ET-287 – Signal Processing using Neural Networks

1. Introduction

Professor Sarah Negreiros de Carvalho Leite
Aeronautics Institute of Technology
Electronic Engineering Division
Department of Telecommunications

sarahnc@ita.br, sarah.leite@gp.ita.br
room: 221

Presentation

Introduce yourself by addressing the following points:

- Academic background
- Current course and areas of specialization
- Interest in this discipline

Course Plan

Unit 1 - Introduction

In this unit, we will cover:

- Concepts definition.
- A brief review of linear algebra using Python.

Understanding the concepts

- Artificial Intelligence (AI)
- Machine Learning (ML)
- Artificial Neural Networks (ANN)
- Deep Learning (DL)

These encompass concepts and technologies that are increasingly present in our daily lives.

It is a challenge to find a scientific field that has not yet benefited from these technological solutions.

Timeline

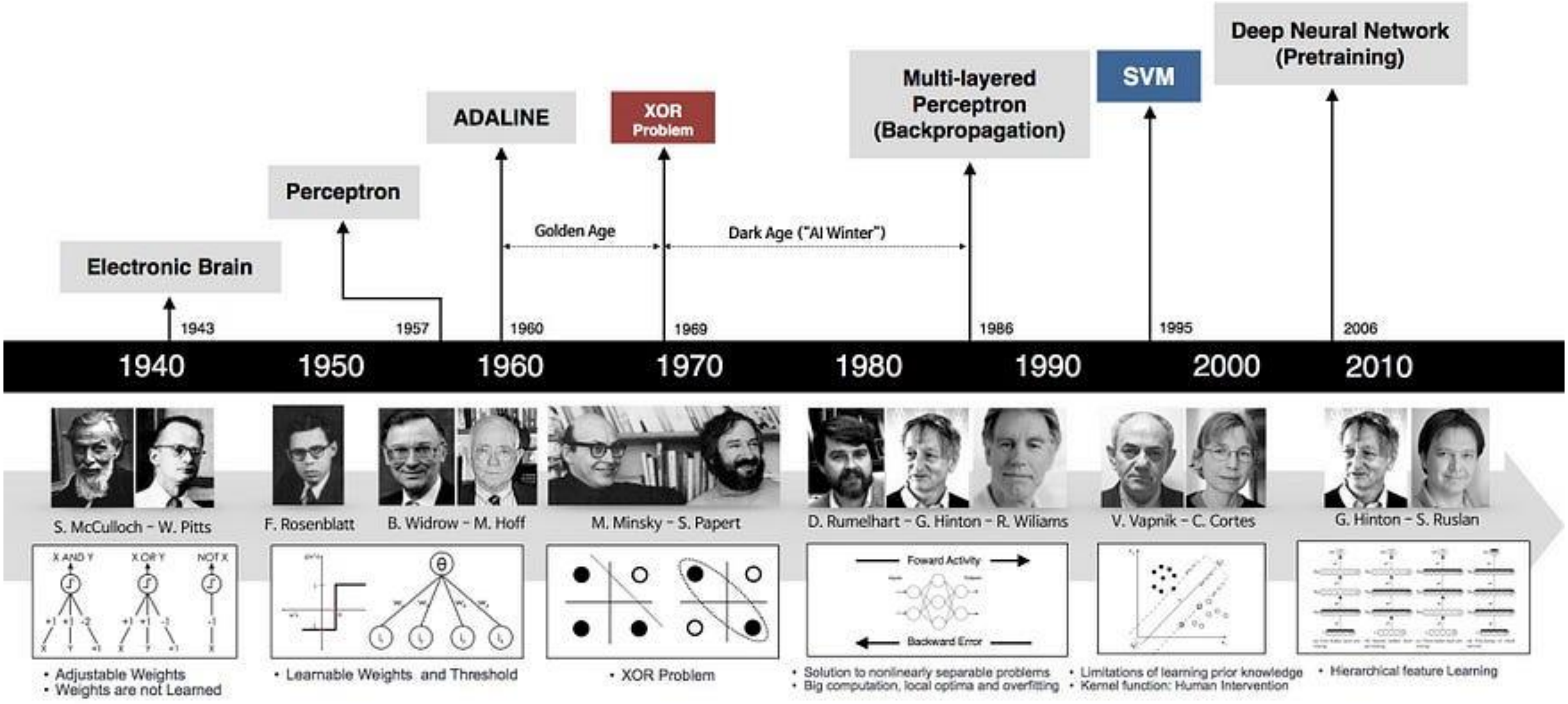
Artificial Intelligence: aims to create machines and programs that make us consider them intelligent.

Machine Learning: the ability of machines to learn from provided data without being explicitly programmed.

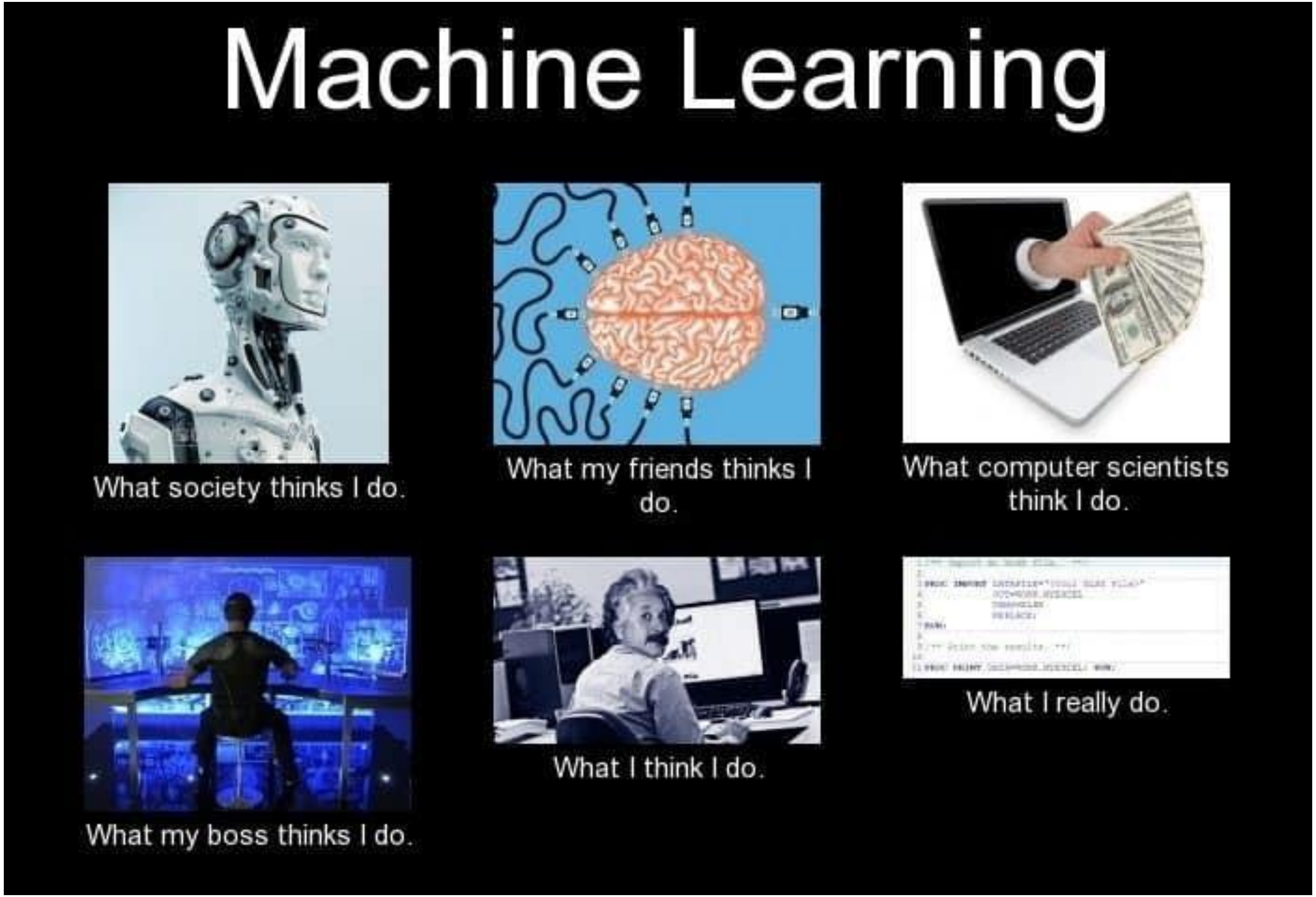
Deep Learning: learning based on deep neural networks.

1950 1980 2006

A timeline on neural networks history



Source: https://sefiks.com/2017/10/14/evolution-of-neural-networks/


Machine Learning
What society thinks I do.
What my friends thinks I do.
What computer scientists think I do.
What my boss thinks I do.
What I think I do.
What I really do.

Introduction

Every time we use a smartphone, we interact with technologies that emerge from these concepts.

Example:

- Tracking and route suggestions,
- Profile and content recommendations on social networks,
- Product suggestions and online shopping sites,
- Among many other functionalities.

Introduction

Devices collect information about our use and our behavior and the way we use programs and applications.

The collected data is processed and provides information that allows algorithms to 'learn' and make predictions and inferences about our 'pattern' of use.

It is a powerful technology that allows you to solve different types of problems.

It is important to be ‘literate’ in this world and understand the basic concepts and how it works.

Example

Can you read the text?

De aorcdo com uma pqsieusa de uma uinrvesriddae ignlsea, não ipomtra em qaul odrem as lrteas de uma plravaa etãso, a úncia csioa iprotmatne é que a piremria e útmlia lrteas etejasm no lgaur crteo. O rseto pdoe ser uma ttaol bçguana que vcoê pdoe andia ler sem pborlmea. Itso oorcre pqorue nós não lmeos cdaa lrtea isladoa, mas a plravaa cmoo um tdoo.

Can a machine read?

Example

Using Google Translator (Pt -> En -> Pt):

De acordo com um pqsieusa de um uinrvesriddae ígneo, não ipomtra no qual os leites de um plravaa etérico odrem, a única razão iprotmatne é que a piremyria e utmlia lrteas eteejasm no crteo lgaur. O texto pode ser um texto confuso que você pode ler sem desfocar. É porque não a vemos como uma ilha isolada, mas falamos dela como um todo.

Example

Using ChatGPT :

O texto que você forneceu está escrito da seguinte forma:

"De acordo com uma pesquisa de uma universidade inglesa, não importa em qual ordem as letras de uma palavra estão, a única coisa importante é que a primeira e última letras estejam no lugar certo. O resto pode ser uma total bagunça que você pode ainda ler sem problema. Isso ocorre porque nós não lemos cada letra isolada, mas a palavra como um todo."

Example

What's missing for Google translator to be able to better translate the text?

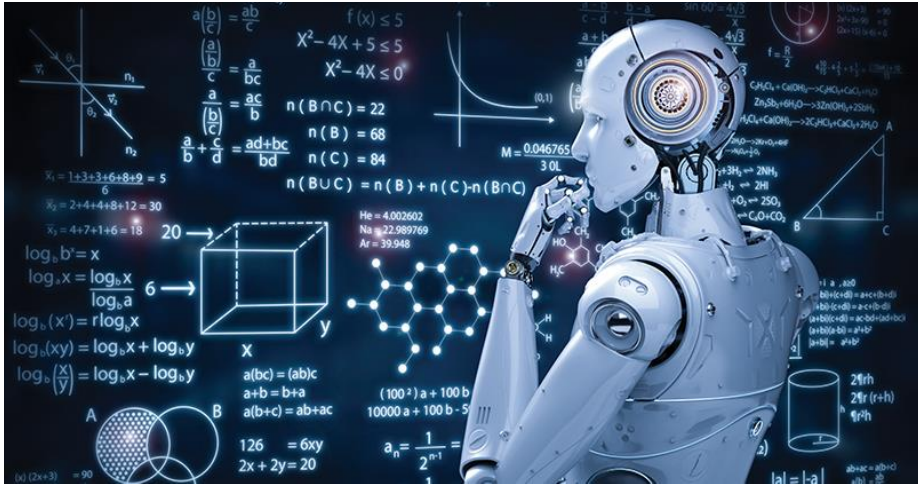
De acordo com um pqsieusa de um uinrvesriddae ígneo, não ipomtra no qual os leites de um plravaa etérico odrem, a única razão iprotmatne é que a piremyria e utmlia lrteas eteejasm no crteo lgaur. O texto pode ser um texto confuso que você pode ler sem desfocar. É porque não a vemos como uma ilha isolada, mas falamos dela como um todo.

De acordo com uma pesquisa de uma universidade inglesa, não importa em qual ordem as letras de uma palavra estão, a única coisa importante é que a primeira e última letras estejam no lugar certo. O resto pode ser uma total bagunça que você pode ainda ler sem problema. Isso ocorre porque nós não lemos cada letra isolada, mas a palavra como um todo.

How far can learning machines go?

The objective of AI is to enable machines to perform tasks that supposedly require intelligence when done by humans:

- Be able to discuss and explain your knowledge to other people
- Extrapolate and contextualize concepts
- Being able to prepare criticisms and detect limitations
- Propose improvements
- Understand concepts
- Use the knowledge you have



Example: Night guard

A night guard (NG) was doing his usual rounds when he noticed a man (M) kneeling next to a streetlight, searching for something.

NG: “*Did you lose something, sir?*”

M: “*Yes, I lost a bunch of keys.*”

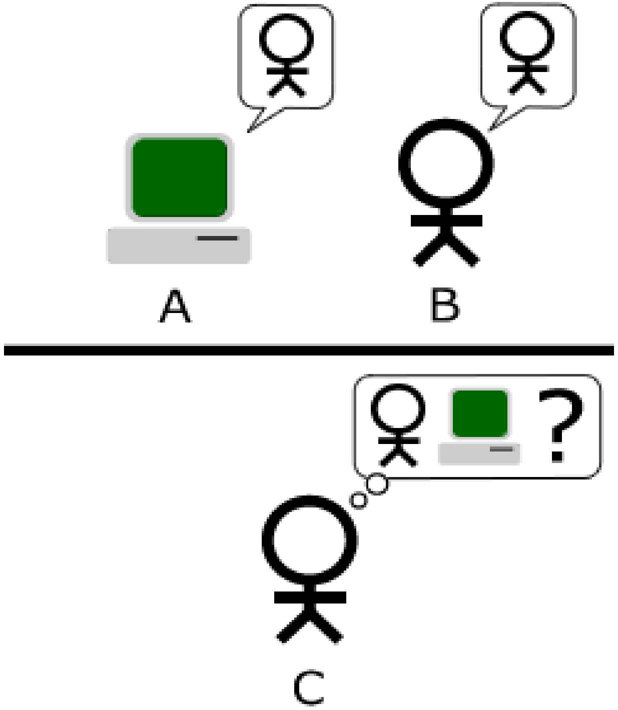
NG: “*Do you have any idea where you lost them?*”

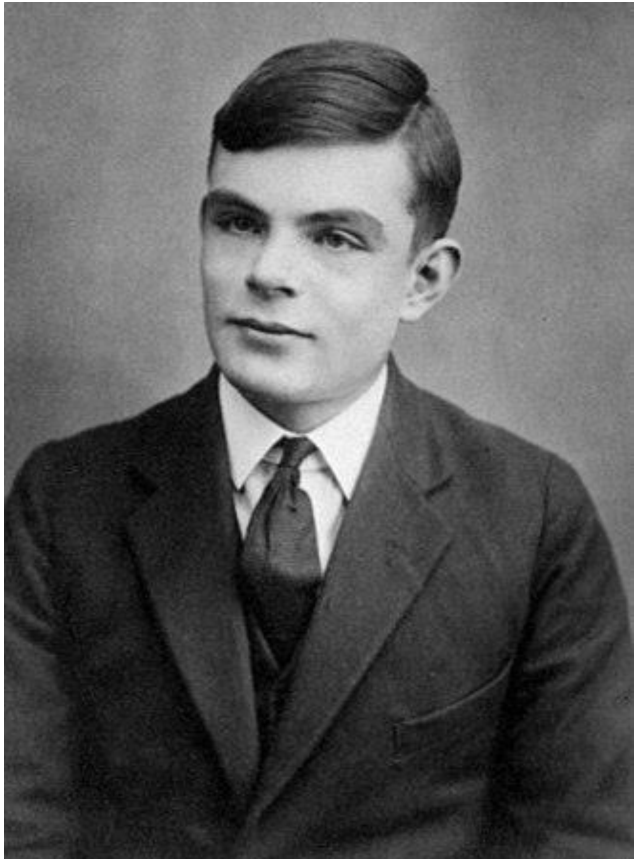
M: “*Yes, it was down at the end of the street, in the dark.*”

NG: “*But then why are you here, searching under this streetlight?*”

M: “*Because this is the only place where there is enough light for me to find it.*”

Turing Test (1950)

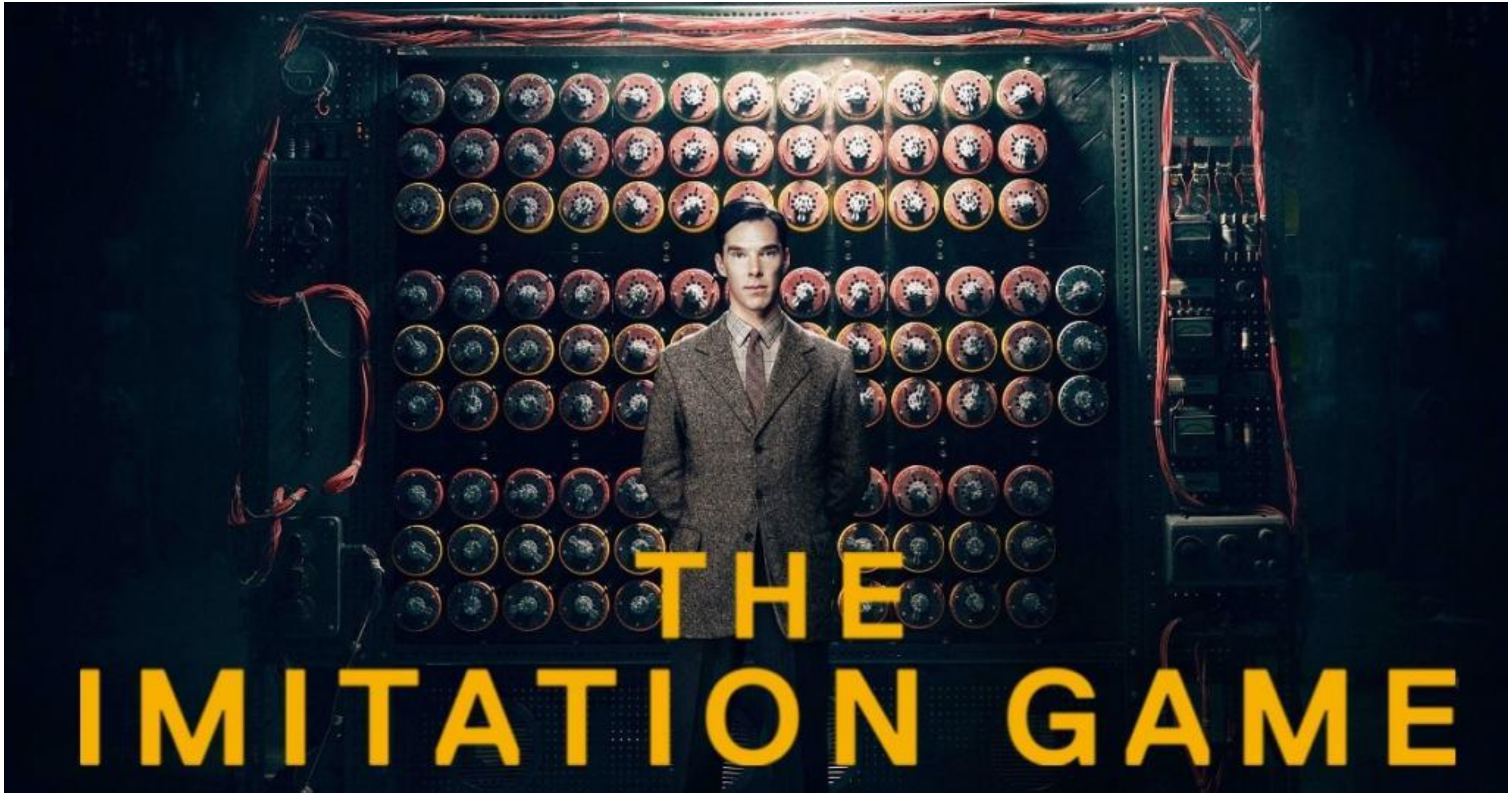




Alan Turing (1912 – 1954)

Sources: https://commons.wikimedia.org/wiki/File:Turing_Test_version_3.png#/media/File:Turing_Test_version_3.png, https://commons.wikimedia.org/wiki/File:Alan_Turing_Aged_16.jpg#/media/File:Alan_Turing_Aged_16.jpg

Movie recommendation



Chinese Room

John Searle (1932)

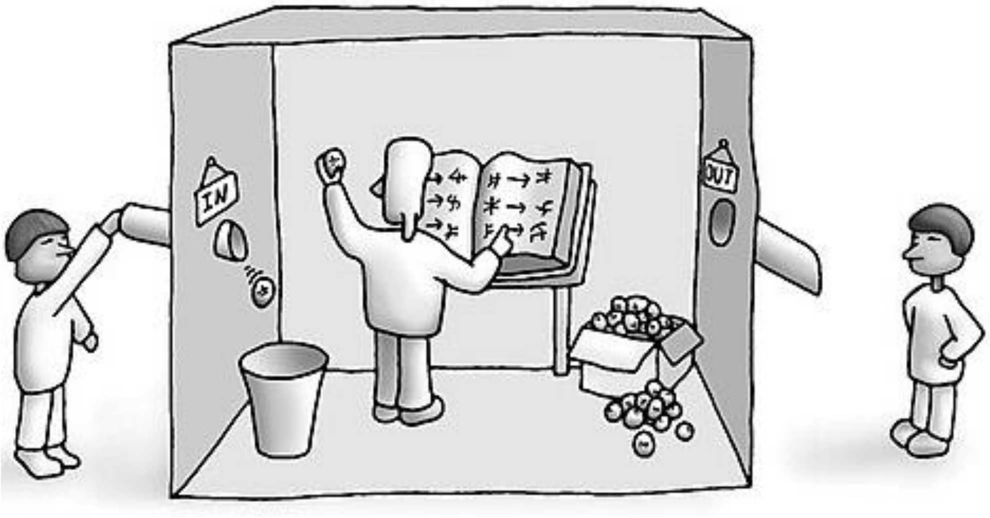
Consider a room with a person unfamiliar with the Chinese language. In the room, there are many baskets containing Chinese ideograms and a rule book explaining how to combine them.

Source: https://commons.wikimedia.org/wiki/File:John_searle2.jpg#/media/Ficheiro:John_searle2.jpg

Searle’s Room

The person then receives questions in Chinese and, consulting the rule book, combines the Chinese ideograms from the baskets to create a new sequence that serves as the answer.

This new sequence is then sent out of the room by the person. Although this individual doesn't know Chinese, they are responding to the questions in the Chinese language.



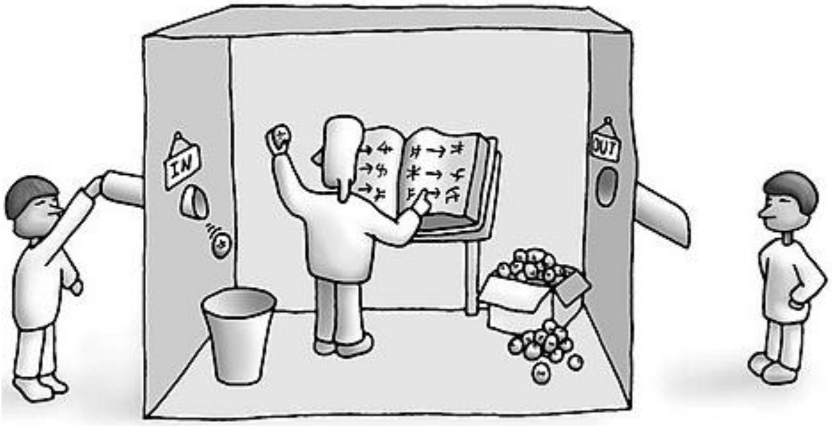
Source: https://commons.wikimedia.org/wiki/File:2-chinese-room.jpg#/media/File:2-chinese-room.jpg

Searle’s Room

Is there any difference between them and a person who is proficient in the Chinese language and answers the same questions without using the rule book?

Yes!! The first one is merely following syntactic rules without fully understanding what they are doing.

The second one is associating semantics (meaning) with what they are doing.



Source: https://commons.wikimedia.org/wiki/File:2-chinese-room.jpg#/media/File:2-chinese-room.jpg

Searle’s Room

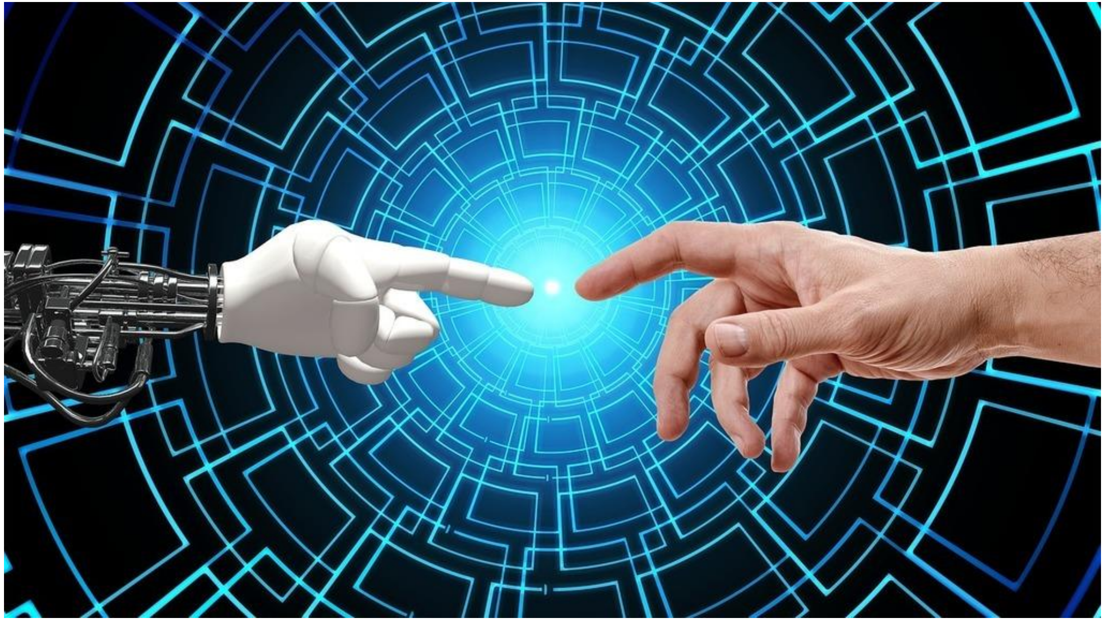
With this reasoning, Searle concluded that computers, being syntactic machines, can replace a person who is unfamiliar with the Chinese language and follows rules.

However, they cannot replace the role of someone who knows the Chinese language and can add semantics to the process.

Computers can execute formal or syntactic processes, but they cannot 'think' because thinking involves semantics.

Searle asserts that computers can never be 'intelligent' because programs alone are not sufficient to attribute minds to computers.

What do you think about the points raised by Turing and Searle in the face of today's AI advances?

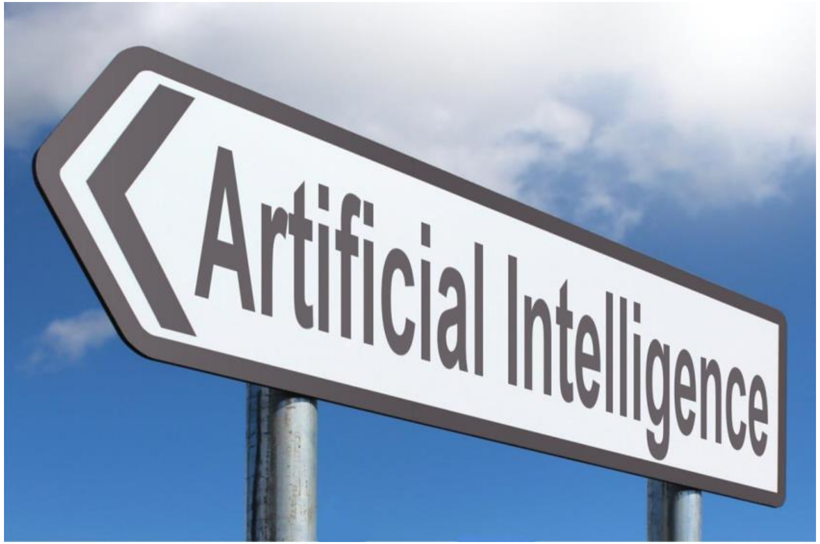


Machine Learning

A machine, particularly one based on digital computation, may not be considered intelligent, but it is possible to make its behavior indistinguishable from intelligent behavior.

Machines with AI should be able to:

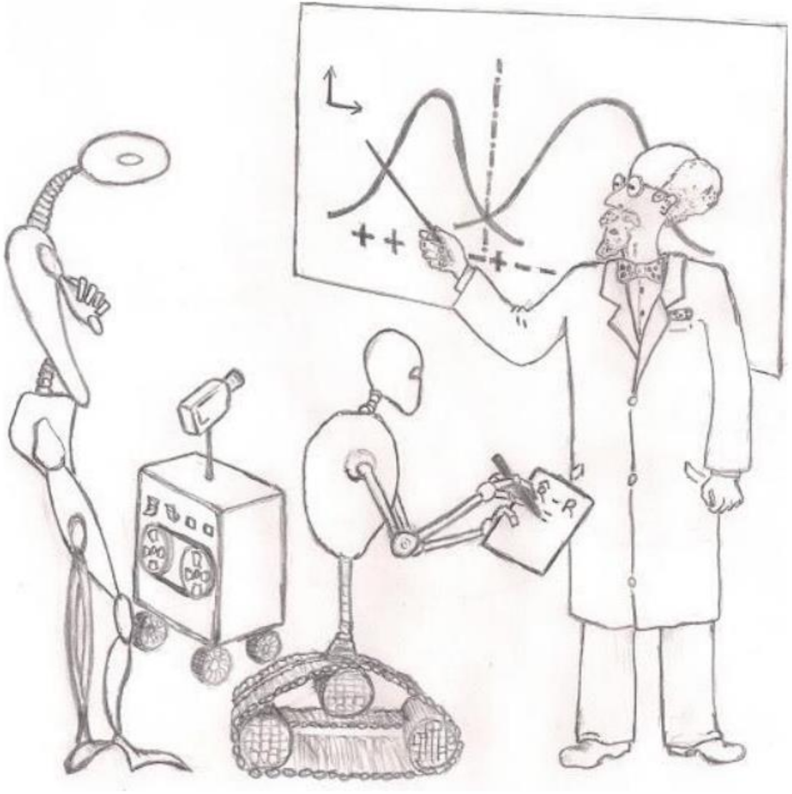
- Store knowledge,
- Apply it to solve problems,
- Acquire new knowledge through experience.



Human vs. Machine Learning

Do machines and humans learn in the same way?

What are the differences?



Human vs. Machine Learning

Humans learn through a combination of experience, instruction, observation, imitation, reflexion, and interaction with other people and with the environment.

Machine Learning

Machines primarily learn based on mathematical algorithms and data, without the ability for human-like reflection or experience.

Machines extracting knowledge directly from available data, aiming to:

- Categorize them,
- Make predictions,
- Assist in decision-making,
- Detect faults,
- Perform diagnostics,
- Recognize patterns,
- Retrieve information,
- Etc.

Human vs. Machine Learning

Humans:

knowledge ⇔ experience/observation/repetition/interaction/reflexion/…

Machines:

knowledge ⇔ data

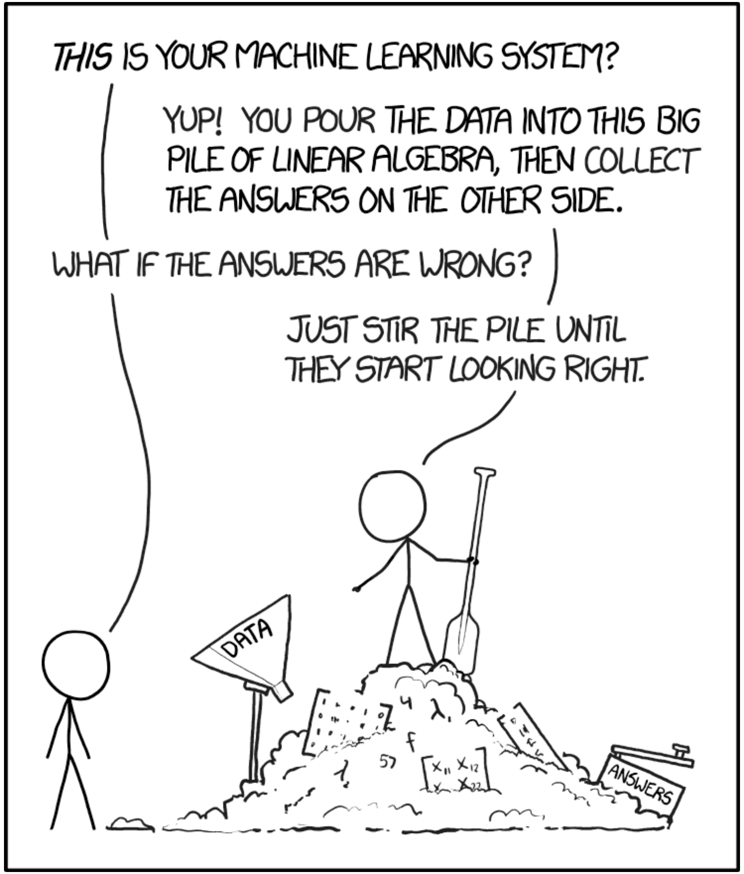
A very brief review of linear algebra with Python

Before we begin the study of neural networks and machine learning algorithms, we will provide a brief review of Python and of important concepts in Linear Algebra.

Typically, machine learning algorithms deal with a large amount of data organized in specific forms or tables. To handle these algorithms, this information is often represented in the form of matrices, with the types of values for a particular object placed in the columns and individual entries in the rows.

A very brief review of linear algebra with Python

Machine learning algorithms deal with a large amount of data organized in specific forms or tables. To handle these algorithms, this information is often represented in the form of matrices, with the types of values for a particular object placed in the columns and individual entries in the rows.



Scalar

Scalar is a single numerical value belonging to the set of real numbers $\mathbb{R}$, for example, the number 5.

$$x = 5.$$

The magnitude of a real scalar x is given in the form: $|x| = \begin{cases} x & if\ x \geq 0 \\ -x & if\ x < 0 \end{cases}$

Vector

Vector is a one-dimensional array of data. The vector can be a row or column. A vector with n real numbers belongs to the domain $\mathbb{R}^n$.

$$\boldsymbol{v} = [1\ 2\ 3] \text{ - row vector} \qquad \boldsymbol{w} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \text{ - column vector}$$

with $\boldsymbol{v} = \boldsymbol{w}^T$.

$\mathbf{0}_n$ is the zero vector of dimension n, with all elements equal to zero.
$\mathbf{1}_n$ is the vector of dimension n, with all elements equal to 1.

Matrix

Matrix is an array of data with m rows and n columns.

The matrix is said to have dimensions mxn. If the matrix is composed of real numbers, then it belongs to the domain $\mathbb{R}^{mxn}$ or $\boldsymbol{X} \in \mathbb{R}^{mxn}$.

$$\boldsymbol{X} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

For matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ of appropriate dimensions such that the product $\boldsymbol{XY}$ exists, we have the following relationship: $(\boldsymbol{XY})^T = \boldsymbol{Y}^T \boldsymbol{X}^T$.

Special matrices

Identity Matrix $(\boldsymbol{I} \ \boldsymbol{or} \ \boldsymbol{I_n})$: square matrix with ones on the main diagonal and zeros elsewhere.

$$\boldsymbol{I} = \boldsymbol{I_3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Diagonal Matrix: all off-diagonal elements are zero.

$$\boldsymbol{D} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix}$$

Triangular Matrices: can be upper or lower triangular, with zeros below or above the main diagonal, respectively.

$$\boldsymbol{U} = \begin{bmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & f \end{bmatrix} \qquad \boldsymbol{L} = \begin{bmatrix} a & 0 & 0 \\ b & c & 0 \\ d & e & f \end{bmatrix}$$

Sparse Matrix: with a significant number of zero elements.

$$\boldsymbol{S} = \begin{bmatrix} a & 0 & 0 \\ 0 & 0 & c \\ b & 0 & 0 \end{bmatrix}$$

Tensors

Tensors are arrays formed with 3 or more dimensions. The number of dimensions depends on the object being modeled.

For example: In addition to data in the rows and columns of a matrix, you may need this data at different points in time. To store this sequence of 2-D matrices, a 3-D structure is required. A 3-D tensor has m rows, n columns, and depth k, with dimensions $mxnxk$.

$$t = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \ddots \begin{bmatrix} 1 & 2 & 4 \\ 4 & 0 & 2 \\ 3 & 8 & 1 \end{bmatrix}$$

Outer product

The outer product between two vectors $x \in \mathbb{R}^n$ and y $\in \mathbb{R}^m$ is a matrix of dimensions nxm with a rank of one.

Given $\boldsymbol{x} = [x_1 \quad x_2 \quad \cdots \quad x_n]^T$ and $\boldsymbol{y} = [y_1 \quad y_2 \quad \cdots \quad y_m]^T$ the outer product is:

$$\boldsymbol{x}\boldsymbol{y}^T = \begin{bmatrix} x_1y_1 & x_1y_2 & \cdots & x_1y_m \\ x_2y_1 & x_2y_2 & \cdots & x_2y_m \\ \vdots & \vdots & \ddots & \vdots \\ x_ny_1 & x_ny_2 & \cdots & x_ny_m \end{bmatrix}$$

Observe that vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ can have different dimensions.

Inner product

The inner product between two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, with $\boldsymbol{x} = [x_1 \quad x_2 \quad \dots \quad x_n]^T$ and $\boldsymbol{y} = [y_1 \quad y_2 \quad \dots \quad y_n]^T$ is:

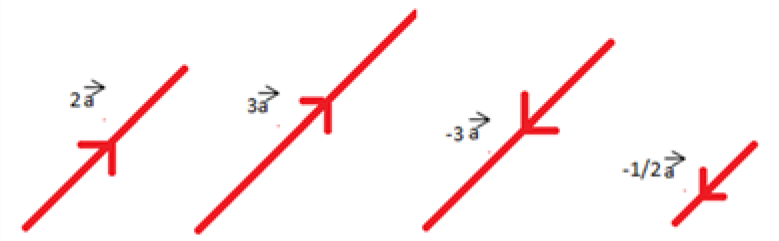
$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^{n} x_i y_i = \boldsymbol{x}^T \boldsymbol{y}$$

Properties:

1. $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{y}, \boldsymbol{x} \rangle$
2. $\langle \boldsymbol{x} + \boldsymbol{y}, \boldsymbol{z} \rangle = \langle \boldsymbol{x}, \boldsymbol{z} \rangle + \langle \boldsymbol{y}, \boldsymbol{z} \rangle$
3. $\langle \alpha . \boldsymbol{x}, \boldsymbol{y} \rangle = \alpha . \langle \boldsymbol{x}, \boldsymbol{y} \rangle$
4. $\langle \boldsymbol{x}, \boldsymbol{x} \rangle \geq 0, \forall \boldsymbol{x} \in \mathbb{R}^n$
5. $\langle \boldsymbol{x}, \boldsymbol{x} \rangle = 0 \Leftrightarrow \boldsymbol{x} = \mathbf{0}$

Matrix x Vector

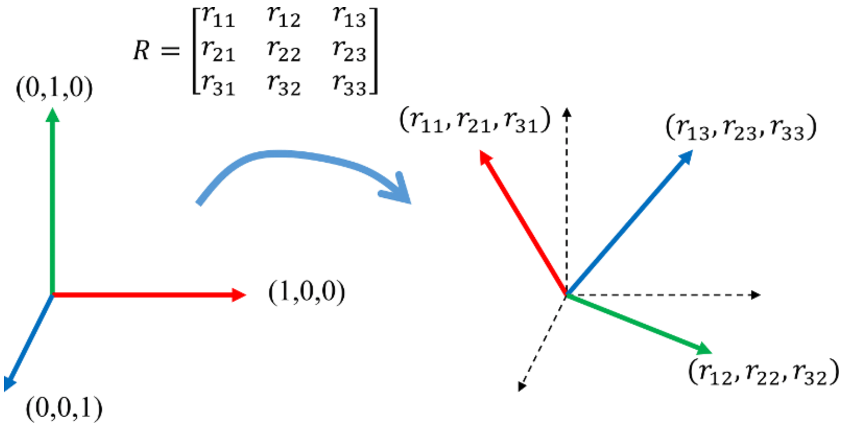
When you multiply a vector by a non-zero scalar, the direction of the vector remains unchanged, only its magnitude is subject to change.



Matrix x vector

On the other hand, when you multiply a vector by a matrix, and here we restrict ourselves to **square** matrices, it is possible to induce rotation and magnitude variation in the vector.

E.g. 3D rotation: $\boldsymbol{v' = Rv}$



If the vector is non-zero, there always exists a square matrix that maps any vector to any other vector in the same vector space.

Figure source: http://motion.pratt.duke.edu/RoboticSystems/CoordinateTransformations.html

Eigenvectors and eigenvalues

Now, given a fixed matrix and varying the vector, in the product between a square matrix and a vector, there are vectors that a given matrix cannot rotate, promoting only the scaling of the vector.

These vectors are the **eigenvectors** of the matrix and the scaling is given by the corresponding **eigenvalue**.

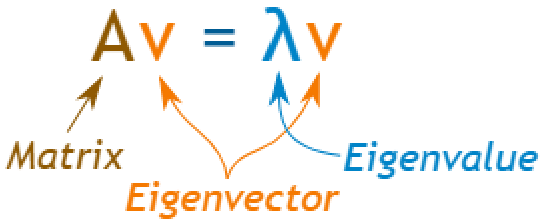


Figure source: http://motion.pratt.duke.edu/RoboticSystems/CoordinateTransformations.html

Eigenvectors and eigenvalues

Let $\boldsymbol{A} \in \mathbb{R}^{nxn}$. A scalar $\lambda \in \mathbb{C}$ is called an **eigenvalue** of $\boldsymbol{A}$ if there exists a non-zero vector $\boldsymbol{x} \in \mathbb{C}^n$, called the associated **eigenvector** to λ, such that:

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

or

$$(\lambda \boldsymbol{I} - \boldsymbol{A})\boldsymbol{x} = \boldsymbol{0}$$

There exists $\boldsymbol{x} \in \mathbb{C}^n$, $\boldsymbol{x} \neq \boldsymbol{0}$, if and only if $\det(\lambda \boldsymbol{I} - \boldsymbol{A}) = 0$.

$\Delta(\lambda) \triangleq \det(\lambda \boldsymbol{I} - \boldsymbol{A})$ is the **characteristic polynomial** of $\boldsymbol{A}$.

Since the degree of $\Delta(\lambda)$ is n, the matrix $\boldsymbol{A}$ has n eigenvalues, not necessarily distinct.

Matrix x vector

For a **rectangular** matrix $\boldsymbol{A} \in \mathbb{R}^{mxn}$, the result of the product belongs to the space spanned by the columns of the matrix, because it is given by a linear combination of the columns of the matrix and the product can be interpreted as a vector formed by the inner products between each row of the matrix and the vector.

$$\boldsymbol{Ax} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} + \cdots + x_n \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a_1}^T \boldsymbol{x} \\ \boldsymbol{a_2}^T \boldsymbol{x} \\ \vdots \\ \boldsymbol{a_m}^T \boldsymbol{x} \end{bmatrix}$$

with $\boldsymbol{a_i} = [a_{i1} \quad a_{i2} \quad \cdots \quad a_{in}]^{\boldsymbol{T}}$.

Operations between matrices

Addition can only occur between matrices of the same dimension.
Addition is distributive:

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

and

$$(\boldsymbol{A}+\boldsymbol{B})\boldsymbol{C}=\boldsymbol{AC}+\boldsymbol{BC}$$

Matrix **product**: to exist, it is necessary that the number of columns in the left matrix coincides with the number of rows in the right matrix of the product.

Matrix multiplication is associative: $(\boldsymbol{AB})\boldsymbol{C}=\boldsymbol{A}(\boldsymbol{BC})$

Matrix multiplication is generally not commutative, even if both products exist, i.e., $\boldsymbol{AB}\neq\boldsymbol{BA}$, although $\boldsymbol{AB}=\boldsymbol{BA}$ can occur for specific pairs of matrices.

Matrices

Consider a square matrix $\boldsymbol{A} \in \mathbb{R}^{nxn}$.

Idempotent: It is said to be idempotent if $\boldsymbol{A}^r = \underbrace{\boldsymbol{A}.\boldsymbol{A}.\ldots.\boldsymbol{A}}_{r} = \boldsymbol{A}$, for $r \geq 1$. If $\boldsymbol{A}$ is idempotent, then $\boldsymbol{I} - \boldsymbol{A}$ is also.

Cofactor: the cofactor of the element a_{ij} with $(i, j = 1, \ldots, n)$ is given by:

$$c_{ij} = (-1)^{i+j} m_{ij}$$

where m_{ij} is the determinant of the matrix formed by eliminating the *i*-th row and *j*-th column from matrix $\boldsymbol{A}$.

Trace: the trace of $\boldsymbol{A}$, represented by $tr(\boldsymbol{A})$, is the sum of the elements on the diagonal of $\boldsymbol{A}$, i.e.:

$$tr(\boldsymbol{A}) = \sum_{i=1}^{n} a_{ij}$$

The trace only applies to square matrices, and can be given by the sum of its eigenvalues.
For square matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of the same dimension, the property holds: $tr(\boldsymbol{AB}) = tr(\boldsymbol{BA})$.

Matrices

Determinant: $|\boldsymbol{A}| = det(\boldsymbol{A}) = \begin{cases} \sum_{i=1}^{n} a_{ij}c_{ij} & \text{for any } j \\ \sum_{j=1}^{n} a_{ij}c_{ij} & \text{for any } i \end{cases}$

The determinant only applies to square matrices and can be given by the product of its eigenvalues.

For square matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of the same dimension, the property holds:

$$\det(\boldsymbol{AB}) = \det(\boldsymbol{A}) \,.\, \det(\boldsymbol{B})$$

If $\det(\boldsymbol{A}) = 0$ The matrix $\boldsymbol{A}$ is said to be **singular**.

A matrix $\boldsymbol{A}$ has an **inverse** if and only if $\det(\boldsymbol{A}) \neq 0$, that is, when $\boldsymbol{A}$ is non-singular.

Matrices

The **inverse** of a matrix $\boldsymbol{A} \in \mathbb{R}^{nxn}$ is a matrix $\boldsymbol{A}^{-1} \in \mathbb{R}^{nxn}$ such that:

$$\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}$$

For a matrix to be invertible, it must be square and non-singular.

Properties:

- $(\boldsymbol{A}^{-1})^T = (\boldsymbol{A}^T)^{-1}$
- $(\boldsymbol{A}\boldsymbol{B})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}$, considering that $\boldsymbol{A}$ and $\boldsymbol{B}$ have inverses.

Matrices

The **pseudo-inverse** of a matrix $\boldsymbol{A} \in \mathbb{R}^{nxn}$ is a matrix $\boldsymbol{A}^{+} \in \mathbb{R}^{nxn}$ such that:

- $\boldsymbol{A}\boldsymbol{A}^{+}\boldsymbol{A} = \boldsymbol{A}$
- $\boldsymbol{A}^{+}\boldsymbol{A}\boldsymbol{A}^{+} = \boldsymbol{A}^{+}$
- $\boldsymbol{A}\boldsymbol{A}^{+} = \boldsymbol{A}^{+}\boldsymbol{A}$ are symmetric matrices.

It can be demonstrated that there exists a unique pseudo-inverse for each matrix.

Properties:

- $\boldsymbol{A}^{+}$=$\boldsymbol{A}^{-1}$, if $\boldsymbol{A}$ is a square matrix and non singular
- $(\boldsymbol{A}^{+})^{+} = \boldsymbol{A}$
- $(\boldsymbol{A}^{+})^{T} = (\boldsymbol{A}^{T})^{+}$
- $(\boldsymbol{A}\boldsymbol{A}^{+})^{T} = \boldsymbol{A}\boldsymbol{A}^{+}$
- $(\alpha\boldsymbol{A})^{+}= \alpha^{-1}\boldsymbol{A}^{+}$, if $\alpha \neq 0$
- $\boldsymbol{A}\boldsymbol{A}^{T}(\boldsymbol{A}^{+})^{T} = \boldsymbol{A}$ and $\boldsymbol{A}^{T}\boldsymbol{A}\boldsymbol{A}^{+} = \boldsymbol{A}^{T}$
- $\boldsymbol{A}^{+} = (\boldsymbol{A}^{T}\boldsymbol{A})^{+}\boldsymbol{A}^{T} = \boldsymbol{A}^{T}\left(\boldsymbol{A}\boldsymbol{A}^{\boldsymbol{T}}\right)^{+}$

Tables and Matrices

Databases typically describe the characteristics of a particular problem in table format.
For instance, to describe each student in a class, we can create a table containing data such as name, age, previous education, and grades for each assignment. By understanding this data, we can propose analyses that generate new types of information, allowing us to make generalizations and identify patterns among students.

Name	Age	Education	Grade 1	Grade 2
Ane	33	Engineering	9.1	9.5
John	27	Computer Science	9.0	8.6
Mary	45	Administration	8.5	9.7
Peter	26	Engineering	6.5	7.8

Tables and Matrices

We can work with tables using the Pandas library or we can place the useful information in a matrix and work with the NumPy library.

Name	Age	Education	Grade 1	Grade 2
Ane	33	Engineering	9.1	9.5
John	27	Computer Science	9.0	8.6
Mary	45	Administration	8.5	9.7
Peter	26	Engineering	6.5	7.8

$$\begin{bmatrix} 1 & 33 & 1 & 9.1 & 9.5 \\ 2 & 27 & 2 & 9.0 & 8.6 \\ 3 & 45 & 3 & 8.5 & 9.7 \\ 4 & 26 & 1 & 6.5 & 7.8 \end{bmatrix}$$

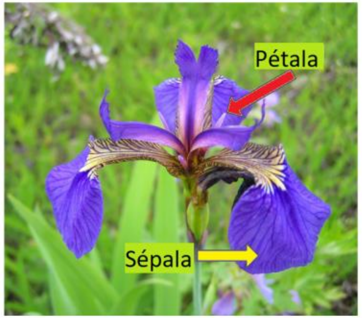
Advantages of vectorization

Data vectorization implies efficiency in processing and saves algorithm execution time.

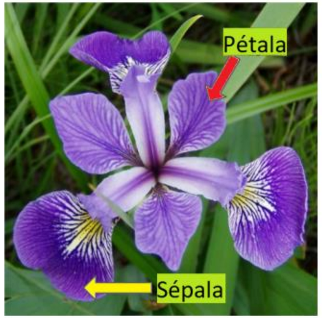
- Concise and more readable code.
- Reduced debugging time due to fewer lines of code.
- Reduced number of loops.

Now, let's see how to manipulate arrays in NumPy.

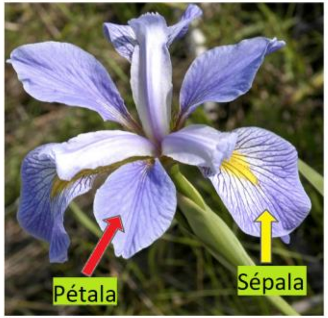
Exercise 1 – Iris Flower Species



Iris setosa



Iris versicolor



Iris virginica

1. Download the dataset: Iris flower dataset available in sklearn.

2. Visualize the shape of the matrix.

3. Find the maximum, minimum, mean and standard deviation values for each variable.

4. How many samples are there for each flower species?

5. Find the maximum, minimum, mean and standard deviation values for each variable of each species.

6. Are there missing data? Are the data consistent with the expected dimensions? Are there outliers? Is the database balanced? Visualize the data through boxplots.

7. Plot the average values for each flower species and the overall average, considering the length of petals and sepals.

8. Plot the heatmap and analyze the correlation between variables.

Exercise 2 - California house-prices

1. Download the dataset: California house-prices available in Sklearn.
2. Visualize the shape of the matrix.
3. Find the maximum and minimum values for each variable.
4. Find the mean and standard deviation for each variable.
5. How many houses have more than 3 bedrooms?
6. Are the data consistent? For example, is the number of rooms always greater than or equal to the number of bedrooms?

For those who had difficulty with Exercise 1.
Try doing Exercise 2 on your own.

Project 1

Combining linear algebra, Python and wines!!!