Artificial Intelligence relies heavily on mathematics. AI algorithms use models that can represent, analyze, and transform digital data. Some of the most important mathematical tools include **linear algebra**, **probability**, and **statistics**. These disciplines make it possible to model data, optimize models and evaluate their performance.

2.1 Linear algebra

Linear algebra forms the basis of Machine Learning and Deep Learning. It allows you to represent the data and parameters of the models in the form of vectors and matrices.

2.1.1 Vectors

A **vector** is an ordered set of numbers. In AI, it often represents an observation or a set of features.

Example :

$$x = (x_1, x_2, \dots, x_n)$$

Each component corresponds to information (size, weight, temperature, intensity, etc.).

The main operations on vectors are:

- Addition,
- Multiplication by a scalar,
- Dot product.

The **dot product** is used to measure the similarity between two vectors:

The **dot product** is used to measure the similarity between two vectors:

$$x \cdot y = \sum_{i=1}^{n} x_i y_i$$

2.1.2 Matrices

A **matrix** is an array of numbers organized into rows and columns. In AI, they are used to represent data sets or transformations.

Example :

$$A \in \mathbb{R}^{mxn}$$

The main operations are:

- Addition,

- Multiplication,
- Transposition,
- Reversal (if possible).

Matrix multiplication is essential for calculating the outputs of neural networks.

2.1.3 Matrix Products

The **matrix product** allows you to combine two compatible matrices:

$$C = A \times B$$

It is used in:

- Propagation in neural networks,
- Data transformation,
- Linear models.

Each neuron essentially performs a product between an input vector and a weight vector.

2.1.4 Standards

A **standard** measures the size or length of a vector.

The most common are :

- L1 standard:

$$\|x\|_1 = \sum |x_i|$$

- L2 standard:

$$\|x\|_2 = \sqrt{\sum x_1^2}$$

- Infinite Standard:

$$\|x\|_\infty = \max|x_i|$$

In AI, standards are used to:

- Measure error,
- Regularize the models,
- Compare vectors.

2.2 Probability and Statistics

Probability and statistics help manage uncertainty, analyze data, and evaluate the performance of AI models.

2.2.1 Random Variables

A **random variable** represents a phenomenon whose value depends on chance.

A distinction is made between:

- Discrete variables (finite number of values),
- Continuous variables (infinite set).

Example: number of defects, temperature, price.

2.2.2 Mathematical Expectation

Expectation is the expected average value of a random variable.

For a discrete variable:

$$E(X) = \sum x_i P(x_i)$$

For a continuous variable:

$$E(X) = \int xf(x)dx$$

In AI, it is used to:

- Evaluate a model,
- Calculate an average loss,
- Optimize algorithms.

2.2.3 Variance and Standard Deviation

The **variance** measures the dispersion around the mean:

$$Var(X) = E[\big(X - E(X)\big)^2]$$

The **standard deviation** is the square root of the variance.

$$\sigma = \sqrt{E[\big(X - E(X)\big)^2]}$$

These measurements help to understand the stability and reliability of data or predictions.

2.2.4 Common Probability Distributions

(a) Uniform Act

All values have the same probability over an interval.

Used for:

- Initialization of weights,

- Simulation.

b) Binomial law

Models the number of successes in a test suite.

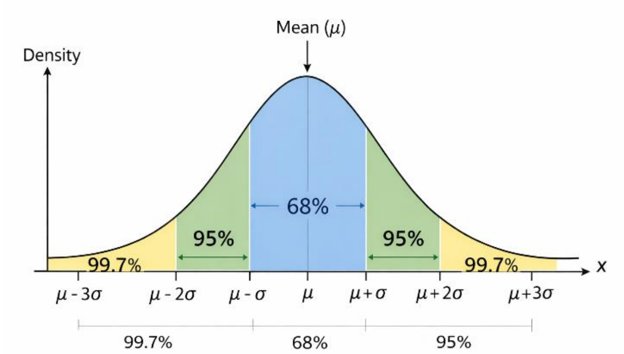
Used for:

- Classification,
- Performance Estimate.

c) Normal distribution (Gaussian)

The most used in AI. It models many natural phenomena.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$



μ (Mean): Slides the bell left or right on the x-axis

Used for:

- Noise,
- Data modeling,
- Standardization.

2.3 Role of mathematics in AI

Mathematics makes it possible to:

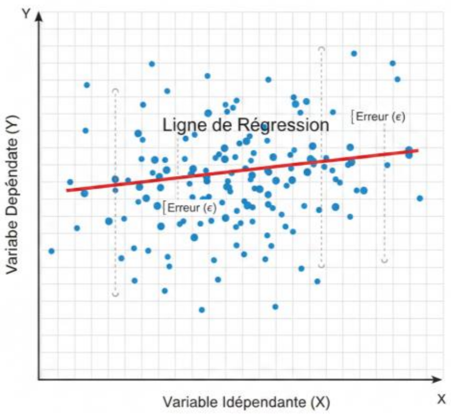
- To represent the data,
- To train the models,
- Optimize the parameters,
- Evaluate the results,
- To guarantee robustness.

Without linear algebra and statistics, there would be no Machine Learning or Deep Learning.

3. Simple linear regression

3.1. Introduction and Purpose

Simple linear regression is one of the fundamental models of supervised machine learning. It allows us to establish a mathematical relationship between an **explanatory variable** x and a **target variable** y.



In mechanics and energy, it is used, for example, to:

- Predict a power from a speed,
- Estimate energy consumption according to load,
- Model a temperature as a function of time or flow,
- Connecting a pressure to a flow rate in a fluid system.

The goal is to automatically learn a law of the type:

$$y = f(x)$$

from experimental data.

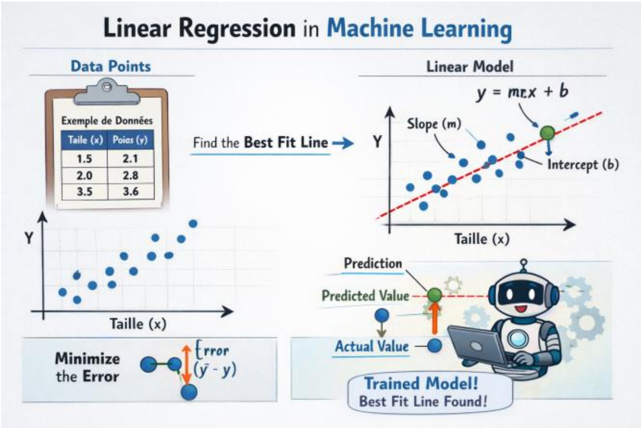
3.2. Mathematical Formulation

Simple linear regression assumes that the relationship between x and y is **linear**:

$$y = wx + b$$

where:

- *w* is the slope (coefficient),
- *b* is the bias (ordinate at the origin),
- *x* is the input variable,
- *there* is the predicted outcome.



For a dataset N, the model becomes:$\{(x_i, y_i)\}_{i=1}^{N}$

$$\hat{y}_i = wx_i + b$$

3.3. Cost Function

To measure the error of the model, a **cost function is defined**. The most commonly used is the **root mean square error (MSE):**

$$J(\omega, b) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

With :

$$\hat{y}_i = wx_i + b$$

This feature strongly penalizes large errors and allows for stable optimization.

Physical interpretation:

- If J is tall→ the pattern is bad.
- If J is small → the model represents the real phenomenon well.

We are therefore looking for:

$$\min_{w,b} J(\omega, b)$$

3.4. Optimization

a) Principle

Optimization is the process of finding the w and b parameters that minimize the cost function.

There are two main methods:

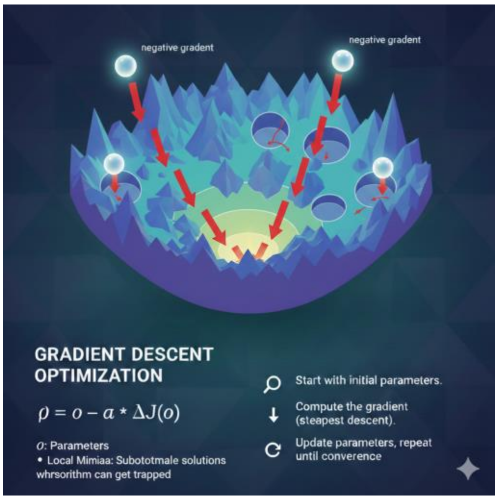
- The analytical solution (least squares),
- The gradient descent.

(b) Gradient descent

The gradient descent gradually adjusts the parameters:

$$w := w - \alpha \frac{\partial J}{\partial w}$$

$$b := b - \alpha \frac{\partial J}{\partial b}$$



Where:

- α is the learning rate,
- are the gradients.

The derivatives are:

$$\frac{\partial J}{\partial w} = \frac{2}{N} \sum x_i (y_i - \hat{y}_i)$$

$$\frac{\partial J}{\partial b} = \frac{2}{N} \sum (y_i - \hat{y}_i)$$

The process is repeated until convergence.

c) Physical sense

At each iteration:

- We measure the error,
- We correct the slope,
- We correct the bias,
- We improve the prediction.

This simulates a gradual adjustment of the physical law from the data.

3.5. Implementation with Scikit-learn

Scikit-learn provides an optimized implementation of linear regression.

(a) Importation

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

b) Example of energy data

```
# Données : vitesse -> puissance
X = np.array([10, 20, 30, 40, 50]).reshape(-1,1)
y = np.array([15, 28, 45, 60, 80])
```

c) Separation of learning and testing

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

d) Creation and training

```
model = LinearRegression()
model.fit(X_train, y_train)
```

e) Model parameters

```
print("Pente w =", model.coef_[0])
print("Biais b =", model.intercept_)
```

(f) Prediction

```
y_pred = model.predict(X_test)
```

g) Visualization

```
plt.scatter(X, y, label="Données réelles")
plt.plot(X, model.predict(X), color='red', label="Modèle")
plt.xlabel("Vitesse")
plt.ylabel("Puissance")
plt.legend()
plt.show()
```

3.6. Model evaluation

The quality of the model is measured with MSE:

```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_test, y_pred)
print("MSE =", mse)
```

3.7. Link with mechanics and energy

Typical applications:

- Estimation of electricity consumption,
- Thermal modeling,
- Efficiency of a machine,
- Flow-pressure laws,
- Speed-power,
- Wear and tear.

Linear regression makes it possible to transform physical measurements into **usable models for prediction and industrial optimization**.