of multiplying on $\boldsymbol{A}^\mathsf{T}$, we also multiply on matrix $\boldsymbol{G} \in \mathbb{R}^{n\times m}$ such that $\boldsymbol{GA}$ has desirable (for inversion) structure:

$$\boldsymbol{Ax} = \boldsymbol{b} \iff \boldsymbol{GAx} = \boldsymbol{Gb}$$
$$\boldsymbol{x}^* = (\boldsymbol{GA})^{-1}\boldsymbol{Gb}$$

$\boldsymbol{G} \in \mathbb{R}^{n\times m}$ is called preconditioner

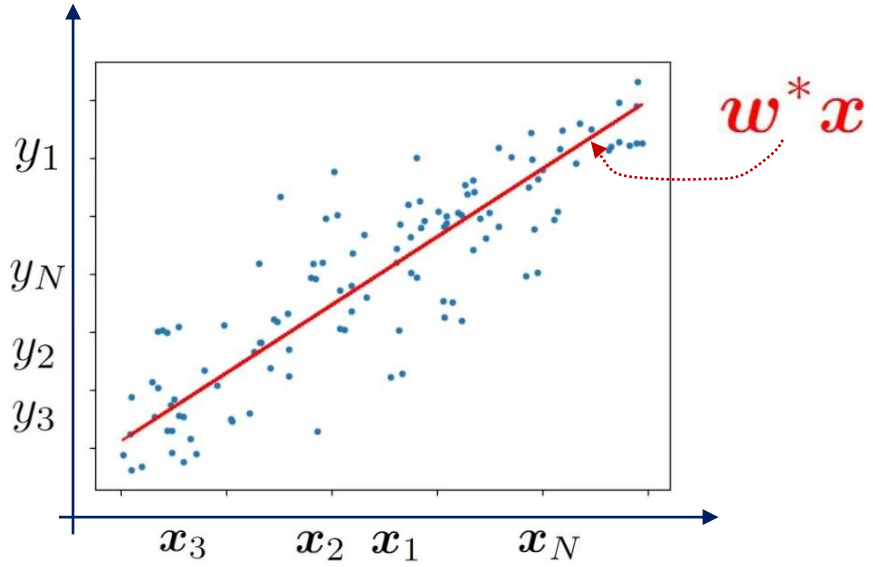
Since this reduction, linear equation solver has complexity bounded by inversion complexity: $\mathcal{O}(n^3)$

In practice, features like sparsity of matrix of the system $\boldsymbol{A}$ along with proper preprocessing can have huge effect on performance.

Solving Linear Equations (ML Application)

In linear regression task, we are given a data-set $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$

and our goal is to find ML model $\boldsymbol{w}^\mathsf{T}\boldsymbol{x}$ that fits this data.



To find the optimal model parameter we aim optimization problem:

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \mathcal{L}_{\text{loss}}(\boldsymbol{w}) = \sum_{i=1}^{N} (\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i - y_i)^2$$

How can we write it in matrix vector form?

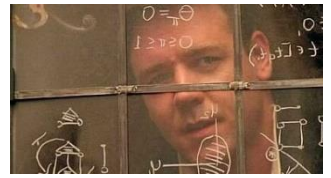
$$\mathcal{D} = \begin{matrix} \boldsymbol{x}_1, y_1 \\ \boldsymbol{x}_2, y_2 \\ \vdots \\ \boldsymbol{x}_N, y_N \end{matrix}$$

$$\boldsymbol{X}_{\text{data}} = \begin{bmatrix} \text{——} & \boldsymbol{x}_1^\mathsf{T} & \text{——} \\ \text{——} & \boldsymbol{x}_2^\mathsf{T} & \text{——} \\ & \vdots & \\ \text{——} & \boldsymbol{x}_N^\mathsf{T} & \text{——} \end{bmatrix}$$

$$\boldsymbol{y}_{\text{data}} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

in this notation

Why?



$$\mathcal{L}_{\text{loss}}(\boldsymbol{w}) = \sum_{i=1}^{N} (\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i - y_i)^2 = ||\boldsymbol{X}_{\text{data}}\boldsymbol{w} - \boldsymbol{y}_{\text{data}}||_2^2$$

Optimal parameter is the solution of the following system

$$\underbrace{\boldsymbol{X}_{\text{data}}^\mathsf{T}\boldsymbol{X}_{\text{data}}\boldsymbol{w}^* = \boldsymbol{X}_{\text{data}}^\mathsf{T}\boldsymbol{y}_{\text{data}}}_{\text{system of linear equations}}$$

Lets have 10 min break

Eigenvalues/Eigenvectors

For square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ a non-zero vector $\boldsymbol{x} \in \mathbb{R}^n$ such that:

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}$$

for some scalar $\lambda \in \mathbb{R}$

is called *eigenvector* of matrix $\boldsymbol{A}$ corresponding to *eigenvalue* λ. For same value λ you might have several eigenvectors.

Example:

$$\boldsymbol{A} = \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix} \qquad \boldsymbol{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \Rightarrow \boldsymbol{A}\boldsymbol{x} \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \cdot 1 + 1 \cdot 0 \\ 0 \cdot 1 + 2 \cdot 0 \end{bmatrix} = 2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 2\boldsymbol{x} \Rightarrow \boldsymbol{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

- is eigenvector with eigenvalue $\lambda = 2$

Geometric meaning:

Linear mapping $\boldsymbol{\Phi} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ transforms any vector $\boldsymbol{x} \in \mathbb{R}^n$ into a vector $\boldsymbol{\Phi}(\boldsymbol{x}) \in \mathbb{R}^m$ such that

$$\boldsymbol{\Phi}(\alpha\boldsymbol{x} + \beta\boldsymbol{z}) = \alpha\boldsymbol{\Phi}(\boldsymbol{x}) + \beta\boldsymbol{\Phi}(\boldsymbol{z}) \qquad \alpha, \beta \in \mathbb{R}$$

For any linear mapping $\boldsymbol{\Phi} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ there exist matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ such that $\boldsymbol{\Phi}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$

$\Rightarrow$ $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$ implies that vector $\boldsymbol{x} \in \mathbb{R}^n$ does not change after transformation defined by $\boldsymbol{A}$

Eigendecomposition and Diagonalisation

Square matrix $\boldsymbol{D} \in \mathbb{R}^{n \times n}$ is called *diagonal* if it has zero entries on all off-diagonal positions:

$$\boldsymbol{D} = \begin{bmatrix} d_{11} & 0 & \dots & 0 \\ 0 & d_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & d_{nn} \end{bmatrix}$$

these positions are called diagonal

these positions are called off-diagonal

Examples: $\boldsymbol{D} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & \frac{2}{5} \end{bmatrix}$ OR $\boldsymbol{D} = \begin{bmatrix} -2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 3 \end{bmatrix}$ OR $\boldsymbol{I} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$

A matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is called *diagonalizable* if there exist an invertible matrix $\boldsymbol{P} \in \mathbb{R}^{n \times n}$ such that:

$$\boldsymbol{A} = \boldsymbol{P}^{-1}\boldsymbol{D}\boldsymbol{P}$$

for some diagonal matrix $\boldsymbol{D} \in \mathbb{R}^{n \times n}$

Theorem (Eigendecomposition of symmetric matrices):

Let $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \dots \boldsymbol{u}_n\}$ be a collection of eigenvectors of symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ corresponding to eigenvalues $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$, i.e. $\boldsymbol{A}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i$ for $i \in \{1, \dots, n\}$. Then, matrix $\boldsymbol{A}$ can be factorized as:

$$\boldsymbol{A} = \begin{bmatrix} | & | & & | \\ \boldsymbol{u}_1 & \boldsymbol{u}_2 & \dots & \boldsymbol{u}_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \begin{bmatrix} — & \boldsymbol{u}_1^\top & — \\ — & \boldsymbol{u}_2^\top & — \\ & \vdots & \\ — & \boldsymbol{u}_n^\top & — \end{bmatrix}$$

Spectral Decomposition of Symmetric Matrix

Let us prove that any symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ can be decomposed as:

$$\boldsymbol{A} = \sum_{i=1}^{n} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^{\mathsf{T}}$$

where $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots \boldsymbol{u}_n\}$ are eigenvectors of $\boldsymbol{A}$ corresponding to eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$

How to show that two equal size matrices $\boldsymbol{A} = \boldsymbol{B}$? ⇨ We can show that their entries are equal: $a_{ij} = b_{ij}$ for $i, j \in \{1, \ldots, n\}$

Using ***Eigendecomposition Theorem***:

$$\boldsymbol{A} = \underbrace{\begin{bmatrix} | & | & \cdots & | \\ \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_n \\ | & | & \cdots & | \end{bmatrix}}_{\boldsymbol{U}} \underbrace{\begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}}_{\boldsymbol{\Lambda}} \underbrace{\begin{bmatrix} — & \boldsymbol{u}_1^{\mathsf{T}} & — \\ — & \boldsymbol{u}_2^{\mathsf{T}} & — \\ & \vdots & \\ — & \boldsymbol{u}_n^{\mathsf{T}} & — \end{bmatrix}}_{\boldsymbol{U}^{\mathsf{T}}}$$

We'll show: $\left[\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathsf{T}}\right]_{ij} = \left[\sum_{m=1}^{n} \lambda_m \boldsymbol{u}_m \boldsymbol{u}_m^{\mathsf{T}}\right]_{ij} = \sum_{m=1}^{n} \lambda_m \left[\boldsymbol{u}_m \boldsymbol{u}_m^{\mathsf{T}}\right]_{ij}$

for arbitrary $i, j \in \{1, \ldots, n\}$

Step 1: Computing $\left[\sum_{m=1}^{n} \lambda_m \left[\boldsymbol{u}_m \boldsymbol{u}_m^{\mathsf{T}}\right]_{ij}\right.$

Notice: $\lambda_m \boldsymbol{u}_m \boldsymbol{u}_m^{\mathsf{T}} = \lambda \begin{bmatrix} [\boldsymbol{u}_m]_1 & [\boldsymbol{u}_m]_2 & \ldots & [\boldsymbol{u}_m]_n \end{bmatrix} \begin{bmatrix} \lambda_1 [\boldsymbol{u}_1]_j \\ \lambda_2 [\boldsymbol{u}_2]_j \\ \vdots \\ \lambda_n [\boldsymbol{u}_n]_j \end{bmatrix} \boldsymbol{u}_n^{\mathsf{T}} = \lambda \begin{bmatrix} [\boldsymbol{u}_m]_1^2 & [\boldsymbol{u}_m]_1 [\boldsymbol{u}_m]_2 & \ldots & [\boldsymbol{u}_m]_1 [\boldsymbol{u}_m]_n \\ [\boldsymbol{u}_m]_2 [\boldsymbol{u}_m]_1 & [\boldsymbol{u}_m]_2^2 & \ldots & [\boldsymbol{u}_m]_2 [\boldsymbol{u}_m]_n \\ \vdots & \vdots & \ddots & \vdots \\ [\boldsymbol{u}_m]_n [\boldsymbol{u}_m]_1 & [\boldsymbol{u}_m]_n [\boldsymbol{u}_m]_2 & \ldots & [\boldsymbol{u}_m]_n^2 \end{bmatrix}$

Compare:

$$\left[\boldsymbol{U}\left[\lambda_m \boldsymbol{u}_m \boldsymbol{u}_m^{\mathsf{T}}\right]_{ij} = \lambda_m [\boldsymbol{u}_m]_i [\boldsymbol{u}_m]_j \right. \quad \begin{bmatrix} \lambda_1 [\boldsymbol{u}_1]_j \\ \lambda_2 [\boldsymbol{u}_2]_j \\ \vdots \\ \lambda_n [\boldsymbol{u}_n]_j \end{bmatrix}$$

lets remember: $\left[\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathsf{T}}\right]_{ij} = \sum_{m=1}^{n} \lambda_m [\boldsymbol{u}_m]_i [\boldsymbol{u}_m]_j$

Hence:

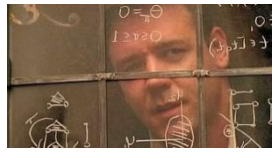
Hence: $\sum_{m=1}^{n} \lambda_m \left[\boldsymbol{u}_m \boldsymbol{u}_m^{\mathsf{T}}\right]_{ij} = \sum_{m=1}^{n} \lambda_m [\boldsymbol{u}_m]_i [\boldsymbol{u}_m]_j$

Positive Semidefinite (PSD) Matrices

A symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is called *positive semidefinite* (*positive definite*) if for any non-zero vector $\boldsymbol{x} \in \mathbb{R}^n$:

$$\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} \geq 0 \quad \left(\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} > 0\right) \qquad \textbf{Notation:} \quad \boldsymbol{A} \succeq 0 \ \left(\boldsymbol{A} \succ 0\right)$$

Exercise: Matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is positive semidefinite (positive definite) if all its eigenvalues are non-negative (positive).



ML application:

In Gaussian Processes, the symmetric covariance function $k(\boldsymbol{x}_1, \boldsymbol{x}_2)$ defines similarity between function values $f(\boldsymbol{x}_1)$ and $f(\boldsymbol{x}_2)$

To be proper covariance function, for any collection of points $\{\boldsymbol{x}_i\}_{i=1}^N$ their *covariance matrix* must be PSD:

$$\boldsymbol{K} = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & k(\boldsymbol{x}_1, \boldsymbol{x}_2) & \dots & k(\boldsymbol{x}_1, \boldsymbol{x}_n) \\ k(\boldsymbol{x}_2, \boldsymbol{x}_1) & k(\boldsymbol{x}_2, \boldsymbol{x}_2) & \dots & k(\boldsymbol{x}_2, \boldsymbol{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_n, \boldsymbol{x}_1) & k(\boldsymbol{x}_n, \boldsymbol{x}_2) & \dots & k(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{bmatrix} \succeq \boldsymbol{0}$$

Examples of such functions include:

- Square Exponential: $k_{\text{SE}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sigma^2 e^{-\frac{||\boldsymbol{x}_1 - \boldsymbol{x}_2||_2^2}{\ell}}$ $\quad \sigma, \ell -$
- Rational Quadratic: $k_{\text{RQ}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \left(1 + \frac{||\boldsymbol{x}_1 - \boldsymbol{x}_2||_2^2}{2\alpha k^2}\right)^{-\alpha}$ $\quad \alpha, k -$

parameters

Cholesky Factorisation

Theorem (Cholesky Factorisation):

Any symmetric, positive definite matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ can be factorized into a product:

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^{\mathsf{T}}$$

where $\boldsymbol{L}$ is *lower-triangular* matrix (called Cholesky factor):

$$\boldsymbol{L} = \begin{bmatrix} \ell_{11} & 0 & \dots & 0 \\ \ell_{21} & \ell_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \dots & \ell_{nn} \end{bmatrix}$$

André-Louis Cholesky (1875 - 1918)

Example:

$$\begin{bmatrix} 4 & 12 & -16 \\ 12 & 37 & -43 \\ -16 & -43 & 98 \end{bmatrix} = \underbrace{\begin{bmatrix} 2 & 0 & 0 \\ 6 & 1 & 0 \\ -8 & 5 & 3 \end{bmatrix}}_{\boldsymbol{L}} \underbrace{\begin{bmatrix} 2 & 6 & -8 \\ 0 & 1 & 5 \\ 0 & 0 & 3 \end{bmatrix}}_{\boldsymbol{L}^{\mathsf{T}}}$$

Computational complexity: $\mathcal{O}(n^3)$

Matrix Norms

Similar to vectors, we can introduce the norm function for matrices:

Norm of a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is any univariate function $||\cdot||$, satisfying norm axioms:

- definiteness: $||\boldsymbol{A}|| = 0 \longrightarrow \boldsymbol{A} = \boldsymbol{0}$
- absolute homogeneity : $||\alpha \boldsymbol{A}|| = |\alpha| ||\boldsymbol{A}||$ for any matrix $\boldsymbol{A}$ and scalar α
- subadditivity or triangular inequality: $||\boldsymbol{A} + \boldsymbol{B}|| \leq ||\boldsymbol{A}|| + ||\boldsymbol{B}||$ for any matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m \times n}$
- Non-negativity: $||\boldsymbol{A}|| > 0$ for any matrix $\boldsymbol{A}$
- Sub-multiplicativety: $||\boldsymbol{AB}|| \leq ||\boldsymbol{A}|| ||\boldsymbol{B}||$ for any matrices $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ $\boldsymbol{B} \in \mathbb{R}^{n \times p}$

ML application:

Non-linear support vector machines is trained using Log-Euclidian kernel: $k_{\mathrm{LE}}(\boldsymbol{A}, \boldsymbol{B}) = \exp\left(-\frac{1}{2\sigma^2} ||\log \boldsymbol{A} - \log \boldsymbol{B}||_{\mathrm{F}}^2\right)$

Frobenius norm

to compare covariance operators $\boldsymbol{A}, \boldsymbol{B}$

Matrix Norms (Examples)

Given matrix $\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_1 & a_{m2} & \dots & a_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n}$

Examples:

- Frobenius norm: $||\boldsymbol{A}||_{\mathrm{F}} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2}$

$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & -1 \end{bmatrix} \longrightarrow ||\boldsymbol{A}||_{\mathrm{F}} = \sqrt{1^2 + 2^2 + 3^2 + 0^2 + 1^2 + (-1)^2} = \sqrt{16} = 4$

- Maximum norm: $||\boldsymbol{A}||_1 = \max_{1 \le j \le n} \sum_{i=1}^{m} |a_{ij}|$

$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & -1 \end{bmatrix} \longrightarrow ||\boldsymbol{A}||_1 = \max\{1+0, 2+1, 3+1\} = \max\{1, 3, 4\} = 4$

- Infinity norm: $||\boldsymbol{A}||_\infty = \max_{1 \le i \le m} \sum_{j=1}^{n} |a_{ij}|$

$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & -1 \end{bmatrix} \longrightarrow ||\boldsymbol{A}||_\infty = \max\{1+2+3, 0+1+1\} = \max\{6, 2\} = 6$

- Spectral norm: $||\boldsymbol{A}||_2 = \sqrt{\lambda_{\max}(\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}})}$

Maximum eigenvalue of $\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}}$

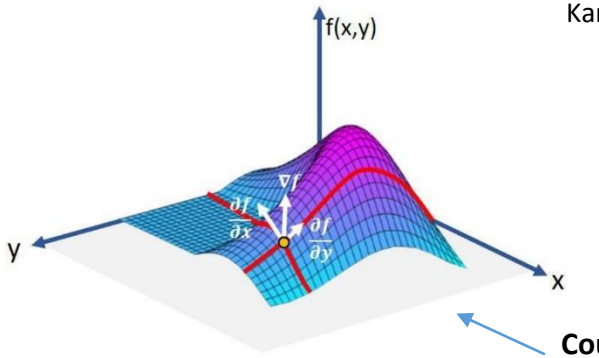
$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & -1 \end{bmatrix} \longrightarrow \boldsymbol{A}\boldsymbol{A}^{\mathsf{T}} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 2 & 1 \\ 3 & -1 \end{bmatrix} = \begin{bmatrix} 14 & -1 \\ -1 & 2 \end{bmatrix}$

$||\boldsymbol{A}||_2 = \sqrt{8 + \sqrt{37}}$

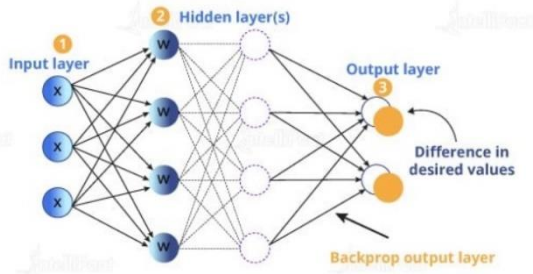
Vector Calculus

- Derivatives, Differentiation Rules.
- Partial Derivatives, Gradient, Hessian, Taylor Series.
- Jacobian of a smooth mapping.
- Gradients with respect to matrices.
- Backpropagation.

Karl Theodor Wilhelm Weierstrass
(1815 - 1897)



Courtesy of Brad Moffat



Courtesy of IntelliPaat

Derivatives, Differentiation Rules.

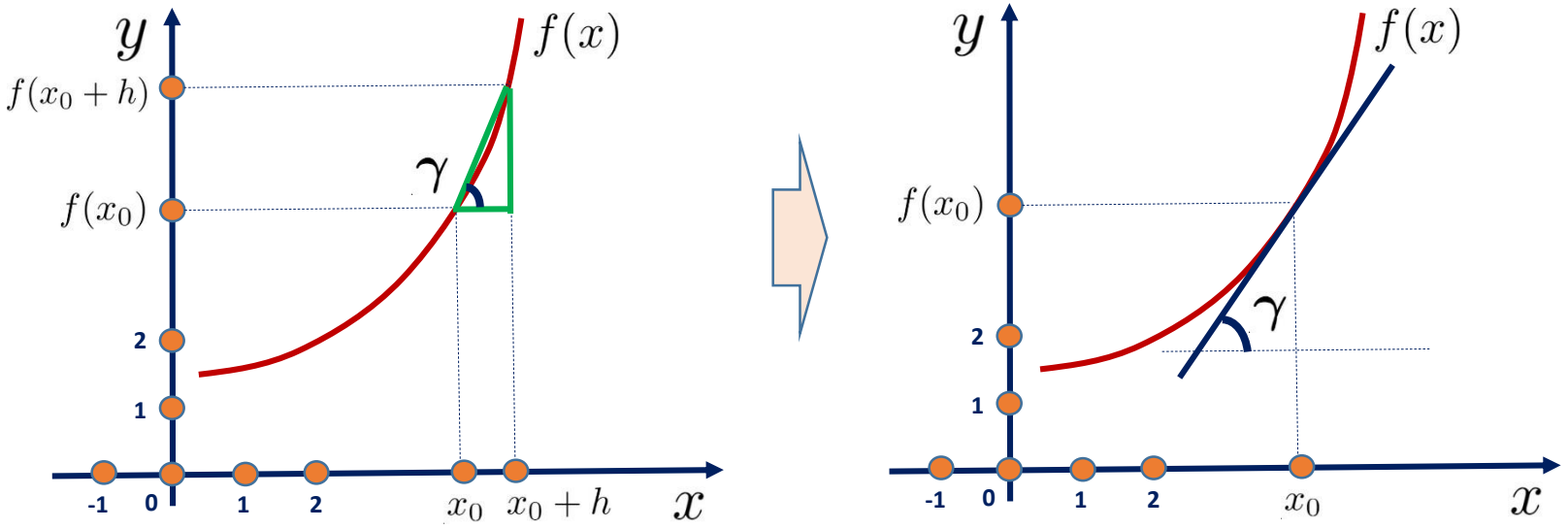
Let $f : \mathbb{R} \to \mathbb{R}$ be a univariate function. The *derivative of function* $f(x)$ at point x_0 is the following finite limit(if it exists):

$$\lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h} = f'(x_0)$$

If function has continuous derivative – it is called smooth.

Geometric interpretation:

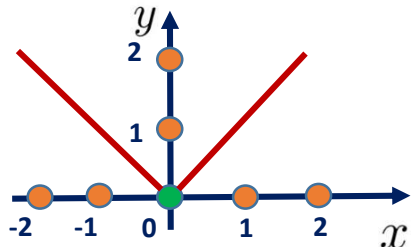
Ratio $\dfrac{f(x_0 + h) - f(x_0)}{h} = \tan(\gamma)$



As $h \to 0$ this ratio is equal to the slope of the tangent line at point x_0

$$\lim_{h \to 0} \tan(\gamma) = f'(x_0)$$

Not all functions are differentiable: $f(x) = |x|$ at $x_0 = 0$



Derivatives, Differentiation Rules

To compute derivative of a function we start with *known derivate expressions* and the apply *differentiation rules*:

Known Derivative Expressions:

$x' = 1$	$(x^2)' = 2x$	$(\csc x)' = -\cot x \csc x$	$(\arcsin x)' = \frac{1}{\sqrt{1-x^2}}$
$(x^n)' = nx^{n-1}$	$\left(\frac{1}{x}\right)' = -\frac{1}{x^2}$	$(\arccos x)' = -\frac{1}{\sqrt{1-x^2}}$	$(\arctan x)' = \frac{1}{1+x^2}$
$\left(\frac{1}{x^n}\right)' = -\frac{n}{x^{n+1}}$	$(\sqrt{x})' = \frac{1}{2\sqrt{x}}$	$(\operatorname{arccot} x)' = -\frac{1}{1+x^2}$	$(\operatorname{arcsec} x)' = \frac{1}{\lvert x\rvert\sqrt{x^2-1}}$
$(\sqrt[m]{x})' = \frac{1}{m\sqrt[m]{x^{m-1}}}$	$(a^x)' = a^x \ln a$	$(\operatorname{arccsc} x)' = -\frac{1}{\lvert x\rvert\sqrt{x^2-1}}$	$(\sinh x)' = \cosh x$
$(e^x)' = e^x$	$(\log_a x)' = \frac{1}{x \ln a}$	$(\cosh x)' = \sinh x$	$(\tanh x)' = \operatorname{sech}^2 x$
$(\ln x)' = \frac{1}{x}$	$(\sin x)' = \cos x$	$(\coth x)' = -\operatorname{csch}^2 x$	$(\operatorname{sech} x)' = -\operatorname{sech} x \tanh x$
$(\cos x)' = -\sin x$	$(\tan x)' = \frac{1}{\cos^2 x} = \sec^2 x$	$(\operatorname{csch} x)' = -\operatorname{csch} x \coth x$	$(\operatorname{arcsinh} x)' = \frac{1}{\sqrt{x^2+1}}$
$(\cot x)' = -\frac{1}{\sin^2 x} = -\csc^2 x$	$(\sec x)' = \tan x \sec x$		

Example:

$$(x^2)' = \lim_{h\to 0} \frac{(x+h)^2 - x^2}{h} = \lim_{h\to 0} \frac{x^2 + 2hx + h^2 - x^2}{h} = \lim_{h\to 0} \frac{2hx + h^2}{h} = \lim_{h\to 0} (2x + h) = 2x$$

Derivatives, Differentiation Rules

To compute derivative of a function we start with *known derivate expressions* and the apply *differentiation rules*:

Differentiation rules:

If functions $f(x), g(x)$ are differentiable at point x , then:

- Product rule: $(f(x)g(x))' = f'(x)g(x) + f(x)g'(x)$

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

- Sum rule: $(\alpha f(x) + \beta g(x))' = \alpha f'(x) + \beta g'(x)$

  for any scalars: $\alpha, \beta \in \mathbb{R}$

- Chain rule: $(f[g(x)])' = f'[g(x)]g'(x)$

Examples:

$$(x^2 \cos x)' = (x^2)' \cos x + x^2 (\cos x)' = 2x \cos x - x^2 \sin x$$

Using: $(\cos x)' = -\sin x$ **and** $(x^2)' = 2x$

$$\left(\frac{x^2}{\cos x}\right)' = \frac{(x^2)' \cos x - x^2 (\cos x)'}{\cos^2 x} = \frac{2x \cos x + x^2 \sin x}{\cos^2 x}$$

$$(4x^2 - 2\cos x)' = 4(x^2)' - 2(\cos x)' = 8x + 2\sin x$$

$$(\cos x^2)' = -\sin x^2 (x^2)' = -\sin x^2 (2x) = -2x \sin x^2$$

Using: $f(z) = \cos z$ **and** $g(x) = x^2$

Partial Derivatives

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a multivariate function. The *partial derivative of function* $f(\boldsymbol{x})$ at point $\boldsymbol{x}_0 = [[\boldsymbol{x}_0]_1, \ldots, [\boldsymbol{x}_0]_n]^\mathsf{T}$ are the following finite limits (if they exist):

$$\frac{\partial f(\boldsymbol{x}_0)}{\partial x_1} = \lim_{h_1 \to 0} \frac{f([\boldsymbol{x}_0]_1 + h_1, [\boldsymbol{x}_0]_2, \ldots, [\boldsymbol{x}_0]_n) - f([\boldsymbol{x}_0]_1, [\boldsymbol{x}_0]_2, \ldots, [\boldsymbol{x}_0]_n)}{h_1}$$

$$\frac{\partial f(\boldsymbol{x}_0)}{\partial x_2} = \lim_{h_2 \to 0} \frac{f([\boldsymbol{x}_0]_1, [\boldsymbol{x}_0]_2 + h_2, \ldots, [\boldsymbol{x}_0]_n) - f([\boldsymbol{x}_0]_1, [\boldsymbol{x}_0]_2, \ldots, [\boldsymbol{x}_0]_n)}{h_2}$$

$$\vdots$$

$$\frac{\partial f(\boldsymbol{x}_0)}{\partial x_n} = \lim_{h_n \to 0} \frac{f([\boldsymbol{x}_0]_1, [\boldsymbol{x}_0]_2, \ldots, [\boldsymbol{x}_0]_n + h_n) - f([\boldsymbol{x}_0]_1, [\boldsymbol{x}_0]_2, \ldots, [\boldsymbol{x}_0]_n)}{h_n}$$

Example:

For $f(x_1, x_2 x_3) = x_1^2 + x_1 x_2 + \frac{x_3^3}{x_2}$ compute

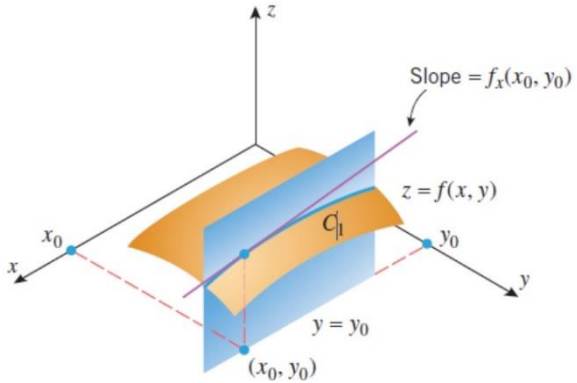
partial derivatives at point $\boldsymbol{x}_0 = [1, 3, 5]^\mathsf{T}$

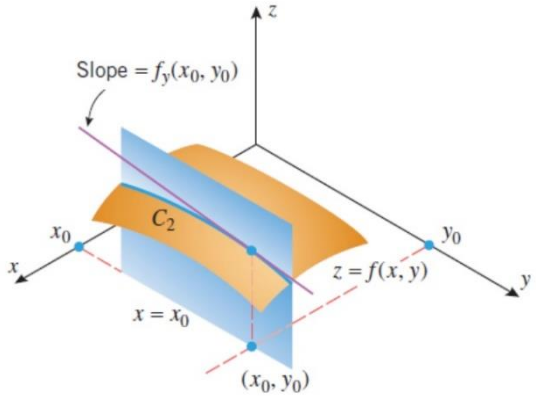
$$\frac{\partial f(\boldsymbol{x})}{\partial x_1} = 2x_1 + x_2 \Longrightarrow \frac{\partial f(\boldsymbol{x}_0)}{\partial x_1} = 2[\boldsymbol{x}_0]_1 + [\boldsymbol{x}_0]_2 = 2 \cdot 1 + 3 = 5$$

$$\frac{\partial f(\boldsymbol{x})}{\partial x_2} = x_1 - \frac{x_3^3}{x_2^2} \Longrightarrow \frac{\partial f(\boldsymbol{x}_0)}{\partial x_2} = [\boldsymbol{x}_0]_1 - \frac{[\boldsymbol{x}_0]_3^3}{[\boldsymbol{x}_0]_2^2} = 1 - \frac{5^3}{3^2} = -\frac{116}{9}$$

$$\frac{\partial f(\boldsymbol{x})}{\partial x_3} = \frac{3x_3^2}{x_2} \Longrightarrow \frac{\partial f(\boldsymbol{x}_0)}{\partial x_3} = \frac{3[\boldsymbol{x}_0]_3^2}{[\boldsymbol{x}_0]_2} = 3\frac{5^2}{3} = 25$$

Geometric interpretation:





Partial derivatives define slopes of the tangent line to the curve defined by the intersection of surface of the function and the hyperplane

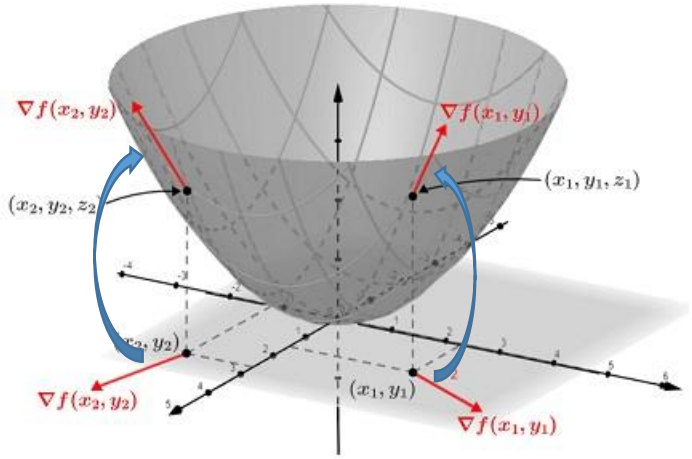
Gradient

Gradient of function $f : \mathbb{R}^n \to \mathbb{R}$ at point $\boldsymbol{x}_0 = [[\boldsymbol{x}_0]_1, \ldots, [\boldsymbol{x}_0]_n]^\mathsf{T}$ is a vector of all partial derivatives:

$$\frac{\mathrm{d}f(\boldsymbol{x}_0)}{\mathrm{d}\boldsymbol{x}} = \left[\frac{\partial f(\boldsymbol{x}_0)}{\partial x_1} \quad \frac{\partial f(\boldsymbol{x}_0)}{\partial x_2} \quad \cdots \quad \frac{\partial f(\boldsymbol{x}_0)}{\partial x_n}\right]^\mathsf{T} \in \mathbb{R}^n$$

Example: For $f(x_1, x_2x_3) = x_1^2 + x_1x_2 + \frac{x_3^3}{x_2}$ compute the gradient at point $\boldsymbol{x}_0 = [1, 3, 5]^\mathsf{T}$ $\Rightarrow$ $\frac{\mathrm{d}f(\boldsymbol{x}_0)}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} \frac{\partial f(\boldsymbol{x}_0)}{\partial x_1} \\ \frac{\partial f(\boldsymbol{x}_0)}{\partial x_2} \\ \frac{\partial f(\boldsymbol{x}_0)}{\partial x_3} \end{bmatrix} = \begin{bmatrix} 5 \\ -\frac{116}{9} \\ 25 \end{bmatrix}$

Geometric interpretation:



Consider paraboloid $z = f(x, y) = x^2 + y^2$, then gradient vectors $\nabla f(x_1, y_1)$ and $\nabla f(x_2, y_2)$ drawn in the $xy-$plane have their initial point placed at (x_1, y_1) and (x_2, y_2) respectively.

Note that the gradient vectors are calculated in component form but are translated to their respective input points on the $xy-$plane to better show the direction associated with those points.

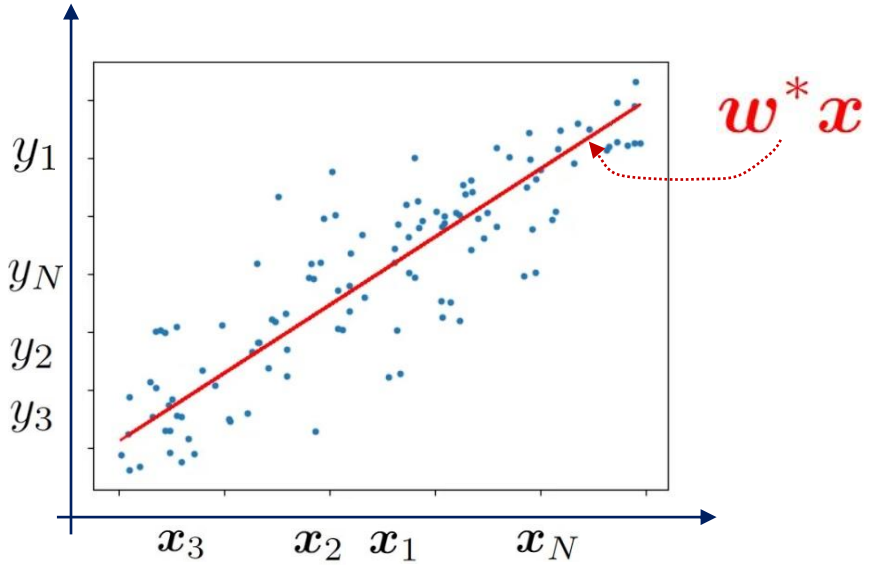
If we fold the $xy-$ plane so that $(x, y) \to (x, y, x^2 + y^2)$, then each corresponding gradient vector can also be mapped onto the paraboloid.

The gradient vectors mapped to (x_1, y_1, z_1) and (x_2, y_2, z_2) show the direction of fastest increase.

Gradient (ML Application)

In linear regression task, we are given a data-set $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$.

Our goal is to find ML model $\boldsymbol{w}^\mathsf{T}\boldsymbol{x}$ that fits this data.



To find the optimal model parameter we aim optimization problem:

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \mathcal{L}_{\text{loss}}(\boldsymbol{w}) = \sum_{i=1}^{N} (\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i - y_i)^2$$

At optimal point gradient vanishes: $\nabla_{\boldsymbol{w}} \mathcal{L}_{\text{loss}}(\boldsymbol{w}^*) = \boldsymbol{0}$

Let us compute this gradient:

using sum rule

re-group ter

using chain rule

Denote:

$$\nabla_{\boldsymbol{w}} \mathcal{L}_{\text{loss}}(\boldsymbol{w}) = \begin{bmatrix} \frac{\mathcal{L}_{\text{loss}}(\boldsymbol{w})}{\partial w_1} \\ \frac{\mathcal{L}_{\text{loss}}(\boldsymbol{w})}{\partial w_2} \\ \vdots \\ \frac{\mathcal{L}_{\text{loss}}(\boldsymbol{w})}{\partial w_n} \end{bmatrix}$$

fix arbitrary

$$= 2\boldsymbol{X}_{\text{data}}^\mathsf{T}\boldsymbol{X}_{\text{data}}\boldsymbol{w} - 2\boldsymbol{X}_{\text{data}}^\mathsf{T}\boldsymbol{y}_{\text{data}}$$

$$= 2\boldsymbol{X}_{\text{data}}^\mathsf{T}\boldsymbol{X}_{\text{data}}\boldsymbol{w} - 2\boldsymbol{X}_{\text{data}}^\mathsf{T}\boldsymbol{y}_{\text{data}}$$

$$\boldsymbol{X}_{\text{data}} = \begin{bmatrix} \text{——} & \boldsymbol{x}_1^\mathsf{T} & \text{——} \\ \text{——} & \boldsymbol{x}_2^\mathsf{T} & \text{——} \\ \vdots & \vdots & \vdots \\ \text{——} & \boldsymbol{x}_N^\mathsf{T} & \text{——} \end{bmatrix} \qquad \boldsymbol{y}_{\text{data}} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

$$\sum_{i=1}^{N} \boldsymbol{x}_i \boldsymbol{x}_i^\mathsf{T} = \boldsymbol{X}_{\text{data}}^\mathsf{T}\boldsymbol{X}_{\text{data}} \qquad \sum_{i=1}^{N} y_i \boldsymbol{x}_i = \boldsymbol{X}_{\text{data}}^\mathsf{T}\boldsymbol{y}_{\text{data}}$$

$$\nabla_{\boldsymbol{w}} \mathcal{L}_{\text{loss}}(\boldsymbol{w}^*) = \boldsymbol{0} \quad \Rightarrow \quad \boldsymbol{X}_{\text{data}}^\mathsf{T}\boldsymbol{X}_{\text{data}}\boldsymbol{w}^* = \boldsymbol{X}_{\text{data}}^\mathsf{T}\boldsymbol{y}_{\text{data}}$$

Why?

Hessian

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a multivariate function. Each partial derivative $\frac{\partial f(\boldsymbol{x})}{\partial x_j}$ is itself multivariate function: $\frac{\partial f(\boldsymbol{x})}{\partial x_j} : \mathbb{R}^n \to \mathbb{R}$

Hence, if it is differentiable, we can study partial derivatives of $\frac{\partial f(\boldsymbol{x})}{\partial x_j}$ itself:

$$\nabla_{\boldsymbol{x}} \frac{\partial f(\boldsymbol{x})}{\partial x_j} = \begin{bmatrix} \frac{\partial^2 f(\boldsymbol{x})}{\partial x_j \partial x_1} \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_j \partial x_2} \\ \vdots \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_j^2} \\ \vdots \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_j \partial x_n} \end{bmatrix}$$

combining into one matrix:

$$\nabla^2_{\boldsymbol{x}} f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1^2} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_3 \partial x_1} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_1} \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_2} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2^2} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_3 \partial x_2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_2} \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_3} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2 \partial x_3} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_3^2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_n} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2 \partial x_n} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_3 \partial x_n} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n^2} \end{bmatrix} \in \mathbb{R}^{n \times n}$$

Hessian matrix of partial derivatives

$\nabla_{\boldsymbol{x}} \frac{\partial f(\boldsymbol{x})}{\partial x_1}$ $\nabla_{\boldsymbol{x}} \frac{\partial f(\boldsymbol{x})}{\partial x_2}$ $\nabla_{\boldsymbol{x}} \frac{\partial f(\boldsymbol{x})}{\partial x_3}$ $\nabla_{\boldsymbol{x}} \frac{\partial f(\boldsymbol{x})}{\partial x_n}$

Example: For $f(x_1, x_2 x_3) = x_1^2 + x_1 x_2 + \frac{x_3^3}{x_2}$ compute Hessian matrix at point $\boldsymbol{x}_0 = [1, 3, 5]^\mathsf{T}$

First column:

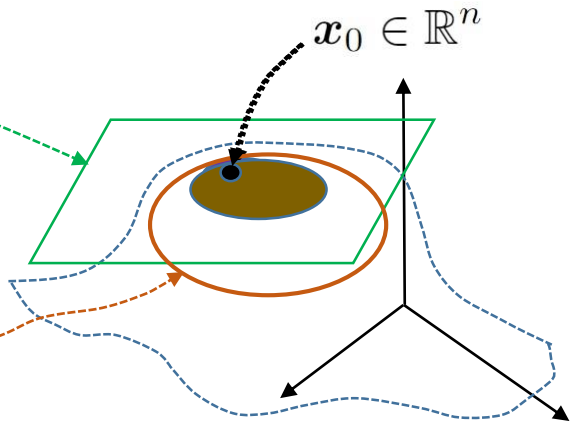
$$\nabla^2_{\boldsymbol{x}} f(\boldsymbol{x}) = \begin{bmatrix} 2 & 1 & 0 \\ 1 & \frac{2x_3^3}{x_2^3} & -\frac{3x_3^2}{x_2^2} \\ 0 & -\frac{3x_3^2}{x_2^2} & -\frac{6x_3}{x_2} \end{bmatrix} \begin{matrix} \frac{\partial}{x_1}\left(\frac{3x_3^2}{x_2}\right) \\ \frac{\partial}{x_2}\left(\frac{x_3^2}{x_2}\right) \\ \frac{\partial}{x_3}\left(\frac{x_3^2}{x_2}\right) \end{matrix} \nabla^2_{\boldsymbol{x}} f(\boldsymbol{x}_0) = \begin{bmatrix} 2 & 1 & 0 \\ 1 & \frac{2[\boldsymbol{x}_0]_3^3}{[\boldsymbol{x}_0]_2^3} & -\frac{3[\boldsymbol{x}_0]_3^2}{[\boldsymbol{x}_0]_2^2} \\ 0 & -\frac{3[\boldsymbol{x}_0]_3^2}{[\boldsymbol{x}_0]_2^2} & -\frac{6[\boldsymbol{x}_0]_3}{[\boldsymbol{x}_0]_2} \end{bmatrix} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & \frac{2 \cdot 125}{3^3} & -\frac{3 \cdot 5^2}{3^2} \\ 0 & -\frac{3 \cdot 5^2}{3^2} & -\frac{6 \cdot 5}{3} \end{bmatrix} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & \frac{250}{27} & -\frac{25}{3} \\ 0 & -\frac{25}{3} & -10 \end{bmatrix}$$

Hessian (Geometric Interpretation). Taylor Series

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a multivariate function, at consider the reference point $\boldsymbol{x}_0 \in \mathbb{R}^n$

$$\text{Tangent plane } = f(\boldsymbol{x}_0) + \nabla_{\boldsymbol{x}}^{\mathsf{T}} f(\boldsymbol{x}_0)(\boldsymbol{x} - \boldsymbol{x}_0)$$

Locally around point $\boldsymbol{x}_0 \in \mathbb{R}^n$ tangent plane gives *linear approximation* of the function:



$$\underset{\text{order surface}}{\text{Tangent second}} = f(\boldsymbol{x}_0) + \nabla_{\boldsymbol{x}}^{\mathsf{T}} f(\boldsymbol{x}_0)(\boldsymbol{x} - \boldsymbol{x}_0) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_0)^{\mathsf{T}} \nabla_{\boldsymbol{x}}^2 f(\boldsymbol{x}_0)(\boldsymbol{x} - \boldsymbol{x}_0)$$

Locally around point $\boldsymbol{x}_0 \in \mathbb{R}^n$ tangent second order surface gives *quadratic approximation* of the function:

TAYLOR SERIES:

Similarly, adding higher order derivatives we can approximate "good" functions at any point:

$$f(\boldsymbol{x}) = f(\boldsymbol{x}_0) + \nabla_{\boldsymbol{x}}^{\mathsf{T}} f(\boldsymbol{x}_0)(\boldsymbol{x} - \boldsymbol{x}_0) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_0)^{\mathsf{T}} \nabla_{\boldsymbol{x}}^2 f(\boldsymbol{x}_0)(\boldsymbol{x} - \boldsymbol{x}_0) + \ldots + \frac{\mathrm{D}^k f(\boldsymbol{x}_0)}{k!}(\boldsymbol{x} - \boldsymbol{x}_0)^k + \ldots$$

linear approximation

quadratic approximation

High-order Differentiation operator

How it works:

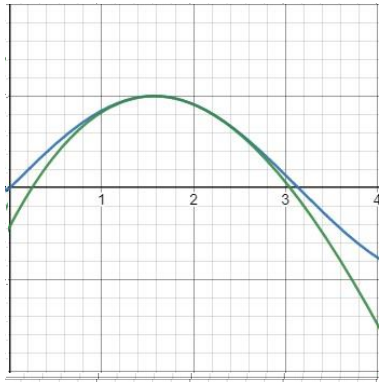
$$f(x) = \sin x$$

$$x_0 = 1.8$$

High order approximation

$$g_3(x) = \sin 1.8 + (\sin 1.8)'(x - 1.8) + \frac{1}{2}(x - 1.8)^2(\sin 1.8)'' + \frac{1}{3!}(x - 1.8)^3(\sin 1.8)'''$$

second order approximation



Lets have 10 min break

Gradient Extensions (I)

Let $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ be a multivariate vector function:

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ \vdots \\ f_m(\boldsymbol{x}) \end{bmatrix}$$

each component of $f_j(\boldsymbol{x})$ is a multivariate function itself: $f_j : \mathbb{R}^n \to \mathbb{R}$

We can define the notion of gradient for $\boldsymbol{f}(\boldsymbol{x})$: $\nabla_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) =$

$$\in \mathbb{R}^{m \times n}$$

$$\begin{bmatrix} \frac{\partial f_1(\boldsymbol{x})}{\partial x_1} & \frac{\partial f_1(\boldsymbol{x})}{\partial x_2} & \cdots & \frac{\partial f_1(\boldsymbol{x})}{\partial x_n} \\ \frac{\partial f_2(\boldsymbol{x})}{\partial x_1} & \frac{\partial f_2(\boldsymbol{x})}{\partial x_2} & \cdots & \frac{\partial f_2(\boldsymbol{x})}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m(\boldsymbol{x})}{\partial x_1} & \frac{\partial f_m(\boldsymbol{x})}{\partial x_2} & \cdots & \frac{\partial f_m(\boldsymbol{x})}{\partial x_n} \end{bmatrix} \begin{matrix} \nabla_{\boldsymbol{x}}^{\mathsf{T}} f_1(\boldsymbol{x}) \\ \nabla_{\boldsymbol{x}}^{\mathsf{T}} f_2(\boldsymbol{x}) \\ \\ \nabla_{\boldsymbol{x}}^{\mathsf{T}} f_m(\boldsymbol{x}) \end{matrix}$$

Jacobian

Example: $\boldsymbol{f}(x_1, x_2, x_3) = \begin{bmatrix} x_1^2 + x_2 x_3 \\ x_3 - x_1^2 x_2 \end{bmatrix} \quad \boldsymbol{x}_0 = [2, -1, 0]^{\mathsf{T}}$

$$\nabla_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial}{\partial x_1}(x_1^2 + x_2 x_3) & \frac{\partial}{\partial x_2}(x_1^2 + x_2 x_3) & \frac{\partial}{\partial x_3}(x_1^2 + x_2 x_3) \\ \frac{\partial}{\partial x_1}(x_3 - x_1^2 x_2) & \frac{\partial}{\partial x_2}(x_3 - x_1^2 x_2) & \frac{\partial}{\partial x_3}(x_3 - x_1^2 x_2) \end{bmatrix} = \begin{bmatrix} 2x_1 & x_3 & x_2 \\ -2x_1 x_2 & -x_1^2 & 1 \end{bmatrix}$$

$$\nabla_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}_0) = \begin{bmatrix} 2 \cdot 2 & 0 & -1 \\ -2 \cdot (2 \cdot (-1)) & -(2)^2 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 0 & -1 \\ 4 & -4 & 1 \end{bmatrix}$$

Gradient Extensions (II)

Let $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ be a function of a matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$:

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix}$$

The gradient of $f(\boldsymbol{X})$ with respect to matrix :

$$\nabla_{\boldsymbol{X}} f(\boldsymbol{X}) = \begin{bmatrix} \frac{\partial f(\boldsymbol{X})}{\partial x_{11}} & \frac{\partial f(\boldsymbol{X})}{\partial x_{12}} & \dots & \frac{\partial f(\boldsymbol{X})}{\partial x_{1n}} \\ \frac{\partial f(\boldsymbol{X})}{\partial x_{21}} & \frac{\partial f(\boldsymbol{X})}{\partial x_{22}} & \dots & \frac{\partial f(\boldsymbol{X})}{\partial x_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f(\boldsymbol{X})}{\partial x_{m1}} & \frac{\partial f(\boldsymbol{X})}{\partial x_{m2}} & \dots & \frac{\partial f(\boldsymbol{X})}{\partial x_{mn}} \end{bmatrix} \in \mathbb{R}^{m\times n}$$

Example: $f(\boldsymbol{X}) = \mathrm{Tr}(\boldsymbol{X}^\mathsf{T}\boldsymbol{X}), \quad \boldsymbol{X}_0 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & -4 & 5 \end{bmatrix}$

Consider function carefully:

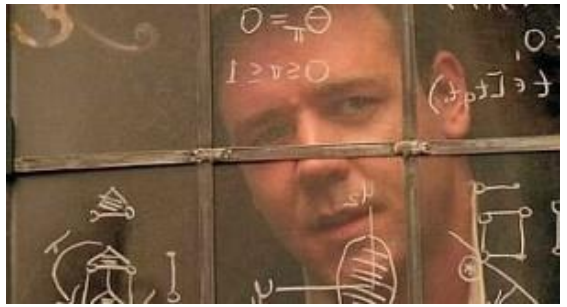
$$\nabla_{\boldsymbol{X}} f(\boldsymbol{X}) = \begin{bmatrix} 2x_{11} & 2x_{12} & \dots & 2x_{1n} \\ 2x_{21} & 2x_{22} & \dots & 2x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 2x_{m1} & 2x_{m2} & \dots & 2x_{mn} \end{bmatrix} = 2\boldsymbol{X}$$

$\boldsymbol{X}^\mathsf{T}$ $\boldsymbol{X}$

$$\nabla_{\boldsymbol{X}} f(\boldsymbol{X}_0) = 2\boldsymbol{X}_0 = \begin{bmatrix} 2 & 0 & -2 \\ 4 & -8 & 10 \end{bmatrix}$$

Exercise:

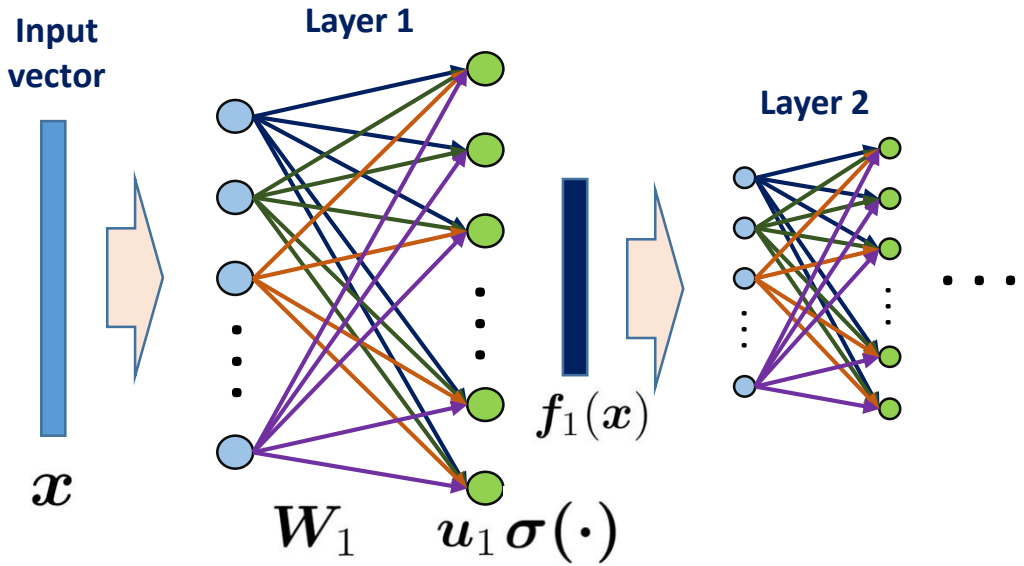
Show: $\nabla_{\boldsymbol{X}} \boldsymbol{a}^\mathsf{T}\boldsymbol{X}\boldsymbol{b} = \boldsymbol{a}\boldsymbol{b}^\mathsf{T}, \quad \boldsymbol{a} \in \mathbb{R}^m, \boldsymbol{b} \in \mathbb{R}^n$



ML Application (Backpropagation I)

For training deep neural network models, the backpropagation algorithm is an efficient way to compute the gradient of an error function with respect to the parameters of the model.

Consider feedforward computation of NN:



repeat

Step 1: Provide input vector $\boldsymbol{x}$ to the first layer of **NN**

Step 2: Input multiplied by layer weights $\boldsymbol{W}_1 : \boldsymbol{x} \rightarrow \boldsymbol{W}_1^{\mathsf{T}}\boldsymbol{x}$

Step 3: Add bias parameters $\boldsymbol{u}_1$: $\boldsymbol{x} \rightarrow \boldsymbol{W}_1^{\mathsf{T}}\boldsymbol{x} + \boldsymbol{u}_1$

Step 4: Pass through activation function $\boldsymbol{\sigma}(\cdot)$: $\boldsymbol{x} \rightarrow \boldsymbol{\sigma}(\boldsymbol{W}_1^{\mathsf{T}}\boldsymbol{x} + \boldsymbol{u}_1) = \boldsymbol{f}_1(\boldsymbol{x})$

Step 5: Use output $\boldsymbol{f}_1(\boldsymbol{x})$ as input for the next layer of **NN**

The computation of K-layer NN with weights $\{\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_K\}$ and biases $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_K\}$ allows iterative form:

$$\boldsymbol{f}_0 = \boldsymbol{x}$$

$$\boldsymbol{f}_i = \boldsymbol{\sigma}\left(\boldsymbol{W}_{i-1}^{\mathsf{T}}\boldsymbol{f}_{i-1} + \boldsymbol{u}_{i-1}\right), \quad i = 1, \ldots, K$$

Group unknown parameter associated with each layer of NN: $\boldsymbol{\Theta}_1 = \{\boldsymbol{W}_1,\ \boldsymbol{u}_1\}, \boldsymbol{\Theta}_2 = \{\boldsymbol{W}_2, \boldsymbol{u}_2\},\ \ldots,\ \boldsymbol{\Theta}_K = \{\boldsymbol{W}_K, \boldsymbol{u}_K\}$

or more compactly:

$$\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \ldots, \boldsymbol{\Theta}_K\}$$

ML Application (Backpropagation II)

Training **NN** constitutes in finding parameters $\boldsymbol{\Theta}$ which best fits observed data $\mathcal{D} = \{\boldsymbol{X}_{\text{data}}, \boldsymbol{y}_{\text{data}}\}$.

The fitness measure is defined via loss function: $\mathcal{L}(\boldsymbol{\Theta}) = ||\boldsymbol{y}_{\text{data}} - \boldsymbol{f}_K(\boldsymbol{\Theta}, \boldsymbol{X}_{\text{data}})||^2$

To obtain the gradients with respect to the parameter $\boldsymbol{\Theta}$, we compute the partial derivatives of $\mathcal{L}(\boldsymbol{\Theta})$ with respect to the parameters $\boldsymbol{\Theta}_j = \{\boldsymbol{W}_j, \boldsymbol{u}_j\}$ of each layer $j = 1, \ldots, K$. The chain rule allows us to determine the partial derivatives as:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Theta}_K} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{f}_K} \frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{\Theta}_K}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Theta}_{K-1}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{f}_K} \frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{f}_{K-1}} \frac{\partial \boldsymbol{f}_{K-1}}{\partial \boldsymbol{\Theta}_{K-1}}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Theta}_{K-2}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{f}_K} \frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{f}_{K-1}} \frac{\partial \boldsymbol{f}_{K-1}}{\partial \boldsymbol{f}_{K-2}} \frac{\partial \boldsymbol{f}_{K-2}}{\partial \boldsymbol{\Theta}_{K-2}}$$

$$\vdots$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Theta}_j} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{f}_K} \frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{f}_{K-1}} \cdots \frac{\partial \boldsymbol{f}_{j+1}}{\partial \boldsymbol{f}_j} \frac{\partial \boldsymbol{f}_j}{\partial \boldsymbol{\Theta}_j}$$

The **orange terms** are partial derivatives of the output of a layer with respect to its inputs.

The **blue terms** are partial derivatives of the output of a layer with respect to its parameters.

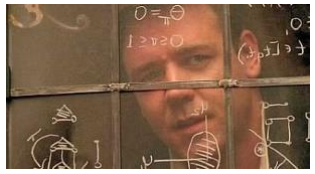
If we calculated $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Theta}_{j+1}}$ then most of its computation can be reused for partial derivative $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Theta}_j}$

Indeed **green boxes** shows the additional terms we need to compute.

These additional factors can be easily computed from our iterative expressions:

$$\boldsymbol{f}_0 = \boldsymbol{x}$$
$$\boldsymbol{f}_i = \boldsymbol{\sigma}\left(\boldsymbol{W}_{i-1}^{\mathsf{T}} \boldsymbol{f}_{i-1} + \boldsymbol{u}_{i-1}\right), \quad i = 1, \ldots, K$$

Exercise:

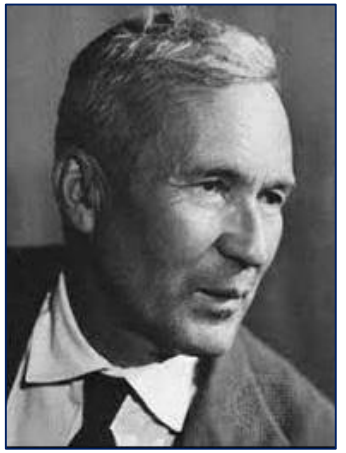


Using derivatives of vector function,

calculate: $\frac{\partial \boldsymbol{f}_{j+1}}{\partial \boldsymbol{f}_j}$ and $\frac{\partial \boldsymbol{f}_j}{\partial \boldsymbol{\Theta}_j}$

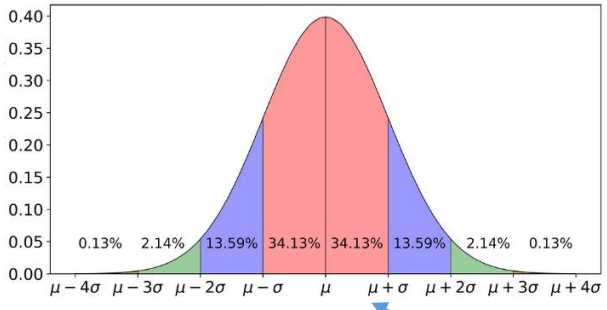
Probability Theory

- Probability and Random Variables.
- Discrete and Continuous Probabilities.
- Sum Rule, Sum Product, Bayes Theorem.
- Summary Statistics, Independence.
- Gaussian Distribution.



Andrey Nikolaevich Kolmogorov
(1903-1987)

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$



Source: Google Images

Probability Space

The *probability space* models a real-world process (referred to as an experiment)with random outcomes. It consists of three components

THE SAMPLE SPACE Ω

The sample space is the set of all possible elementary outcomes of the experiment.

Example:

Coin toss: $\Omega =$ 

Roll a dice: $\Omega =$ 

THE EVENT SPACE $\mathcal{A}$

The event space is the space of potential results of the experiment.
Each event $A \in \mathcal{A}$ consists of elementary outcomes, i.e. $A \subseteq \Omega$
The set of all events is a power set of Ω , and denotes: $\mathcal{A} = 2^{\Omega}$

Event of getting **TAIL** in a coin-flip: $A =$ 

Event of getting **even** number in a roll dice: $B =$ 

THE PROBABILITY $\mathbb{P}$

With each event $A \in \mathcal{A}$, we associate a number $\mathbb{P}(A) \in [0,1]$ that measures the probability or degree of belief that the event will occur. $\mathbb{P}(A)$ is called the probability of A

Probability of getting **TAIL** in a coin-flip: $\mathbb{P}(A) = \frac{1}{2}$

Probability of getting **even** number in a roll dice: $\mathbb{P}(B) = \frac{1}{2}$

Remark: chance of … = the probability of … The chance of … is 33% = The probability of … is 0.33

Random Variable

In machine learning, we often don't refer to the probability space, but instead refer to probabilities on quantities of interest denoted by $\mathcal{T}$

A *random variable* $\mathrm{A} : \boldsymbol{\Omega} \to \mathcal{T}$ is a mapping that takes an element $\omega \in \boldsymbol{\Omega}$ and returns a an element $x \in \mathcal{T}$

Example:

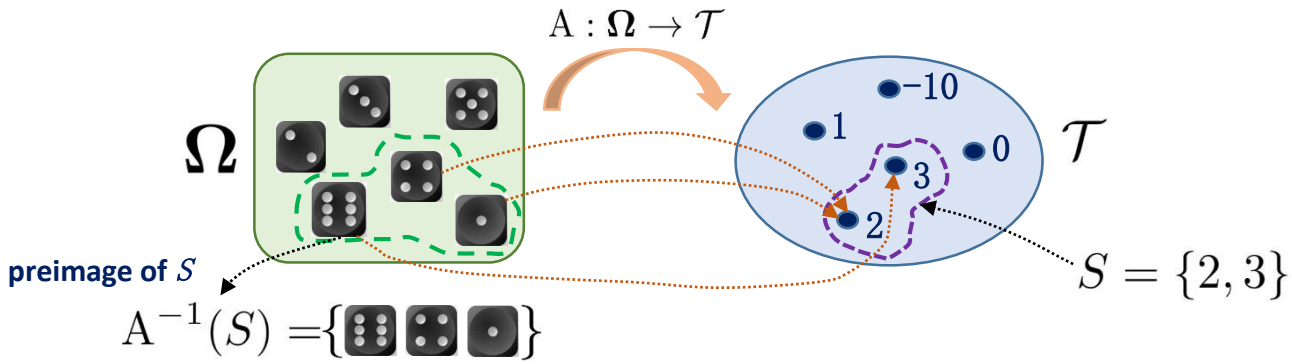
Random variable defined on a set of roll dices:

$\mathrm{A}(\square) = 2 \quad \mathrm{A}(\square) = 1 \quad \mathrm{A}(\square) = -10$

$\mathrm{A}(\square) = 2 \quad \mathrm{A}(\square) = 0 \quad \mathrm{A}(\square) = 3$

In this case: $\mathcal{T} = \{2, 3, -10, 0, 1\}$

For any subset $S \subseteq \mathcal{T}$ we associate $\mathbb{P}_{\mathrm{A}}(S) \in [0, 1]$ (the probability) to a particular event occurring corresponding to random variable A



$$\mathbb{P}_{\mathrm{A}}(S) = \mathbb{P}(\mathrm{A}^{-1}(S)) = \mathbb{P}(\square\square\square) = \mathbb{P}(\square) + \mathbb{P}(\square) + \mathbb{P}(\square) = \frac{1}{6} + \frac{1}{6} + \frac{1}{6} = \frac{1}{2}$$

Random variable A defines probability distribution $\mathbb{P}_{\mathrm{A}}(S) \in [0, 1]$ on the target space $\mathcal{T}$.

Discrete Probabilities

Consider the case when we can enumerate all elements in the target set $\mathcal{T}$.
In other words, $\mathcal{T} = \{x_1, x_2, \ldots, x_k, \ldots\}$

Random variable A defines a probability distribution on $\mathcal{T}$, such that for any $x_i \in \mathcal{T}$ its probability $\mathbb{P}_{\mathrm{A}}(x_i) = \mathbb{P}(\mathrm{A} = x_i) = p_i$.
Hence, discrete random variable is characterized by *a probability mass function* $\mathbb{P}_{\mathrm{A}}() = \{p_1, p_2, \ldots\}$, such that $p_i \geq 0$ and $\sum_i p_i = 1$

Examples:

$\boxed{\mathrm{A} = \mathrm{Bernoulli(p)}}$

Consider $\mathcal{T} = \{0, 1\}$ and :

$$\mathbb{P}_{\mathrm{A}}(1) = \mathbb{P}(\mathrm{A} = 1) = p, \quad \mathbb{P}_{\mathrm{A}}(0) = \mathbb{P}(\mathrm{A} = 0) = 1 - p$$

$\boxed{\mathrm{A} = \mathrm{Poisson}(\lambda)}$

Consider $\mathcal{T} = \{0, 1, 2, \ldots\}$ and :

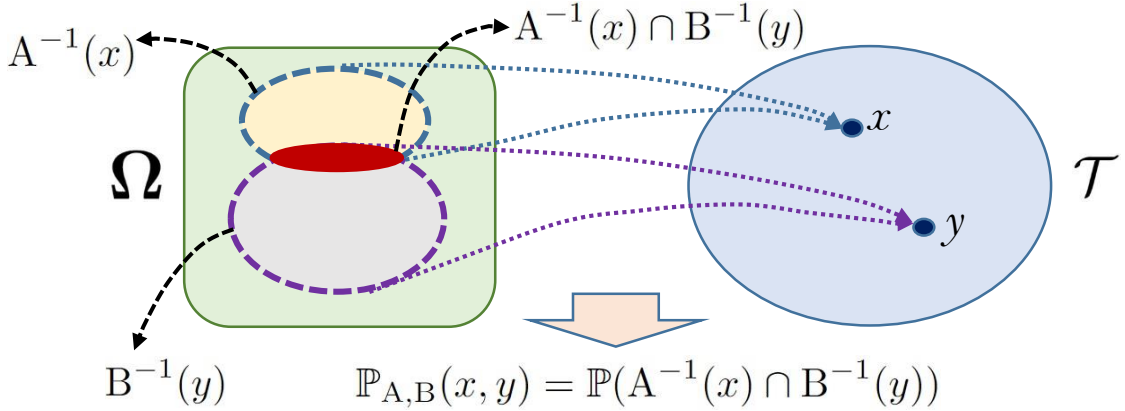
$$\mathbb{P}_{\mathrm{A}}(k) = \mathbb{P}(\mathrm{A} = k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad \text{with} \quad \lambda \in (0, +\infty)$$

JOINT PROBABILITY FOR RANDOM VARIABLES:

Discrete random variables: $\mathrm{A} : \boldsymbol{\Omega} \to \mathcal{T}$ and $\mathrm{B} : \boldsymbol{\Omega} \to \mathcal{T}$, and $x \; y \in \mathcal{T}$.

Joint Probability of $\mathrm{A} = x$ and $\mathrm{B} = y$ is defined as:

$$\mathbb{P}_{\mathrm{A,B}}(x, y) = \mathbb{P}(\omega \in \boldsymbol{\Omega}, \text{ such that } \mathrm{A}(\omega) = x \text{ and } \mathrm{B}(\omega) = y)$$



$\mathrm{A}(\square)=2$ $\mathrm{A}(\square)=1$ $\mathrm{B}(\square)=1$ $\mathrm{B}(\square)=2$
$\mathrm{A}(\square)=2$ $\mathrm{A}(\square)=-10$ and $\mathrm{B}(\square)=2$ $\mathrm{B}(\square)=3$, $x = 2$, $y = 1$
$\mathrm{A}(\square)=0$ $\mathrm{A}(\square)=3$ $\mathrm{B}(\square)=0$ $\mathrm{B}(\square)=3$

$$\mathrm{A}^{-1}(2) = \{\square\,\square\} \text{ and } \mathrm{B}^{-1}(1) = \{\square\}$$

$$\mathrm{A}^{-1}(2) \cap \mathrm{B}^{-1}(1) = \{\square\} \quad \text{Hence: } \mathbb{P}_{\mathrm{A,B}}(2, 1) = \mathbb{P}(\square) = \frac{1}{6}$$

Discrete Probabilities

Consider the case when we can enumerate all elements in the target set $\mathcal{T}$.
In other words, $\mathcal{T} = \{x_1, x_2, \ldots, x_k, \ldots\}$

Random variable A defines a probability distribution on $\mathcal{T}$, such that for any $x_i \in \mathcal{T}$ its probability $\mathbb{P}_{\mathrm{A}}(x_i) = \mathbb{P}(\mathrm{A} = x_i) = p_i$.
Hence, discrete random variable is characterized by *a probability mass function* $\mathbb{P}_{\mathrm{A}}() = \{p_1, p_2, \ldots\}$, such that $p_i \geq 0$ and $\sum_i p_i = 1$

Examples:

$\boxed{\mathrm{A} = \mathrm{Bernoulli(p)}}$

Consider $\mathcal{T} = \{0, 1\}$ and :

$$\mathbb{P}_{\mathrm{A}}(1) = \mathbb{P}(\mathrm{A} = 1) = p, \quad \mathbb{P}_{\mathrm{A}}(0) = \mathbb{P}(\mathrm{A} = 0) = 1 - p$$

$\boxed{\mathrm{A} = \mathrm{Poisson}(\lambda)}$

Consider $\mathcal{T} = \{0, 1, 2, \ldots\}$ and :

$$\mathbb{P}_{\mathrm{A}}(k) = \mathbb{P}(\mathrm{A} = k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad \text{with} \quad \lambda \in (0, +\infty)$$

CONDITIONAL PROBABILITY FOR RANDOM VARIABLES:

Discrete random variables: $\mathrm{A} : \boldsymbol{\Omega} \to \mathcal{T}$ and $\mathrm{B} : \boldsymbol{\Omega} \to \mathcal{T}$, and $x \; y \in \mathcal{T}$.

Conditional Probability of $\mathrm{B} = y$ condition on $\mathrm{A} = x$ is defined as:

$$\mathbb{P}_{\mathrm{B|A}}(y, x) = \mathbb{P}(\omega \in \mathrm{A}^{-1}(\boldsymbol{x}), \text{ such that } \mathrm{B}(\omega) = y)$$

$$\mathbb{P}_{\mathrm{A,B}}(x, y) = \mathbb{P}(\omega \in \boldsymbol{\Omega}, \text{ such that } \mathrm{A}(\omega) = x \text{ and } \mathrm{B}(\omega) = y)$$

Notice, in contrast with joint probability, here we are looking for elementary outcomes giving $\mathrm{B}(\omega) = y$ only in the preimage $\mathrm{A}^{-1}(\boldsymbol{x})$, rather than the whole set $\boldsymbol{\Omega}$.

$$\mathrm{A}^{-1}(2) = \{\text{⚂ ⚃}\} \text{ and } \mathrm{B}^{-1}(1) = \{\text{⚀}\}$$

Hence:

$$\mathbb{P}_{\mathrm{B|A}}(1, 2) = \mathbb{P}(\text{⚀ out of ⚂ ⚃}) = \frac{1}{2}$$

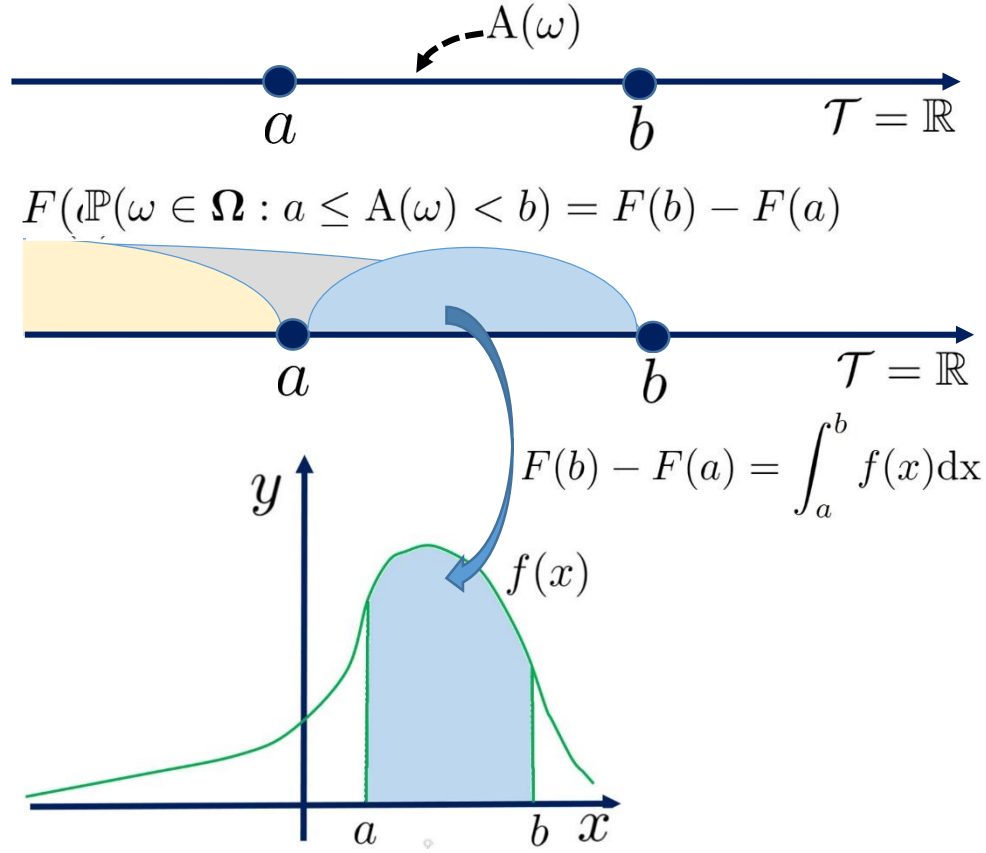
Continuous Probabilities

Consider now when the target set is not countable, for example $\mathcal{T} = \mathbb{R}$.

We interested in the probability that a random variable A is in the interval, i.e. $\mathbb{P}(\omega \in \mathbf{\Omega} : a \leq \mathrm{A}(\omega) < b)$ for $a, b \in \mathbb{R}$.

To compute it, we introduce *cumulative distribution function (cdf)*:

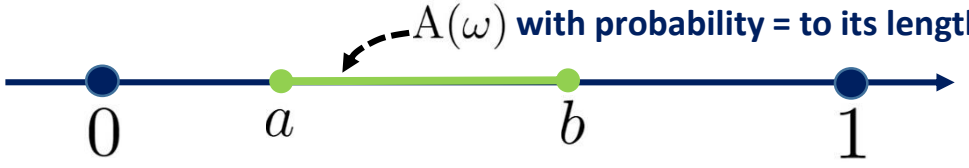
$$F(x) = \mathbb{P}(\omega \in \mathbf{\Omega} : \mathrm{A}(\omega) \leq x)$$



A function $f : \mathbb{R} \to \mathbb{R}$ is called *probability density function (pdf)* if:

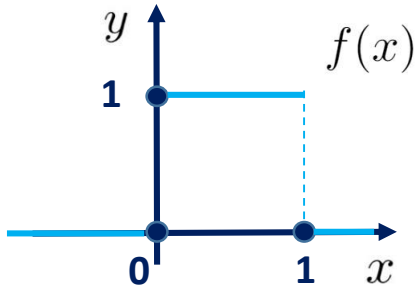
- Non-negative: $f(x) \geq 0$ for any $x \in \mathbb{R}$
- Integrable: $\int_{\mathbb{R}} f(x)\mathrm{dx} = 1$

Example: $\mathrm{A} = \mathrm{Uniform}\ ([0,1])$

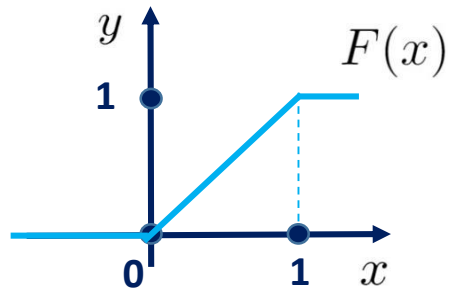


For any interval $(a, b) \subseteq [0, 1]$ we have: $\mathbb{P}(\mathrm{A} \in (a, b)) = b - a$

Probability density function:



Cumulative distribution function:



Sum Rule

Unify notation for discrete and continuous random variables

DISCRETE RANDOM VARIABLES: **CONTINUOUS RANDOM VARIABLES:**

Probability mass functions

$$\begin{aligned} \mathbb{P}_{\mathrm{A}}(\boldsymbol{x}) &= p(\boldsymbol{x}) &&= p_{\mathrm{A}}(\boldsymbol{x}) \\ \mathbb{P}_{\mathrm{A},\mathrm{B}}(\boldsymbol{x},\boldsymbol{y}) &= p(\boldsymbol{x},\boldsymbol{y}) &&= p_{\mathrm{A},\mathrm{B}}(\boldsymbol{x},\boldsymbol{x}) \\ \mathbb{P}_{\mathrm{A}|\mathrm{B}}(\boldsymbol{x}|\boldsymbol{y}) &= p(\boldsymbol{x}|\boldsymbol{y}) &&= p_{\mathrm{A}|\mathrm{B}}(\boldsymbol{x}|\boldsymbol{y}) \end{aligned}$$

Probability density functions

SUM RULE:

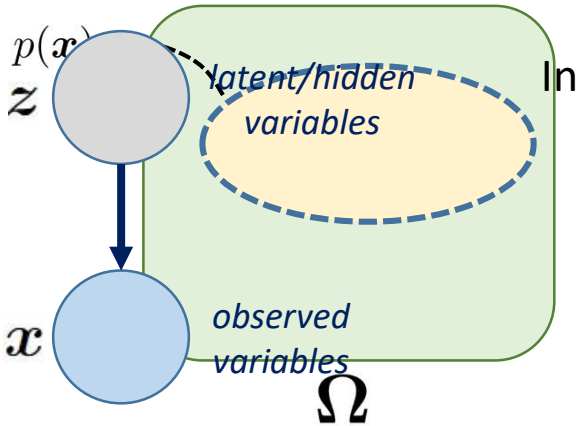
Consider random variables:
$\mathrm{A} : \boldsymbol{\Omega} \to \mathcal{T}_{\mathrm{A}}$ and $\mathrm{B} : \boldsymbol{\Omega} \to \mathcal{T}_{\mathrm{B}}$

$$p(\boldsymbol{x}) = \begin{cases} \sum_{\boldsymbol{y} \in \mathcal{T}_{\mathrm{B}}} p(\boldsymbol{x},\boldsymbol{y}) & \text{if } \mathcal{T}_{\mathrm{B}} \text{ is discrete} \\ \int_{\mathcal{T}_{\mathrm{B}}} p(\boldsymbol{x},\boldsymbol{y}) \mathrm{d}\boldsymbol{y} & \text{if } \mathcal{T}_{\mathrm{B}} \text{ is continuous} \end{cases}$$

$p(\boldsymbol{x})$: marginal distribution

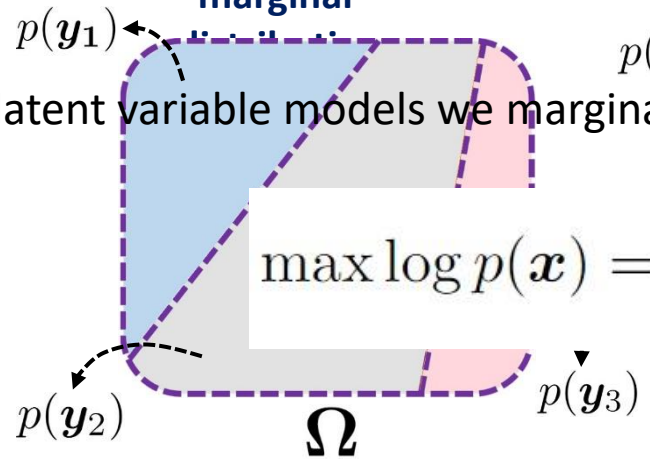
sum out (or integrate out) the set of states $\boldsymbol{y}$ of the random variable B
This operation is called *marginalisation*

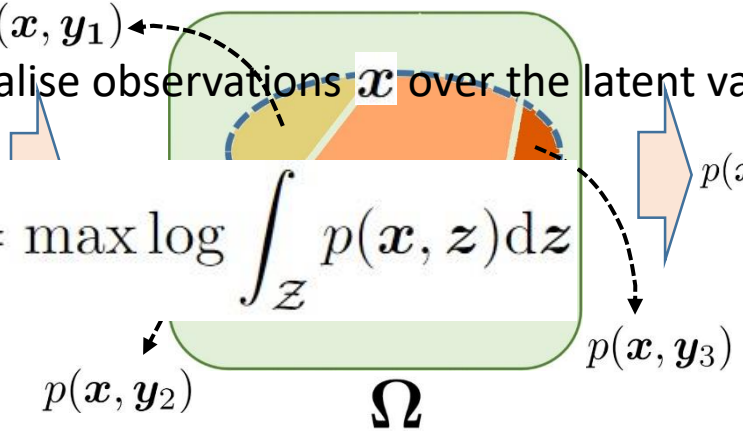
ML application:



In latent variable models we marginalise observations $\boldsymbol{x}$ over the latent variables $\boldsymbol{z}$:

$$\max \log p(\boldsymbol{x}) = \max \log \int_{\mathcal{Z}} p(\boldsymbol{x},\boldsymbol{z}) \mathrm{d}\boldsymbol{z}$$





$$p(\boldsymbol{x}) = p(\boldsymbol{x},\boldsymbol{y}_1) + p(\boldsymbol{x},\boldsymbol{y}_2) + p(\boldsymbol{x},\boldsymbol{y}_3)$$

Product Rule

Unify notation for discrete and continuous random variables

DISCRETE RANDOM VARIABLES: **CONTINUOUS RANDOM VARIABLES:**

Probability mass functions

$$\begin{aligned} \mathbb{P}_{\mathrm{A}}(\boldsymbol{x}) &= p(\boldsymbol{x}) &&= p_{\mathrm{A}}(\boldsymbol{x}) \\ \mathbb{P}_{\mathrm{A,B}}(\boldsymbol{x},\boldsymbol{y}) &= p(\boldsymbol{x},\boldsymbol{y}) &&= p_{\mathrm{A,B}}(\boldsymbol{x},\boldsymbol{x}) \\ \mathbb{P}_{\mathrm{A|B}}(\boldsymbol{x}|\boldsymbol{y}) &= p(\boldsymbol{x}|\boldsymbol{y}) &&= p_{\mathrm{A|B}}(\boldsymbol{x}|\boldsymbol{y}) \end{aligned}$$

Probability density functions

PRODUCT RULE:

Consider random variables:
$\mathrm{A} : \boldsymbol{\Omega} \to \mathcal{T}_{\mathrm{A}}$ and $\mathrm{B} : \boldsymbol{\Omega} \to \mathcal{T}_{\mathrm{B}}$

$$p(\boldsymbol{x},\boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})$$

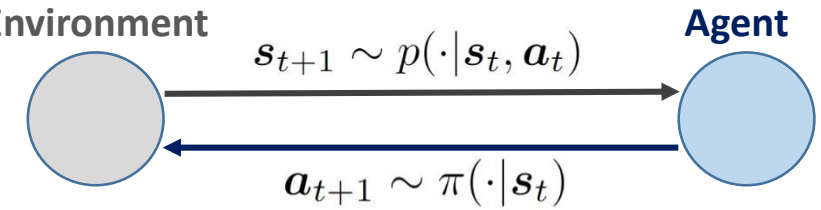
Every joint distribution of two random variables can be factorized of two other distributions.
The two factors are the marginal distribution of the first random variable $p(\boldsymbol{x})$, and the conditional distribution of the second random variable given the first $p(\boldsymbol{y}|\boldsymbol{x})$

ML application:

In *Reinforcement Learning* the probability of state action trajectory:

$\boldsymbol{\tau} = (\boldsymbol{s}_0, \boldsymbol{a}_0, \boldsymbol{s}_1, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_{H-1}, \boldsymbol{s}_H)$ is factorised using of product rule:

$$p(\boldsymbol{\tau}) = p(\boldsymbol{s}_0, \boldsymbol{a}_0, \boldsymbol{s}_1, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_{H-1}, \boldsymbol{s}_H) = p(\boldsymbol{s}_0) \prod_{t=1}^{H} p(\boldsymbol{s}_t|\boldsymbol{a}_{t-1}, \boldsymbol{s}_{t-1})\pi(\boldsymbol{a}_{t-1}|\boldsymbol{s}_{t-1})$$



Bayes Theorem

In machine learning and Bayesian statistics, we are often interested in making inferences of *unobserved* random variable x, given that we have observed other random variable y, related to x

$p(x)$ - our initial guess about hidden variable x

$p(y|x)$ - relationship between x and observed y

Can we draw any conclusions about x given our observations y?

Thomas Bayes
(1701-1761)

YES – using Bayes Formula

Using product rule:

$$p(x,y) = p(y|x)p(x)$$

Similarly:

$$p(x,y) = p(y)p(x|y)$$

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

$p(x)$ - *prior distribution* over hidden variable x

$p(y|x)$ - *likelihood*, describing conditional probability of y given x.

$p(x|y)$ - *posterior distribution*, which is the quantity of interest, as it reflects conditional probability of x given y.

$p(y)$ - *marginal likelihood/evidence* "weighted average" of likelihood $p(y|x)$, with weighs defined by prior $p(x)$

- **using sum rule**
- **and then product rule**

$$p(y) = \sum_x p(y,x) = \sum_x p(y|x)p(x)$$

Summary Statistics (Expected Value)

Weighted average expression over distribution is a special case of *statistics*. A statistic of a random variable is a *deterministic function* of that random variable. They provide a general view on the behavior of a random variable.

Two most popular statistics of random variable are *mean* and *variance* .

MEAN (MATHEMATICAL EXPECTATION, EXPECTED VALUE, FIRST MOMENT)

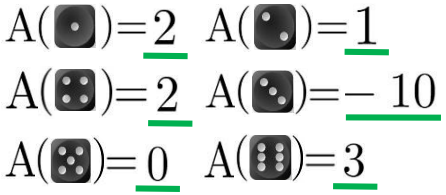
CASE I (Discrete):

Let $\mathrm{A} : \boldsymbol{\Omega} \to \mathcal{T}$ be a discrete random variable with the target set $\mathcal{T} = \{x_1, x_2, \ldots, x_k, \ldots\}$ and the probability mass function $p(\mathrm{A} = x_i) = p_i$ for $i = 1, 2, \ldots$. The expected value of A is a sum:

$$\mathbb{E}[\mathrm{A}] = \sum_i x_i p_i$$

Example:

Consider *a fair dice* (all sides are equally probable = $\frac{1}{6}$) and random variable A:



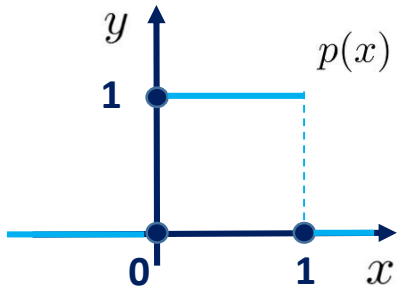
$$\mathbb{E}[\mathrm{A}] = 2 \cdot \frac{1}{6} + 1 \cdot \frac{1}{6} + (-10) \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} + 0 \cdot \frac{1}{6} + 3 \cdot \frac{1}{6} = -\frac{1}{3}$$

CASE II (Continuous):

Let $\mathrm{A} : \boldsymbol{\Omega} \to \mathcal{T}$ be a continuous random variable with the target Set $\mathcal{T} = \mathbb{R}^d$ and the probability density function $p(\boldsymbol{x})$. The mean value of A is given by an integral (if it exist):

$$\mathbb{E}[\mathrm{A}] = \int_{\mathcal{T}} \boldsymbol{x} p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$$

Consider $\mathrm{A} = \mathrm{Uniform}\ ([0,1])$ with pdf $p(x)$:



$$\mathbb{E}[\mathrm{A}] = \int_{-\infty}^{+\infty} x p(x) \mathrm{d}x = \int_0^1 x \mathrm{d}x = \frac{x^2}{2}\Big|_0^1 = \frac{1}{2}$$

Summary Statistics (Variance)

Weighted average expression over distribution is a special case of *statistics*. A statistic of a random variable is a *deterministic function* of that random variable. They provide a general view on the behavior of a random variable.

Two most popular statistics of random variable are *mean* and *variance* .

VARIANCE (DISPERSION, SECOND MOMENT)

CASE I (Discrete):

Let $\mathrm{A} : \Omega \to \mathcal{T}$ be a discrete random variable with the target set $\mathcal{T} = \{x_1, x_2, \ldots, x_k, \ldots\}$ and the probability mass function $p(\mathrm{A} = x_i) = p_i$ for $i = 1, 2, \ldots$. The variance of A is a sum:

$$\mathbb{V}[\mathrm{A}] = \mathbb{E}[(\mathrm{A} - \mathbb{E}[\mathrm{A}])^2] = \sum_i p_i (x_i - \mathbb{E}[\mathrm{A}])^2$$

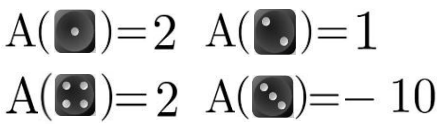
CASE II (Continuous):

Let $\mathrm{A} : \Omega \to \mathcal{T}$ be a continuous random variable with the target Set $\mathcal{T} = \mathbb{R}^d$ and the probability density function $p(\boldsymbol{x})$. Then, the variance of A is given by an integral (if it exist):

$$\mathbb{V}[\mathrm{A}] = \int_{\mathcal{T}} ||\boldsymbol{x} - \mathbb{E}[\mathrm{A}]||_2^2 p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$$

Example:

Consider *a fair dice* (all sides are equally probable = $\frac{1}{6}$) and random variable A:



We know: $\mathbb{E}[\mathrm{A}] = -\frac{1}{3}$

$$\mathbb{V}[\mathrm{A}] = 2 \cdot \frac{1}{6}\left[2 + \frac{1}{3}\right]^2 + \frac{1}{6}\left[1 + \frac{1}{3}\right]^2 + \frac{1}{6}\left[3 + \frac{1}{3}\right]^2 + \frac{1}{6}\left[0 + \frac{1}{3}\right]^2 + \frac{1}{6}\left[-10 + \frac{1}{3}\right]^2 = \frac{176}{9}$$

Consider $\mathrm{A} = \mathrm{Uniform}\,([0,1])$:

We know: $\mathbb{E}[\mathrm{A}] = \frac{1}{2}$

$$\mathbb{V}[\mathrm{A}] = \int_{-\infty}^{+\infty} \left[x - \frac{1}{2}\right]^2 p(x)\mathrm{d}x = \int_0^1 \left[x - \frac{1}{2}\right]^2 \mathrm{d}x = \left[\frac{x^3}{3} - \frac{x^2}{2} + \frac{x}{4}\right]\Big|_0^1 = \frac{1}{12}$$

Summary Statistics (ML Application)

Weighted average expression over distribution is a special case of *statistics*. A statistic of a random variable is a *deterministic function* of that random variable. They provide a general view on the behavior of a random variable.

Mean and Variance are used almost everywhere in ML:

VARIOUS ML APPLICATIONS OF MEAN/VARIANCE:

- **In Supervised Learning:**

  Training loss function for ML models is usually represented as *empirical mean*:

$$\mathcal{L}_{\text{training}}(\boldsymbol{\Theta}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{\text{tr}}}\left[(y - \text{NN}_{\boldsymbol{\Theta}}(\boldsymbol{x}))^2\right] = \frac{1}{N}\sum_{i=1}^{N}(y_i - \text{NN}_{\boldsymbol{\Theta}}(\boldsymbol{x}_i))^2$$

  Here:

  $\mathcal{D}_{\text{tr}} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$ — training data-set

  $\text{NN}_{\boldsymbol{\Theta}}(\boldsymbol{x})$ — ML model (Neural Network)

- **In Bayesian Optimisation:**

  UCB acquisition function for Bayesian Optimisation has the following form:

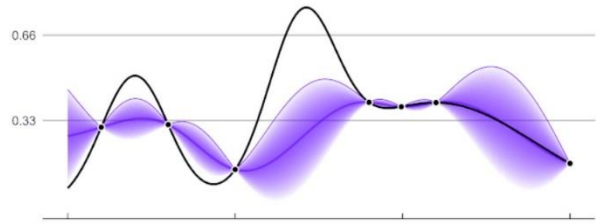
$$\alpha_{\text{UCB}}(\boldsymbol{x}) = \mu_{\text{post}}(\boldsymbol{x};\mathcal{D}_{\text{obs}}) + \beta\sigma_{\text{post}}(\boldsymbol{x};\mathcal{D}_{\text{obs}})$$

  Here:

  $\mu_{\text{post}}(\boldsymbol{x};\mathcal{D}_{\text{obs}})$ — is *posterior mean of GP model*

  $\sigma^2_{\text{post}}(\boldsymbol{x};\mathcal{D}_{\text{obs}})$ — is *posterior variance of GP model*

  $\mathcal{D}_{\text{obs}} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{M}$ — is collection of black-box observations.



Independence

Two random variables A, B are *statistically independent* if and only if:

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x})p(\boldsymbol{y})$$

Intuitively, A, B are independent if the value of one of them does not add any additional information about the value of another.

ML application:

In machine learning, we often consider problems that can be modeled as *independent and identically distributed(i.i.d.)* random variables. For example, in supervised learning, training data-set is assumed to consist of i.i.d. samples. In Reinforcement learning, for a fixed policy the sampled trajectories are i.i.d.

Examples:

Flipping a coin several times, then random variables associated with each such trial are independent.

Roll a dice (random variable A) and then sample a point $\mathrm{B} \sim \mathrm{Uniform}\ ([0,1])$. Then, A, B are independent

Roll a dice (random variable A) and then sample a point $\mathrm{B} \sim \mathrm{Uniform}\ ([0,1])$. Consider random variable

$$\mathrm{C} = \begin{cases} 2\mathrm{B} & \text{if } \mathrm{A} = \{ \text{⚅} \} \\ \frac{1}{2}\mathrm{B} & \text{otherwise} \end{cases}$$

Then, A, C are not independent.

SOME PROPERTIES OF RANDOM VARIABLES:

- If A, B are independent, then: $p(\boldsymbol{y}|\boldsymbol{x}) = p(\boldsymbol{y})$ and $p(\boldsymbol{x}|\boldsymbol{y}) = p(\boldsymbol{x})$
- If A, B are independent, then: $\mathbb{V}[\mathrm{A} + \mathrm{B}] = \mathbb{V}[\mathrm{A}] + \mathbb{V}[\mathrm{B}]$
- If A, B are independent, then: $\mathbb{E}[\mathrm{AB}] = \mathbb{E}[\mathrm{A}]\mathbb{E}[\mathrm{B}]$ and $\mathbb{E}\left[\frac{\mathrm{A}}{\mathrm{B}}\right] = \frac{\mathbb{E}[\mathrm{A}]}{\mathbb{E}[\mathrm{B}]}$
- If A, B are independent, then: $\mathbb{E}[f(\mathrm{A})g(\mathrm{B})] = \mathbb{E}[f(\mathrm{A})]\mathbb{E}[g(\mathrm{B})]$, where $f(\cdot), g(\cdot)-$ are integrable.
- If A, B are random variables (might be dependent), then: $\mathbb{E}[\alpha\mathrm{A} + \beta\mathrm{B}] = \alpha\mathbb{E}[\mathrm{A}] + \beta\mathbb{E}[\mathrm{B}]$, where $\alpha, \beta \in \mathbb{R}$ **Linearity of Expectation**

Gaussian Random Variable

The Gaussian (normal) distribution is the most well-studied probability distribution for continuous-valued random variables. It has application in many fields of ML, including diffusion models, normalizing flows, Kalman filters, Gaussian processes ,etc.

Carl Friedrich Gauss
(1777-1855)

The Gaussian random variable X can be 1-dimensional (i.e. $\mathrm{X} \in \mathbb{R}$) or $\mathrm{d}\text{-dimensional}$ (i.e. $\mathrm{X} \in \mathbb{R}^d$) and can be fully determined by its first and second moments:

1-DIMENSIONAL CASE:

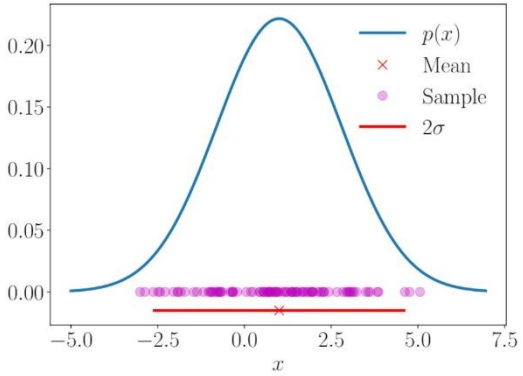
Probability density function:

$$p(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

$\mu \in \mathbb{R}$ - is expected value of X

$\sigma^2 \in \mathbb{R}$ - is variance of X

Examples*:



$$p(x) = \mathcal{N}(1, 1.69)$$

*Source: "Mathematics for Machine Learning". https://mml-book.com.

d-DIMENSIONAL CASE:

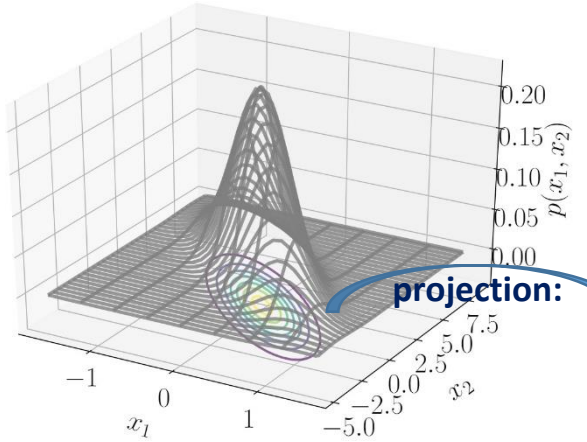
$$p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{d}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right)$$

$\boldsymbol{\mu} \in \mathbb{R}^d$ - is expected value of X

$\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ - is variance of X

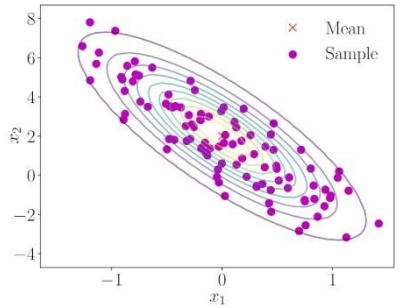
covariance matrix:

$$\boldsymbol{\Sigma} = \mathbb{E}\left[(\boldsymbol{x}-\boldsymbol{\mu})^{\mathsf{T}}(\boldsymbol{x}-\boldsymbol{\mu})\right]$$



projection:

$$p(x_1, x_2) = \mathcal{N}\left(\begin{bmatrix}0\\2\end{bmatrix}, \begin{bmatrix}0.3 & -1\\-1 & 5\end{bmatrix}\right)$$



Properties or Gaussian

The Gaussian (normal) distribution is the most well-studied probability distribution for continuous-valued random variables. It has application in many fields of ML, including diffusion models, normalizing flows, Kalman filters, Gaussian processes ,etc.

Carl Friedrich Gauss
(1777-1855)

1-DIMENSIONAL CASE:

$$p(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

d-DIMENSIONAL CASE:

$$p(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = (2\pi)^{-\frac{d}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

Sum of independent Gaussians: Let $\mathrm{X} = \mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathrm{Y} = \mathcal{N}(\mu_2, \sigma_2^2)$ be two independent Gaussian random variables. Then, for any $\alpha, \beta \in \mathbb{R}$:

$$\alpha\mathrm{X} + \beta\mathrm{Y} = \mathcal{N}(\alpha\mu_1 + \beta\mu_2, \alpha^2\sigma_1^2 + \beta^2\sigma_2^2)$$

$\boldsymbol{d}$-dimensional version for $\mathrm{X} = \mathcal{N}(\boldsymbol{\mu_1}, \boldsymbol{\Sigma_1}) \in \mathbb{R}^d$ and $\mathrm{Y} = \mathcal{N}(\boldsymbol{\mu_2}, \boldsymbol{\Sigma_2}) \in \mathbb{R}^d$:

$$\alpha\mathrm{X} + \beta\mathrm{Y} = \mathcal{N}(\alpha\boldsymbol{\mu_1} + \beta\boldsymbol{\mu_2}, \alpha^2\boldsymbol{\Sigma_1} + \beta^2\boldsymbol{\Sigma_2})$$

Linear transformation of Gaussian: Let $\mathrm{X} = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \in \mathbb{R}^d$ be a multivariate Gaussian, and $\boldsymbol{A} \in \mathbb{R}^{d\times p}$ be an arbitrary matrix. Consider random variable $\mathrm{Y} = \boldsymbol{A}\mathrm{X} \in \mathbb{R}^p$ - which is linear transformation of X by $\boldsymbol{A}$. Then: $\mathrm{Y} = \mathcal{N}(\boldsymbol{A\mu}, \boldsymbol{A\Sigma A}^{\mathsf{T}})$

Mixture of univariate normal densities: Let $p_1(x|\mu_1, \sigma_1^2)$ and $p_2(x|\mu_2, \sigma_2^2)$ be two Gaussian densities and $\alpha \in (0, 1)$. Consider density:

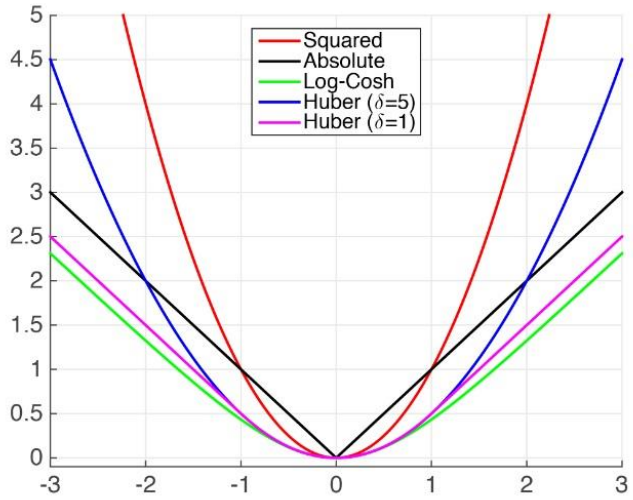
$$q(x) = \alpha p_1(x|\mu_1, \sigma_1^2) + (1-a)p_2(x|\mu_2, \sigma_2^2)$$

$$\mathbb{E}[\mathrm{Q}] = \alpha\mu_1 + (1-\alpha)\mu_2$$

$$\mathbb{V}[\mathrm{Q}] = [\alpha\sigma_1^2 + (1-\alpha)\sigma_2^2] + \left([\alpha\mu_1^2 + (1-\alpha)\mu_2^2] - [\alpha\mu_1 + (1-\alpha)\mu_2]^2\right)$$

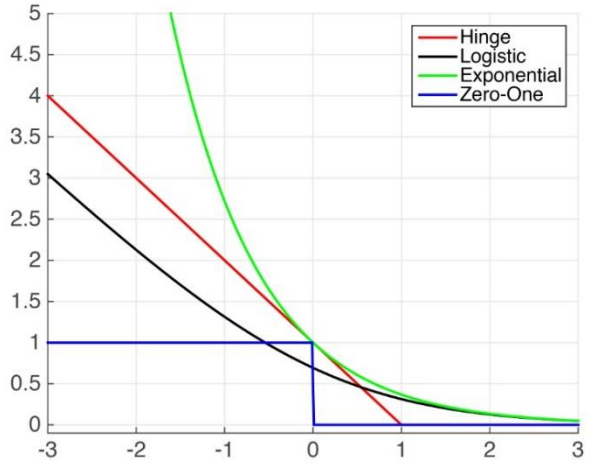
Loss Functions in ML

- Regression Loss Function



Source: https://www.cs.cornell.edu

- Classification Loss Function



Regression Loss Function (MLE)

REGRESSION TASK SETUP:

$f(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ - unknown function we want to model.

$\Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}) : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$ - our ML model (typically NN) to approximate $f(\boldsymbol{x})$.

$\mathcal{D}_{\text{tr}} = \{\boldsymbol{x}_i, y_i\}_{i=1}^m$ - observed data-set, where $\boldsymbol{x}_i \in \mathbb{R}^n$ input variables, and $y_i = f(\boldsymbol{x}_i) + \epsilon_i$ are noisy observations. $\epsilon_i \sim \mathcal{N}(0, \sigma^2_{\text{noise}})$

GOAL: Find parameter $\boldsymbol{\Theta}^* \in \mathbb{R}^d$ such that $\Phi_{\boldsymbol{\Theta}^*}(\boldsymbol{x}) \approx f(\boldsymbol{x})$

We assume that $y(\boldsymbol{x}) = \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}) + \mathcal{N}(0, \sigma^2_{\text{noise}}) \Rightarrow y(\boldsymbol{x}) \sim \mathcal{N}(\Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}), \sigma^2_{\text{noise}})$

Hence, probability density of points sampled independently defines the *likelihood function*:

$$p(y_1, \ldots, y_m | \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}_1), \ldots, \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}_m)) \overset{\text{i.i.d. samples}}{=} \prod_{i=1}^m p(y_i | \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}_i), \sigma^2_{\text{noise}}) = \prod_{i=1}^m \frac{1}{\sigma_{\text{noise}}\sqrt{2\pi}} \exp\left(-\frac{(y_i - \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}_i))^2}{2\sigma^2_{\text{noise}}}\right)$$

$$\boldsymbol{\Theta}^* = \arg\min_{\boldsymbol{\Theta}} \sum_{i=1}^m (y_i - \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}_i))^2 \Rightarrow \arg\max_{\boldsymbol{\Theta}} \left[\prod_{i=1}^m \frac{1}{\sigma_{\text{noise}}\sqrt{2\pi}} \exp\left(\frac{(y_i - \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}_i))^2}{2\sigma^2_{\text{noise}}}\right)\right] = \arg\max_{\boldsymbol{\Theta}} \sum_{i=1}^m \left[\frac{(y_i - \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}_i))^2}{2\sigma^2_{\text{noise}}}\right] =$$

In other words, *maximising log-likelihood* we arrive at *mean squared error (MSE)* loss function.

If we replace $(y_i - \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}_i))^2 \Longrightarrow |y_i - \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}_i)|$ we get *mean absolute error (MAE)* loss: $\boldsymbol{\Theta}^* = \arg\min_{\boldsymbol{\Theta}} \sum_{i=1}^m |y_i - \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}_i)|$

Regression Loss Function (MAP)

REGRESSION TASK SETUP:

$f(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ - unknown function we want to model.

$\Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}) : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$ - our ML model (typically NN) to approximate $f(\boldsymbol{x})$.

$\mathcal{D}_{\text{tr}} = \{\boldsymbol{x}_i, y_i\}_{i=1}^m$ - observed data-set, where $\boldsymbol{x}_i \in \mathbb{R}^n$ input variables, and $y_i = f(\boldsymbol{x}_i) + \epsilon_i$ are noisy observations. $\epsilon_i \sim \mathcal{N}(0, \sigma^2_{\text{noise}})$

Previously, we found parameter value $\boldsymbol{\Theta}^*$ maximising the probability of observing data points $\{y_1, y_2, \ldots, y_m\}$

Now, lets treat $\boldsymbol{\Theta} \in \mathbb{R}^d$ as random variable itself and for a given data $\{y_1, y_2, \ldots, y_m\}$ aim to find most probable value of model parameter $\tilde{\boldsymbol{\Theta}}$:

Let assume prior distribution $\boldsymbol{\Theta} \sim p_{\text{prior}} = \mathcal{N}(\mathbf{0}, \tau^2 \boldsymbol{I})$. Then we need to maximise posterior distribution: $p(\boldsymbol{\Theta}|y_1, \ldots, y_m)$

Using Bayes Formula: $p(\boldsymbol{\Theta}|y_1, \ldots, y_m) = \frac{p(y_1, \ldots y_m|\Phi_{\boldsymbol{\Theta}}(x_1), \ldots, \Phi_{\boldsymbol{\Theta}}(x_m))p_{\text{prior}}(\boldsymbol{\Theta})}{p(y_1, \ldots, y_m)}$ gives:

$$\tilde{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta}} \sum_{i=1}^m \left[-\frac{(y_i - \Phi_{\boldsymbol{\Theta}}(x_i))^2}{2\sigma^2_{\text{noise}}} \right] - \frac{||\boldsymbol{\Theta}||_2^2}{2\tau^2} = \arg\min_{\boldsymbol{\Theta}} \sum_{i=1}^m \frac{(y_i - \Phi_{\boldsymbol{\Theta}}(x_i))^2}{2\sigma^2_{\text{noise}}} + \frac{\sigma^2_{\text{noise}}}{\tau^2} ||\boldsymbol{\Theta}||_2^2$$

Step 2: Taking log does not change position of max

$$= \arg\max_{\boldsymbol{\Theta}} \log [p(y_1, \ldots, y_m|\Phi_{\boldsymbol{\Theta}}(x_1), \ldots, \Phi_{\boldsymbol{\Theta}}(x_m))p_{\text{prior}}(\boldsymbol{\Theta})] =$$

$$= \arg\max_{\boldsymbol{\Theta}} \log [p(y_1, \ldots, y_m|\Phi_{\boldsymbol{\Theta}}(x_1), \ldots, \Phi_{\boldsymbol{\Theta}}(x_m))] + \log [p_{\text{prior}}(\boldsymbol{\Theta})] =$$

In other words, *maximising a posteriory probability* we arrive at *L-2 regularized mean squared error (MSE)* loss function.

If we replace prior from Gaussian to Laplace, i.e $p_{\text{prior}} = \text{Laplace}(0, \beta)$ then we arrive to *L-1 regularised mean squared* loss function:

Step 3:

$$= \arg\max_{\boldsymbol{\Theta}} \left[\tilde{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} \sum_{i=1}^m \frac{(y_i - \Phi_{\boldsymbol{\Theta}}(x_i))^2}{2\sigma^2_{\text{noise}}} + \frac{2\sigma^2_{\text{noise}}}{\beta} ||\boldsymbol{\Theta}||_{1\text{-}2} \right) \right] =$$

Classification Loss Function

CLASSIFICATION TASK SETUP:

$\mathcal{D}_{\text{tr}} = \{\boldsymbol{x}_i, y_i\}_{i=1}^m$ - observed data-set, where $\boldsymbol{x}_i \in \mathbb{R}^n$ input variables, and each $y_i \in \{0, 1\}$.

$\text{Classifier}_{\boldsymbol{\Theta}}(\boldsymbol{x}) = \begin{bmatrix} \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}) \\ 1 - \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}) \end{bmatrix}$ - our ML model classifier (typically NN), outputting two dimensional *stochastic vector*.

where $\Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}) : \mathbb{R}^d \times \mathbb{R}^n \to [0, 1]$ defines probability that $y(\boldsymbol{x}) = 1$

GOAL: Find parameter $\boldsymbol{\Theta}^* \in \mathbb{R}^d$ such that classifier maximizes the proper labelling of input points.

It means, $\Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}) : \mathbb{R}^d \times \mathbb{R}^n \to [0, 1]$ should be close to one when input point $\boldsymbol{x}$ belong to the first class ($y(\boldsymbol{x}) = 1$), and it should be close to zero when input point $\boldsymbol{x}$ belongs to the second class ($y(\boldsymbol{x}) = 0$).

To combine these requirements, consider the following likelihood expression:

$$p(y(\boldsymbol{x})|\boldsymbol{\Theta}) = \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x})^{y(\boldsymbol{x})}(1 - \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}))^{1-y(\boldsymbol{x})}$$

Notice, each output $y(\boldsymbol{x})$, it returns the expression we need to maximization:

$$p(y(\boldsymbol{x})|\boldsymbol{\Theta}) = \begin{cases} \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}) & \text{if } y(\boldsymbol{x}) = 1 \\ 1 - \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}) & \text{if } y(\boldsymbol{x}) = 0 \end{cases}$$

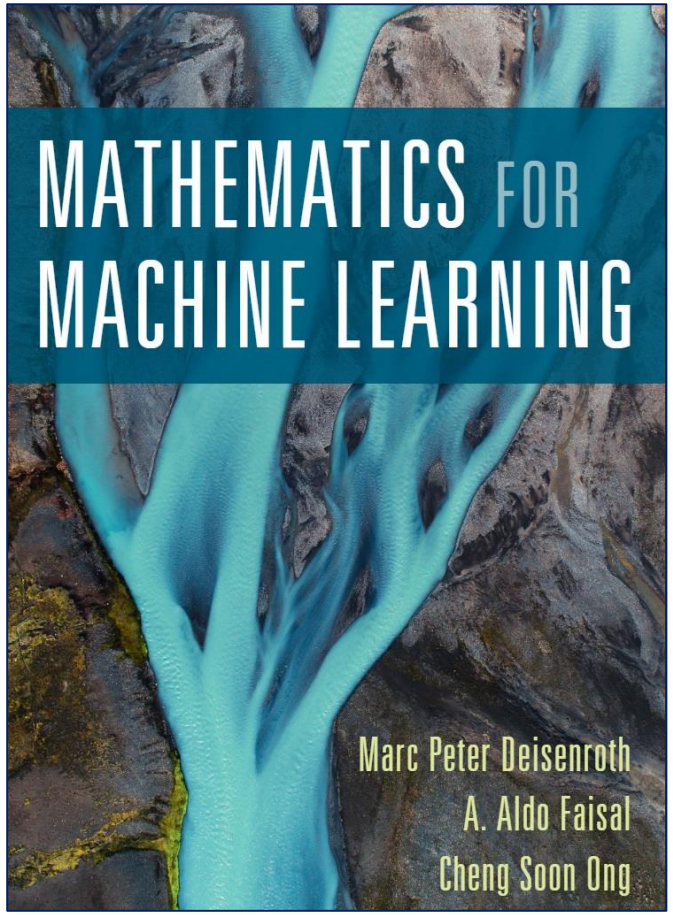
Aiming maximisation of the likelihood:

$$\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} p(y_1, \ldots, y_m|\boldsymbol{\Theta}) \overset{\text{Step 1}}{=} \arg\max_{\boldsymbol{\Theta}} \prod_{i-1}^{m} p(y_i|\boldsymbol{\Theta}) \overset{\text{Step 2}}{=} \arg\max_{\boldsymbol{\Theta}} \prod_{i=1}^{m} \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}_i)^{y_i}(1 - \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}_i))^{1-y_i} =$$

Step 1: Using definition for the likelihood

$$\overset{\text{Step 3}}{=} \arg\max_{\boldsymbol{\Theta}} \log \left[\prod_{i=1}^{m} \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}_i)^{y_i}(1 - \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}_i))^{1-y_i} \right] = \arg\max_{\boldsymbol{\Theta}} \sum_{i=1}^{m} y_i \log [\Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}_i)] + (1 - y_i) \log [1 - \Phi_{\boldsymbol{\Theta}}(\boldsymbol{x}_i)]$$

We arrive at *cross entropy loss function*.


MATHEMATICS FOR
MACHINE LEARNING
Marc Peter Deisenroth
A. Aldo Faisal
Cheng Soon Ong

Thank you