Machine Learning & Deep Learning

Reda Khoufache & Bilal FAYE

Master 2 MIAGE & BUT3 Informatique

Université Paris Descartes & UIT Villetaneuse

2023/2024

Introduction into mathematical modeling

Consider predicting a person's height from his age. Mathematically, this task can be written as follows:

$$height = f(age)$$

where f is a logarithmic function with parameters:

$$f(age) = 2.2\log(80 \times age).$$

In model $f(age) = 2.2\log(80 \times age)$:

- Both input and output (age and height) are single numbers
- In the case where the input and output are vectors, the model would have been expressed with matrices
- **Linear Algebra** is the branch of mathematics that study linear transformations between vectors in a linear fashion
- In Machine Learning or Deep Learning, models are built over 4 algebraic objects:
  1. Scalars
  2. Vectors
  3. Matrices
  4. Tensors

Scalars

Scalars are single numbers defined by the set to which they belong. For example, number of persons in the classroom is $p \in \mathbb{N}$ while the temperature is $t \in \mathbb{R}$. In Python, scalars are defined as:

```python
n = 52
t = 12.5
print("we have " + str(n) + " students")
print("and the temperature is " + str(t) + " degrees.")
```

Vectors

A vector is an ordered array of numbers. Let $\mathbf{x}$ be a vector with d elements

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}.$$

If $x_1, \ldots, x_d \in \mathbb{R}$ then we write $\mathbf{x} \in \mathbb{R}^d$; we say that $\mathbf{x}$ is a d-dimensional vector or a vector of dimension d.

Vectors

Let $\mathbf{x}$ and $\mathbf{y}$ be two vectors defined in $\mathbb{R}^2$ such that

$$\mathbf{x} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \ \mathbf{y} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}.$$

$\mathbf{x}$ and $\mathbf{y}$ are geometrically represented as follows

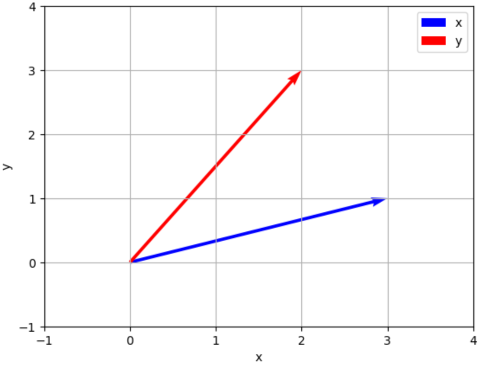

Figure: Geometrical representation of $\mathbf{x}$ and $\mathbf{y}$

Vectors

Vectors are essentially characterized by:

1. Direction
2. Length obtained with the Pythagorean theorem

$$\|\mathbf{x}\| = \sqrt{\sum_{i=1}^{d} x_i^2}.$$

- Vectors are objects characterized by direction and length
- Points have no direction, and their length is equal to zero

We tend to represent vectors as points for readability but remember these points are related to the origin and possess direction and length.

Vectors: Transposition

Definition

Let $\mathbf{x} \in \mathbb{R}^d$ be vector, by default, $\mathbf{x}$ is defined as column vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}.$$

$\mathbf{x}$ transpose is written $\mathbf{x}^T$, and is defined by

$$\mathbf{x}^T = [x_1, \ldots, x_d].$$

Then, the column vector $\mathbf{x}$ can be written as a row vector through the transpose operator:

$$\mathbf{x} = [x_1, \ldots, x_d]^T$$

Column and row versions of the same vector differ and relate through the transpose operator.

Vectors: Scalar multiplication

Definition

Let $\alpha \in \mathbb{R}$ be a scalar and $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{x}$ can be scaled by α as follows

$$\alpha\mathbf{x} = \begin{bmatrix} \alpha x_1 \\ \vdots \\ \alpha x_d \end{bmatrix} \in \mathbb{R}^d.$$

Example: Let $\alpha = 2$, $\beta = -\frac{1}{2}$ be two scalars, and $\mathbf{x}$, $\mathbf{y} \in \mathbb{R}^2$ two 2-dimensional vectors given by

$$\mathbf{x} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}.$$

Hence, the scaled vectors by α and β respectively are

$$\alpha\mathbf{x} = \begin{bmatrix} 6 \\ 2 \end{bmatrix}, \quad \beta\mathbf{y} = \begin{bmatrix} -1 \\ -\frac{3}{2} \end{bmatrix}.$$

Vectors: Unit vector

Definition

Let $\mathbf{x} \in \mathbb{R}^d$. The unit-vector of $\mathbf{x}$ is $\mathbf{u}$ defined as

$$\mathbf{u} = \frac{1}{\|\mathbf{x}\|}\mathbf{x}$$

$\mathbf{u}$ has same direction as $\mathbf{x}$ with $\|\mathbf{u}\| = 1$.

Vectors: Addition

Definition

Let $\mathbf{x}$ and $\mathbf{y} \in \mathbb{R}^d$. Addition between $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$\mathbf{x} + \mathbf{y} = \begin{bmatrix} x_1 + y_1 \\ \vdots \\ x_d + y_d \end{bmatrix} \in \mathbb{R}^d$$

The addition of two vectors and multiplication by scalars must always produce vectors, i.e., if $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ two vectors and $\lambda \in \mathbb{R}$ a scalar, then we have

$$\mathbf{x} + \lambda \mathbf{y} \in \mathbb{R}^d$$

Vectors: Dot product

Definition

Dot product between $\mathbf{x}$ and $\mathbf{y} \in \mathbb{R}^d$ is scalar defined as

$$\mathbf{x} \cdot \mathbf{y} = \|\mathbf{x}\| \times \|\mathbf{y}\| \times \cos(\theta),$$

where θ is the angle between $\mathbf{x}$ and $\mathbf{y}$;
$\cos(\theta) \in [-1, 1]$ measures how much $\mathbf{x}$ and $\mathbf{y}$ are oriented in terms of direction;

A more convenient way to compute dot-product is the algebraic formula.

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{d} x_i y_i.$$

Consider the five vectors below

$$\mathbf{x} = \begin{bmatrix} 2 \\ 1 \end{bmatrix},\ \mathbf{y} = \begin{bmatrix} 4 \\ 2 \end{bmatrix},\ \mathbf{p} = \begin{bmatrix} -1 \\ 3 \end{bmatrix},\ \mathbf{u} = \begin{bmatrix} 3 \\ -1 \end{bmatrix},\ \mathbf{z} = \begin{bmatrix} -2 \\ -1 \end{bmatrix}$$

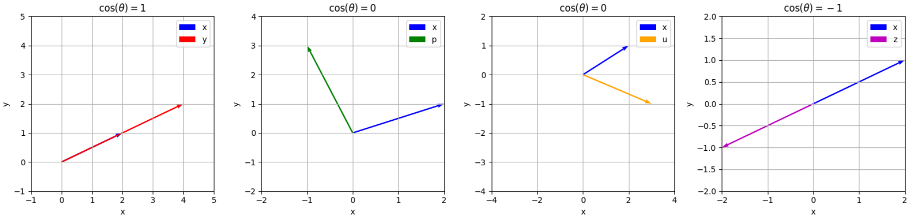

Figure: Illustration of dot product

Vectors: Illustration of dot product

$$\mathbf{x} \cdot \mathbf{y} = \|\mathbf{x}\| \times \|\mathbf{y}\| \times \cos(\theta).$$

$\cos(\theta)$ measures similarity between $\mathbf{x}$ and $\mathbf{y}$ in terms of direction where:

- $\cos(\theta) = 1$ means perfect alignment
- whereas $\cos(\theta) = -1$ means perfect opposition.

Consider the following dot products:

- $\mathbf{x}, \mathbf{y}$ are perfectly aligned resulting into $\cos(\theta) = 1$;
- $\mathbf{x}, \mathbf{p}$ and $\mathbf{x}, \mathbf{u}$ are perpendicular resulting into null dot products because of $\cos(\frac{\pi}{2}) = \cos(\frac{3\pi}{2}) = 0$;
- $\mathbf{x}, \mathbf{z}$ are aligned but look into opposite direction producing $\cos(\theta) = -1$;
- vectors forming angle $\theta \in \left]\frac{3\pi}{2}, \frac{\pi}{2}\right[$ with $\mathbf{x}$ tend to the direction of $\mathbf{x}$;
- inversely, vectors forming angle $\theta \in \left]\frac{\pi}{2}, \frac{3\pi}{2}\right[$ with $\mathbf{x}$ evolve in opposite direction to $\mathbf{x}$.

Matrices

Definition

Visually, matrices tend to be defined as a grid of numbers where elements are identified by row and column indices.

Algebraically, matrices are functions that linearly transform one vector into another. A matrix $\mathbf{W}$ of m rows and d columns (i.e., $\mathbf{W} \in \mathbb{R}^{m \times d}$) is defined as:

$$\mathbf{W} = \begin{bmatrix} w_{11} & \dots & w_{1d} \\ \vdots & \ddots & \vdots \\ w_{m1} & \ddots & w_{md} \end{bmatrix}.$$

$\mathbf{W}$ transforms input vectors $\mathbf{x} \in \mathbb{R}^d$ to output vectors $\mathbf{y} \in \mathbb{R}^m$;

Matrices: Transposition

Definition

Let $\mathbf{A} \in \mathbb{R}^{m \times d}$ be a matrix with m rows and d columns;

$$\mathbf{A} = \begin{bmatrix} a_{11} & \dots & a_{1d} \\ \vdots & \ddots & \vdots \\ a_{m1} & \ddots & a_{md} \end{bmatrix}.$$

Transposed $\mathbf{A}$ noted $\mathbf{A}^T$ is a matrix with d rows and m columns where each j-th column vector in $\mathbf{A}^T$ corresponds to the j-th row vector in $\mathbf{A}$;

$$\mathbf{A}^T = \begin{bmatrix} a_{11} & \dots & a_{m1} \\ \vdots & \ddots & \vdots \\ a_{1d} & \ddots & a_{md} \end{bmatrix}.$$

Matrices: Scalar-Matrix Multiplication

Let $\alpha \in \mathbb{R}$ and $\mathbf{A} \in \mathbb{R}^{m \times d}$, product $\alpha\mathbf{A}$ is defined as

$$\alpha\mathbf{A} = \begin{bmatrix} \alpha a_{11} & \dots & \alpha a_{1d} \\ \vdots & \ddots & \vdots \\ \alpha a_{m1} & \ddots & \alpha a_{md} \end{bmatrix}.$$

$\alpha\mathbf{A} = \mathbf{A}\alpha$

Matrices: Vector-Matrix Multiplication

Let $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{A} \in \mathbb{R}^{m \times d}$, as described above, $\mathbf{A}\mathbf{x}$ is merely the sum

$$\mathbf{A}\mathbf{x} = x_1 \begin{bmatrix} a_{11} \\ \vdots \\ a_{m1} \end{bmatrix} + \cdots + x_d \begin{bmatrix} a_{m1} \\ \vdots \\ a_{md} \end{bmatrix}.$$

The weighted sum above is compactly expressed as sums of products between elements of the vector with elements of matrix rows

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} \sum_{j=1}^{m} a_{1j} x_j \\ \vdots \\ \sum_{j=1}^{m} a_{mj} x_j \end{bmatrix}.$$

Matrices: Matrix-Matrix Multiplication

Product between matrices follows the same principle as matrix-vector product;
Consider 2×2 matrices $\mathbf{A}$ and $\mathbf{B}$;

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}.$$

The product $\mathbf{AB}$ is

$$\mathbf{AB} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix},$$

Definition

Matrix multiplication can be generalized:
Let $\mathbf{A} \in \mathbb{R}^{m \times d}$ and $\mathbf{B} \in \mathbb{R}^{d \times n}$;

Note that the number of columns in $\mathbf{A}$ must be equal to the number of rows in $\mathbf{B}$;
The dimension of the resulting matrix is $m \times n$

$$\mathbf{A} \times \mathbf{B} = \begin{bmatrix} \sum_{j=1}^{d} a_{1j} b_{1j} & \dots & \sum_{j=1}^{d} a_{1j} b_{nj} \\ \vdots & \ddots & \vdots \\ \sum_{j=1}^{d} a_{mj} b_{1j} & \dots & \sum_{j=1}^{d} a_{mj} b_{nj} \end{bmatrix}.$$

Matrices: Few properties on operations between Matrices

$$\mathbf{A}(\mathbf{B}+\mathbf{C}) = \mathbf{AB} + \mathbf{AC}.$$

Multiplication is associative:

$$\mathbf{A}\left(\mathbf{BC}\right) = \left(\mathbf{AB}\right)\mathbf{C}.$$

Multiplication is not commutative:

$$\mathbf{AB} \neq \mathbf{BA}.$$

Transpose of multiplication:

$$\left(\mathbf{AB}\right)^T = \mathbf{B}^T\mathbf{A}^T.$$

Matrices: Special matrices

Matrices are defined by their dimensions but also by the values inside the cells;
There are some types of matrices with useful properties:

- Diagonal
- Identity
- Inverse
- Symmetry
- Orthonormality

Diagonal Matrix:
Diagonal matrices are square with null values except in the diagonal

$$\mathbf{D} = \begin{bmatrix} d_{11} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ddots & d_{dd} \end{bmatrix}.$$

Identity Matrix:
Identity matrices are diagonal matrices with 1s in the diagonal;

Identity matrices preserve input vectors both in direction and in length.

Matrices: Special matrices

Inverse:

Let $\mathbf{A} \in \mathbb{R}^{m \times d}$ be a matrix with m rows and d columns;

We define $\mathbf{A}^{-1} \in \mathbb{R}^{d} \times \mathbb{R}^{m}$ as the inverse of $\mathbf{A}$ that is:

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}.$$

Existence of $\mathbf{A}^{-1}$ must satisfy:

- $\mathbf{A}$ must be square (number of rows = number of columns);
- no linear dependence amongst columns of $\mathbf{A}$.

Matrices: Special matrices

Symmetry:
A symmetric matrix is any matrix that is equal to its own transpose:

$$\mathbf{A} = \mathbf{A}^T.$$

Symmetric matrices often arise when entries are generated by some functions of two arguments that do not depend on the order of arguments, such as Covariance or Distance between data.

Matrices: Special matrices

Orthonormality

An orthogonal matrix is a square whose:

- rows are mutually orthonormal;
  - $A_i \cdot A_j = 0,$ for $i \neq j$, where $\cdot$ represents the dot product.
  - $\|A_i\| = 1$, where $\|\cdot\|$ represents the Euclidean norm (magnitude).
- columns are mutually orthonormal.

$$\mathbf{A}^T\mathbf{A} = \mathbf{A}\mathbf{A}^T = \mathbf{I}.$$

The equation above implies that:

$$\mathbf{A}^{-1} = \mathbf{A}^T.$$

Matrix: Determinant

Definition

Matrix determinant $\mathbf{det}(\mathbf{A})$ is a scalar value:

- Whose sign is positive if the order between the basis vectors is unchanged and negative otherwise.
- Whose value equals 0 if at least one column vector of $\mathbf{A}$ is linearly dependent on the others (which leads to space contraction).

For 2×2 matrices

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix},$$

determinant is given by $\mathbf{det}(A) = ad - cb$.

Matrix: Determinant

Definition

For 3×3 matrices

$$\mathbf{A} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix},$$

the determinant is computed as follows:

$$\mathbf{det}(\mathbf{A}) = a.\mathbf{det}\left(\begin{bmatrix} e & f \\ h & i \end{bmatrix}\right) - b.\mathbf{det}\left(\begin{bmatrix} d & f \\ g & i \end{bmatrix}\right) + c.\mathbf{det}\left(\begin{bmatrix} d & e \\ g & h \end{bmatrix}\right).$$

The same principle applies to higher dimensions.

Matrices: Eigenvalues & Eigenvectors

Definition

When a matrix $\mathbf{A}$ is applied on input vectors, it tends to alter its length and/or direction;
Under some circumstances, certain vectors $\mathbf{v}$ get altered only in their length, which is

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}, \quad \lambda \neq 0, \mathbf{v} \neq \mathbf{0}.$$

These kinds of vectors are known as eigenvectors, and the coefficient of the length alteration is the corresponding eigenvalue.
Note that if $\mathbf{v}$ is an eigenvector of $\mathbf{A}$, then:

- $\forall\ s \neq 0$, $s\mathbf{v}$ is eigenvector of $\mathbf{A}$;
- $s\mathbf{v}$ still has the same eigenvalue as $\mathbf{v}$.

Matrices: Eigenvalues & Eigenvectors

Definition

Eigenvalues are computed through the characteristic equation defined as

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}, \mathbf{A}\mathbf{v} - \lambda\mathbf{v} = 0,$$

resulting into polynomial in λ where the roots are the eigenvalues of $\mathbf{A}$.

$$\mathbf{det}\left(\mathbf{A} - \lambda\mathbf{I}\right) = 0.$$

Definition

Factorization splits linear transformations into atomic steps like distinguishing between rotations and length alteration.

Let $\mathbf{A}$ a diagonalizable matrix (square, invertible, and verify other conditions);
If $\mathbf{A}$ has m linearly independent eigenvectors $\{\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(m)}\}$ with corresponding eigenvalues $\{\lambda_1, \dots, \lambda_m\}$ then $\mathbf{A}$ can be factorized as;

$$\mathbf{A} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1}.$$

When m eigenvectors are concatenated into matrix $\mathbf{V}$ and the associated eigenvalues in the diagonal matrix $\boldsymbol{\Lambda}$, $\mathbf{A}$ can be factorized as

- Where $\mathbf{V}$ is concatenation of m eigenvectors ordered by descending absolute values of eigenvalues.
- If $\mathbf{A}$ is symmetric the matrix of eigenvectors $\mathbf{Q}$ is orthogonal and the factorization becomes
$$\mathbf{A} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^T.$$

Matrix: Matrix factorization - Singular Value Decomposition

Singular Value Decomposition (SVD) provides another way to factorize a matrix;

While eigendecomposition does not apply to any matrix, SVD is;

SVD decomposition has a lot of applications like Image compression, Recommender Systems (Netflix, Spotify, Amazon, etc.);

Matrix: Matrix factorization - Singular Value Decomposition

Definition

Let $\mathbf{M}$ be a $m \times d$ matrix. The SVD decomposition of $\mathbf{M}$ is

$$\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T,$$

where for a given input vector $\mathbf{x}$:

- $\mathbf{V}^T$ is $d \times d$ orthogonal matrix and performs rotation on $\mathbf{x}$ (we call $\mathbf{x}_V$);
- $\boldsymbol{\Sigma}$ is $m \times d$ and performs stretching of $\mathbf{x}_\mathbf{V}$ then projects it into m dimensions (we call $\mathbf{x}_{V\Sigma}$);
- $\mathbf{U}$ is $m \times m$ orthogonal matrix and performs last rotation on $\mathbf{x}_{V\Sigma}$.

Matrix: Matrix factorization - Singular Value Decomposition

Definition

$\mathbf{V}$ is constructed by the eigenvectors of $\mathbf{M}^T\mathbf{M}$;

$\mathbf{U}$ is constructed by the eigenvectors of $\mathbf{M}\mathbf{M}^T$;

$\mathbf{\Sigma}$ is a diagonal matrix whose entries are called singular values and correspond to the square root of the eigenvalues of $\mathbf{M}\mathbf{M}^T$ or $\mathbf{M}^T\mathbf{M}$;

The number of non-zero singular values equals the rank of $\mathbf{M}$;

If $\mathbf{\Sigma}$ is not square, the rest of the diagonal is filled with zeros.

Matrix: python illustration

```python
import numpy as np
#Creation of a 2 X 2 matrix
A = np.array([[2,1],[1,2]], dtype = np.float32)
print('Matrix A has the dimension ' + str(A.shape))
print(A)
#Product between two matrices
B = np.array([[3,4,1], [4,7,0]])
AB = A.dot(B)
print('Dimension of A.B = ' + str(AB.shape))
print(AB)
#Transpose
ABt = AB.transpose()
print('Dimension of transposed AB = ' + str(ABt.shape))
print('\nTransposed AB = ')
print(ABt)
#Diagonal Matrix
D = np.diag([1,2,3,4,5,6])
print(D)
```

Matrix: python illustration

```python
import numpy as np

#Determinant
A = np.array([[1, 2], [3, 4]])
detA = np.linalg.det(A)
print('Determinant of A = ' + str(detA))

#eigenvalues & eigenvectors
A = np.array([[2,1],[1,2]], dtype = np.float32)
lambdas, V = np.linalg.eig(A)
print('eigenvalues = ')
print(lambdas)
print('eigenvectors = ')
print(V)

#SVD decomposition
U, Sigma, Vt = np.linalg.svd(A)
```