Artificial Intelligence
CSC411

Lecture Set-01
Introduction to AI

22 BSCS

Dr. Ali Asghar Manjotho
Assistant Professor, CSE-MUET

ali.manjotho@faculty.muet.edu.pk
ali.manjotho.ali@gmail.com

Contents

- What is Artificial Intelligence?
- When and when not to use AI?
- Definitions of AI
    - Thinking Humanly
    - Thinking Rationally
    - Acting Humanly
    - Acting Rationally
- Approaches to Implement AI
- Types of AI

Contents

Contents

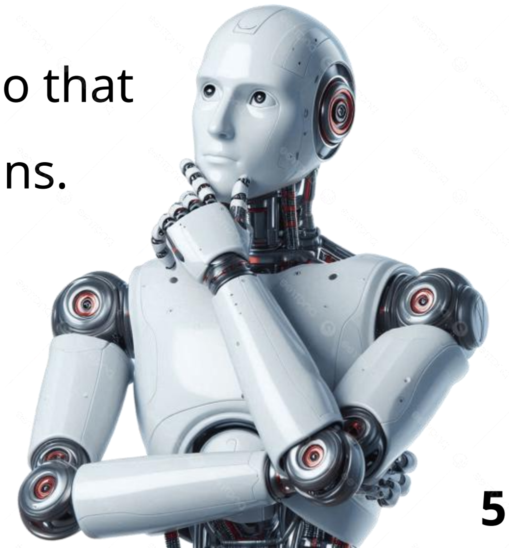
What is Artificial Intelligence (AI)?

- Artificial Intelligence is the study of how to make computers do things at which, at the moment, people are better.

- Embedding human intelligence in to the machines so that they can **Think**, **Act**, and **Make Decisions** like humans.

Artificial = Man made

Intelligence = Ability to think and reason



When and when not to use AI?

- AI is **not a magic** solution to everything.
- **Don’t use AI when:**
  - The **procedure is known** ▶ You already know exactly how to solve the problem.
  - You can write a simple **traditional program** ▶ Rule-based logic is enough.
  - The **rules are finite and easy to encode** ▶ The machine doesn’t need to learn; it just needs instructions.
- This means AI is ***overkill*** for clear, deterministic problems.

When and when not to use AI?

- **Use AI when:**
  - The **procedure is unknown** ▶ You cannot manually define the steps.
  - A **traditional program cannot be written** ▶ The logic is too complex, vague, or contextual.
  - The **rules are infinite or impossible** to define ▶ The machine must learn patterns from data.
- This ***aligns perfectly*** with modern machine learning practice.

When and when not to use AI?

Non-AI Tasks

- Printing a document
- Barcode scanning
- Sorting numbers
- Calculating tax
- Temperature conversion
- Detecting if a number is prime

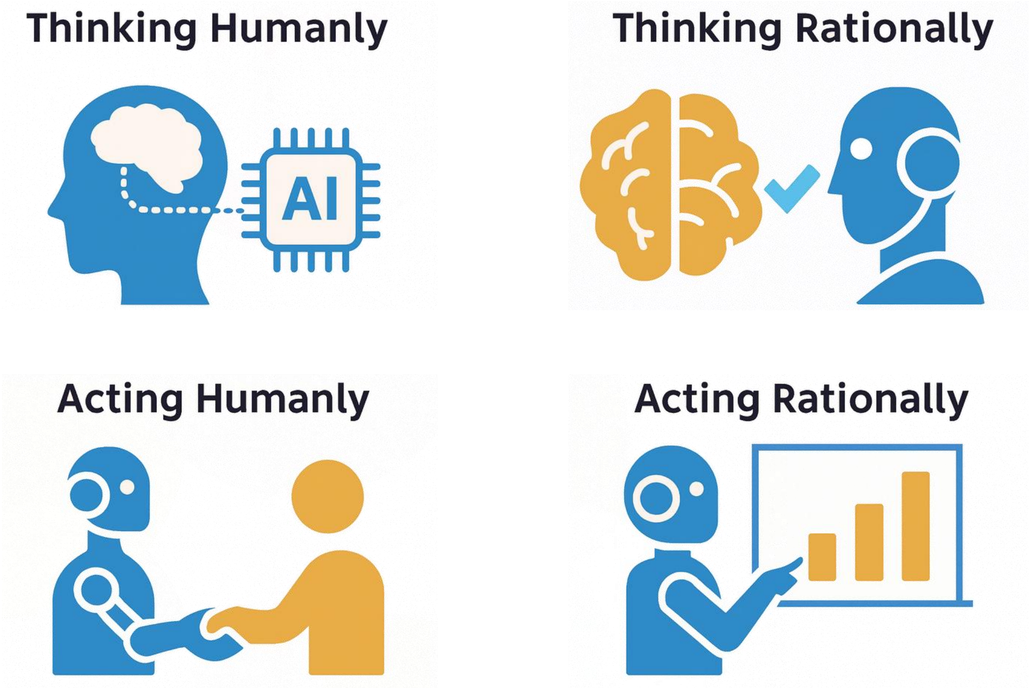
AI Tasks

- Speech recognition
- Image classification
- Medical diagnosis
- Natural language understanding
- Face recognition
- Predicting stock market trends
- Recommender system

Definitions of AI

Human vs. Rational

Thought vs. Behavior



Definitions of AI

Thinking Humanly	Thinking Rationally
Make machines think the way humans think—replicating human reasoning, thought processes, and decision-making. (Understanding and modeling how the human mind works)	Make machines think logically based on rules, evidence, or formal reasoning to determine what is logically correct. (Ideal, logical reasoning — what is "correct" according to logic)
Acting Humanly	**Acting Rationally**
Make machines behave like humans—responding, interacting, and acting the way a human would in a given situation. (Human-like behavior, Turing Test)	Make machines take the best possible action in a given situation to achieve their goals. (Rational agents — choosing actions that maximize success)

Definitions of AI

Thinking Humanly	Thinking Rationally
Machines that think like humans by imitating human cognitive processes	Machines that think logically, based on principles of reasoning and correctness
Acting Humanly	**Acting Rationally**
Machines that act like humans, reproducing human behavior in specific situations	Machines that act optimally, choosing the best action to achieve goals

Definitions of AI

Thinking Humanly	Thinking Rationally
"The exciting new effort to make computers think ... *machines with minds*, in the full and literal sense." (Haugeland, 1985) "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning ..." (Bellman, 1978)	"The study of mental faculties through the use of computational models." (Charniak and McDermott, 1985) "The study of the computations that make it possible to perceive, reason, and act." (Winston, 1992)
Acting Humanly	**Acting Rationally**
"The art of creating machines that perform functions that require intelligence when performed by people." (Kurzweil, 1990) "The study of how to make computers do things at which, at the moment, people are better." (Rich and Knight, 1991)	"Computational Intelligence is the study of the design of intelligent agents." (Poole *et al.*, 1998) "AI ...is concerned with intelligent behavior in artifacts." (Nilsson, 1998)

Thinking Humanly

AI that tries to think the way humans think ▶ cognitive modeling.

Examples:

- **Cognitive architectures like ACT-R:**
  Models human memory, learning, and reasoning.
- **Simulated problem-solving experiments:**
  AI systems replicating how humans solve puzzles or math problems.
- **Human-like reasoning systems:**
  Systems designed to mimic human thought errors or biases.
- **Psychological AI models:**
  AI that models human decision-making processes.

Key idea: The system tries to reproduce the mental steps a human would take.

Thinking Rationally

AI that uses logic and mathematical reasoning ▶ what is correct.

Examples:

- **Logic-based expert systems:**
  Medical diagnosis systems using rules: `IF fever AND rash → measles`.
- **Theorem provers:**
  Systems that prove mathematical statements logically (e.g., Prolog program).
- **Knowledge-based reasoning systems:**
  AI that uses propositional/predicate logic to derive conclusions.
- **Automated planning systems using formal logic**
  Uses formal logic to determine the correct action sequence to achieve a goal.

Key idea: The system uses logic, not human-style thinking.

Acting Humanly

AI that tries to behave like a human ▶ Turing test focus.

Examples:

- **Chatbots designed to behave like humans:**
  Systems that try to pass the Turing Test.
- **Humanoid robots performing human-like tasks:**
  E.g., Pepper robot greeting customers.
- **Voice assistants mimicking human conversation:**
  Siri, Alexa, Google Assistant (human-like responses).
- **Gesture or emotion-mimicking robots:**
  Robots that use human-like facial expressions or gestures.

Key idea: Behavior should seem *human*, even if the internal thinking is different.

Acting Rationally

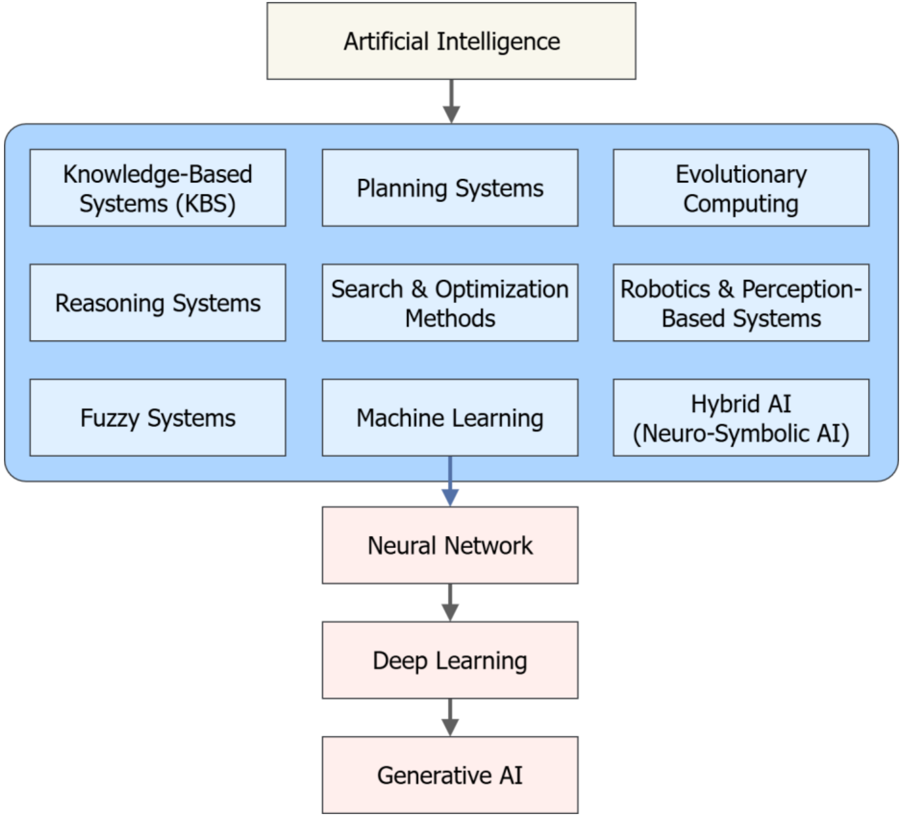
AI that takes the optimal action ▶ rational agent approach.

Examples:

- **Self-driving cars and Autonomous drones:**
  Choose best actions to drive safely, navigate obstacles and optimize flight paths.
- **Robotic vacuum cleaners:**
  Plan paths to clean efficiently based on sensors.
- **Recommendation systems:**
  Recommend the best product/movie to maximize user satisfaction.
- **Game-playing agents:**
  Chess/Go AI choosing moves to maximize winning chance.

Key idea: AI acts to maximize goal achievement, not necessarily like a human.

Approaches to Implement AI



Approaches to Implement AI

Machine Learning (ML): ML systems learn patterns from data instead of being manually programmed.

Examples:

- Neural Networks
- Decision Trees
- Support Vector Machines
- Random Forest

- **ML is currently the most dominant approach.**

Approaches to Implement AI

- Sub fields of ML include:
  - Neural Network (NN): Inspired from human brain neurons.
  - Deep Learning (DL): Extension of NN.
  - Generative AI (GenAI): GenAI is the DL for generating new data.

Approaches to Implement AI

Knowledge-Based Systems (KBS): AI is achieved through explicit rules, logic, and knowledge representation, not data.

Examples:

- Expert Systems
- Logic-based reasoning systems
- Ontologies and rule engines

- **KBS AI encodes human knowledge manually.**

Approaches to Implement AI

Search & Optimization Methods: **AI solves problems by exploring possible solutions.**

Examples:

- A* Search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Hill climbing
- Genetic algorithms

- **Many classical AI problems (like pathfinding) rely on search, not learning.**

Approaches to Implement AI

Reasoning Systems (Logical AI): AI uses formal logic to reason and make decisions.

Examples:

- Propositional logic
- Predicate logic
- Constraint satisfaction problems

- **This approach focuses on thinking rationally.**

Approaches to Implement AI

Planning Systems: AI achieves goals by generating an optimal sequence of actions.

Examples:

- Temporal Planning Systems: Handle actions with time constraints
- Motion Planning Algorithms: Robot navigation
- AI Planning in Games: NPC behavior planning, strategy generation

- **Planning systems help AI choose not just what to do, but how and when to do it to achieve goals efficiently.**

Approaches to Implement AI

Robotics & Perception-Based Systems: AI becomes intelligent through interaction with the physical world.

Examples:

- Sensor-based navigation
- SLAM (Simultaneous Localization and Mapping) in autonomous vehicles

- **This emphasizes acting rationally in dynamic environments.**

Approaches to Implement AI

Evolutionary Computation: AI is achieved through biologically inspired evolution.

Examples:

- Genetic Algorithms
- Genetic Programming
- Evolution Strategies

- **These systems evolve solutions over generations.**

Approaches to Implement AI

Fuzzy Systems: AI handles imprecision and uncertainty using fuzzy logic.

Examples:

- Fuzzy control systems in appliances
- Decision systems involving vague conditions

Fuzzy quantities:

- Low, high
- Slightly far
- Very close

Vague conditions:

If the temperature is *somewhat high*, set the fan speed to *moderate*.

- **Fuzzy systems provide a powerful way for AI to operate in environments with uncertainty and imprecision, offering flexible decision-making capabilities.**

Approaches to Implement AI

Hybrid AI (Neuro-Symbolic AI): AI combining symbolic AI (logic, rules) and machine learning (patterns from data).

Examples:

- Explainable AI (XAI): transparency of symbolic reasoning + power of ML to create interpretable models
- Legal reasoning: legal rules + data-driven insights to assist in legal decision-making
- Scientific discovery: logical reasoning + experimental data to formulate hypotheses and discover new scientific principles

- **Hybrid AI integrates the best of both worlds: the interpretability and structure of symbolic AI with the adaptability and power of machine learning, enabling more robust, transparent, and intelligent systems.**

Types of AI

- Types of AI
  - Artificial Narrow Intelligence (ANI)
  - Artificial General Intelligence (AGI)
  - Artificial Super Intelligence (Super AI)

Types of AI

- Artificial Narrow Intelligence, also known as **Weak AI**, is the only type of AI that exists today.
- Any other form of AI is theoretical.
- It can be trained to perform a **single** or **narrow task**, often far faster and better than a human mind can.
    - Siri, Alexa
    - ChatGPT, Claude, DeepSeek
    - Autonomous Vehicles

Types of AI

- Artificial General Intelligence (AGI), also known as **Strong AI**, is today nothing more than a theoretical concept.
- AGI can use previous learnings and skills to accomplish new tasks in a different context without the need for human beings to train the underlying models.
- This ability allows AGI to learn and perform any intellectual task that a human being can.

Types of AI

- Super AI is commonly referred to as **artificial superintelligence** and, like AGI, is strictly theoretical.
- If ever realized, Super AI would think, reason, learn, make judgements and possess cognitive abilities that **surpass** those of human beings.
- It surpasses intelligence of human in solving-problem, creativity, and overall abilities.

AI Paradigms

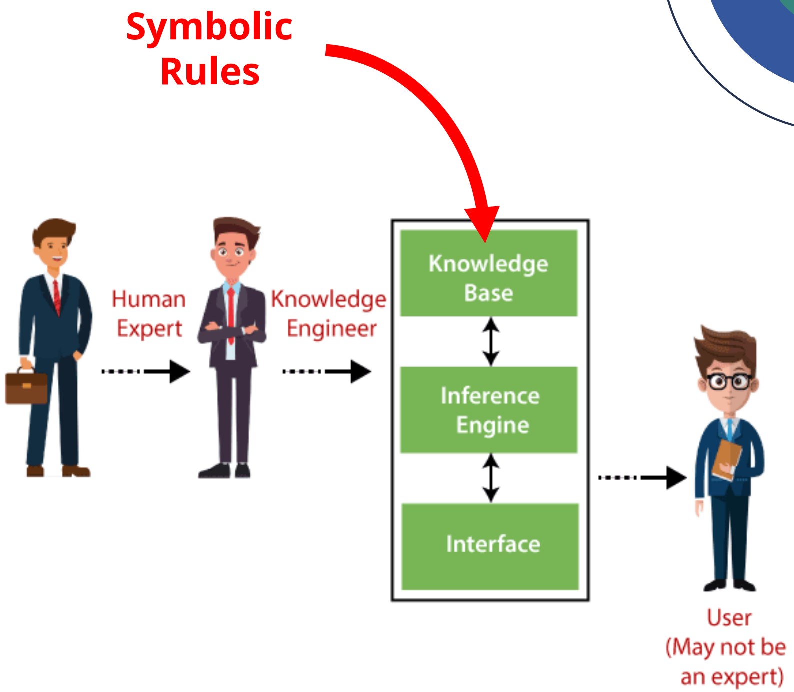
- Artificial Intelligence (AI) is a vast field with various approaches to creating intelligent systems.
- Two primary paradigms within AI are:

Symbolic AI

Connectionist AI

AI Paradigms – Symbolic AI

- Symbolic AI, also known as **classical AI**, represents knowledge explicitly using symbols and rules.
- It relies on logic and formal reasoning to solve problems and make decisions.



AI Paradigms – Symbolic AI

- **Example: MYCIN**, an early AI system for treating blood infections.
- Instead of learning from data, MYCIN used hundreds of **`IF–THEN`** rules created by doctors.

Organism Detection Rules

```
IF the infection is bacterial
AND the gram stain is positive
AND the shape is spherical
THEN the organism is likely streptococcus
```

Antibiotics Recommendation

```
IF the patient is allergic to penicillin
THEN avoid prescribing penicillin-based antibiotics
```

AI Paradigms – Symbolic AI

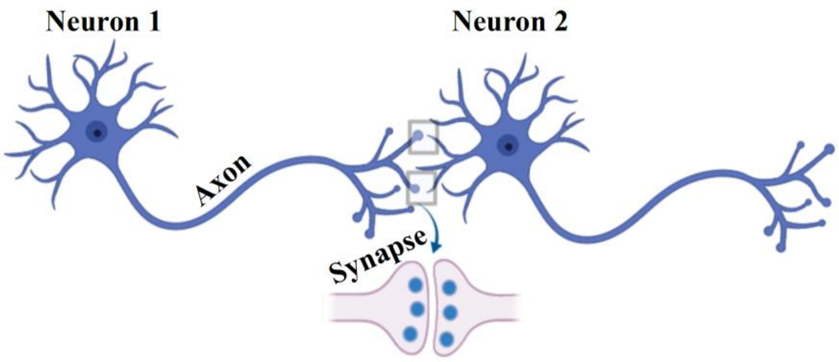
- **Pros of Symbolic AI:**
  - **Transparency:** Symbolic AI systems are interpretable and transparent, as their reasoning process can be easily understood and traced.
  - **Logical Reasoning:** These systems excel in tasks requiring logical reasoning and problem-solving, such as theorem proving and expert systems.
  - **Knowledge Representation:** Symbolic AI can effectively represent complex knowledge structures and relationships using symbols and rules.

AI Paradigms – Symbolic AI

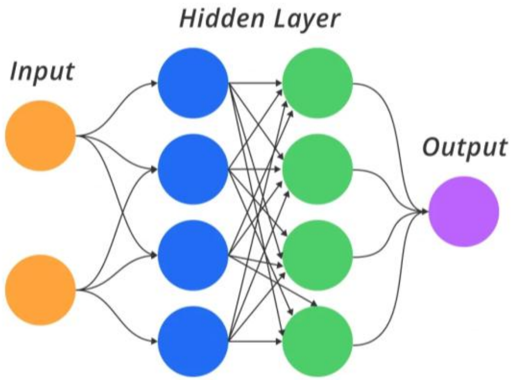
- **Cons of Symbolic AI:**
  - **Scalability:** Symbolic AI struggles with scalability when dealing with large, complex, and dynamic datasets.
  - **Flexibility:** These systems are less flexible and adaptable to new, unseen situations or changes in the environment.
  - **Learning:** Symbolic AI relies on hand-coded rules and knowledge, making it difficult to learn from raw data and adapt over time.

AI Paradigms – Connectionist AI

- Connectionist AI, also known as **neural networks** or **sub-symbolic AI**, represents knowledge through connections and weights within a network of artificial neurons.
- It is inspired by the structure and functioning of the human brain.



Biological Neurons and Connections



Artificial Neurons and Connections

AI Paradigms – Connectionist AI

- **Pros of Connectionist AI:**
  - **Learning from Data:** Connectionist AI excels at learning from large amounts of data, making it suitable for tasks like image and speech recognition.
  - **Adaptability:** These systems can adapt to new data and environments through training, improving their performance over time.
  - **Pattern Recognition:** Neural networks are highly effective at recognizing patterns and making predictions based on learned representations.

AI Paradigms – Connectionist AI

- **Cons of Connectionist AI:**
  - **Interpretability:** Connectionist AI systems are often seen as "black boxes" due to their lack of transparency and interpretability.
  - **Resource Intensive:** Training neural networks requires significant computational resources and large datasets.
  - **Overfitting:** These systems are prone to overfitting, where they perform well on training data but poorly on new, unseen data.

Symbolic AI vs. Connectionist AI

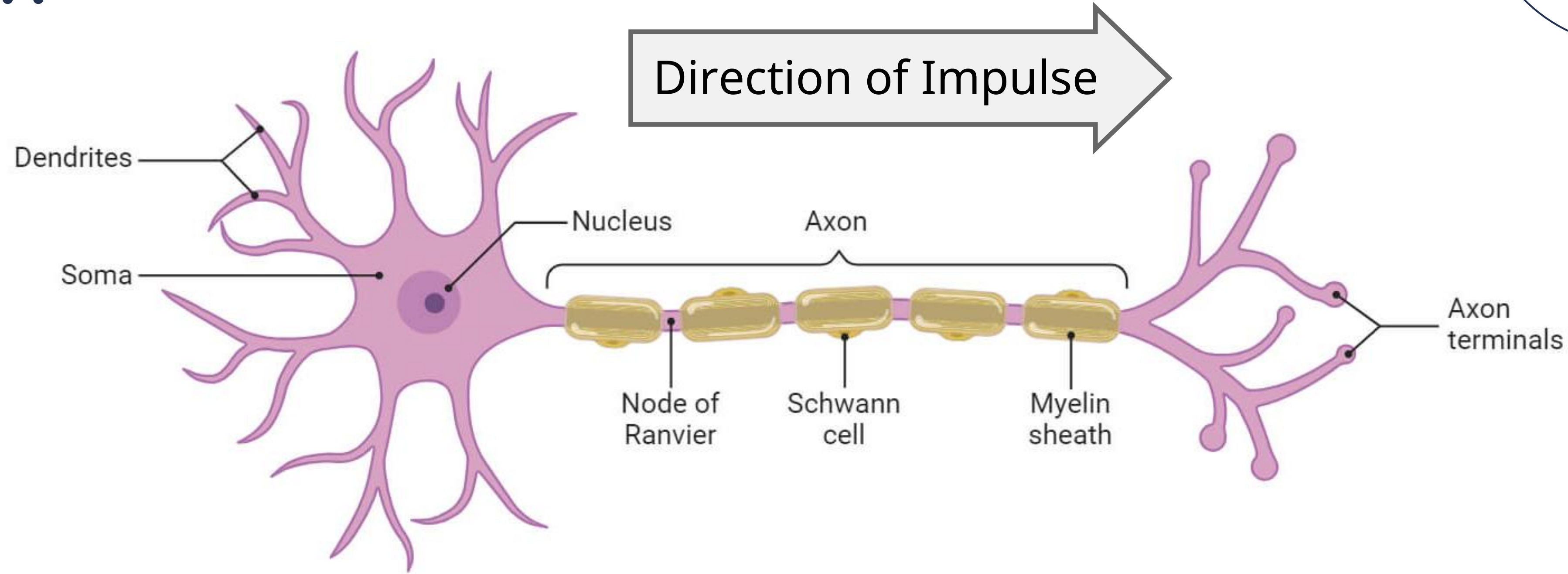
Feature	Symbolic AI	Connectionist AI
Knowledge Representation	Uses explicit symbols and rules	Uses connections and weights in networks
Transparency	High, interpretable and traceable	Low, often considered a "black box"
Reasoning Ability	Excels in logical reasoning and problem-solving	Excels in pattern recognition and prediction
Scalability	Struggles with large, complex datasets	Scales well with large datasets
Flexibility	Less flexible and adaptable	Highly adaptable through training
Learning Method	Relies on hand-coded rules	Learns from data through training
Resource Requirements	Generally less resource-intensive	Requires significant computational resources
Risk of Overfitting	Lower risk	Higher risk

Nerve Cells (Neurons)

- Human brain is composed of million of processing elements called **nerve cells** or **neurons**.
- The main function of nerve cells is to **receive**, **process**, and **transmit** information.
- These cells:
  - Receive signals from different sensory organs or other neurons
  - Process this information
  - Transmit signals to target cells, such as other neurons, muscles, or other organs.

Nerve Cells (Neurons)

Anatomy of a Neuron

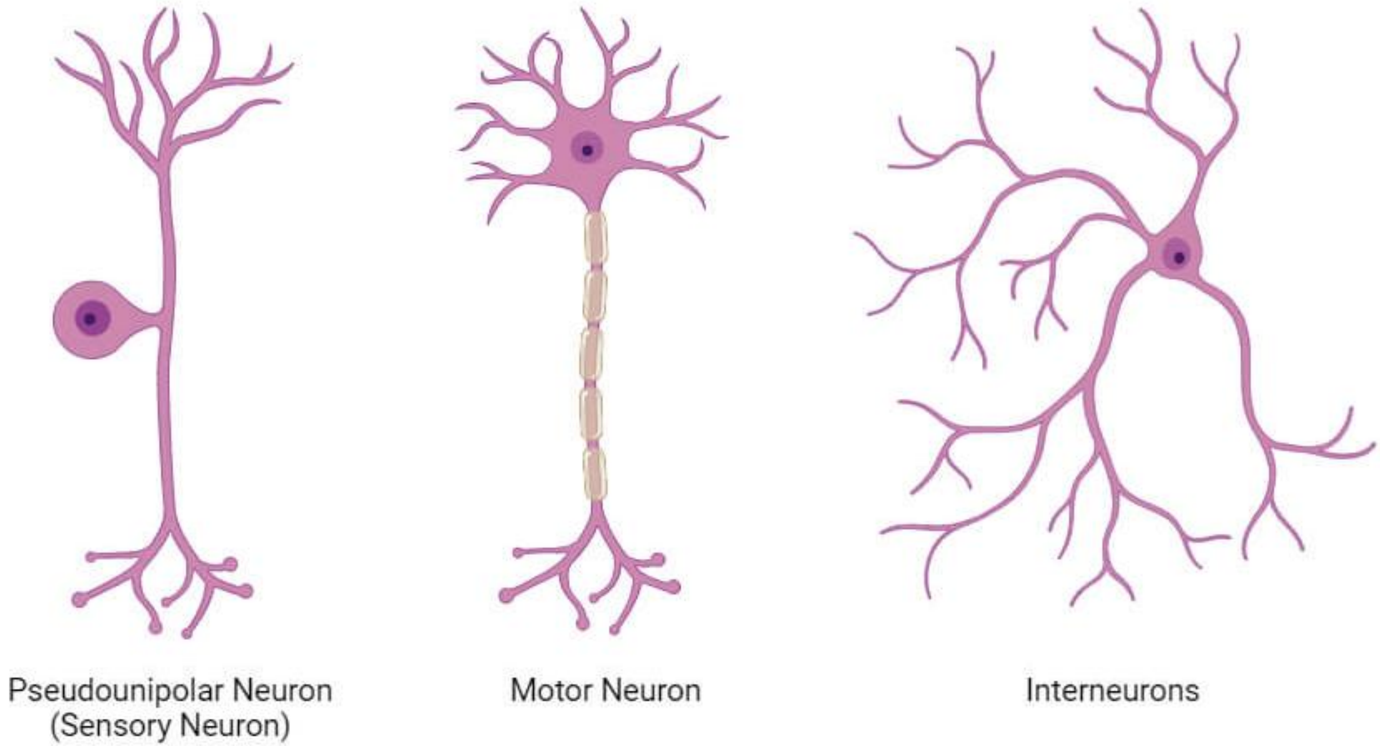


Types of Nerve Cells (Neurons)

- Nerve cells are classified into three main types based on their function.

1. **Sensory neurons** are nerve cells that detect and transmit signals from the external environment to the central nervous system (CNS).
2. **Motor neurons** are nerve cells that transmit signals from the CNS to muscles. These neurons initiate and control voluntary and involuntary muscle movements.
3. **Interneurons** are the most abundant type of nerve cells and are only found in the central nervous system. These cells act as connectors between sensory and motor nerve cells.

Types of Nerve Cells (Neurons)



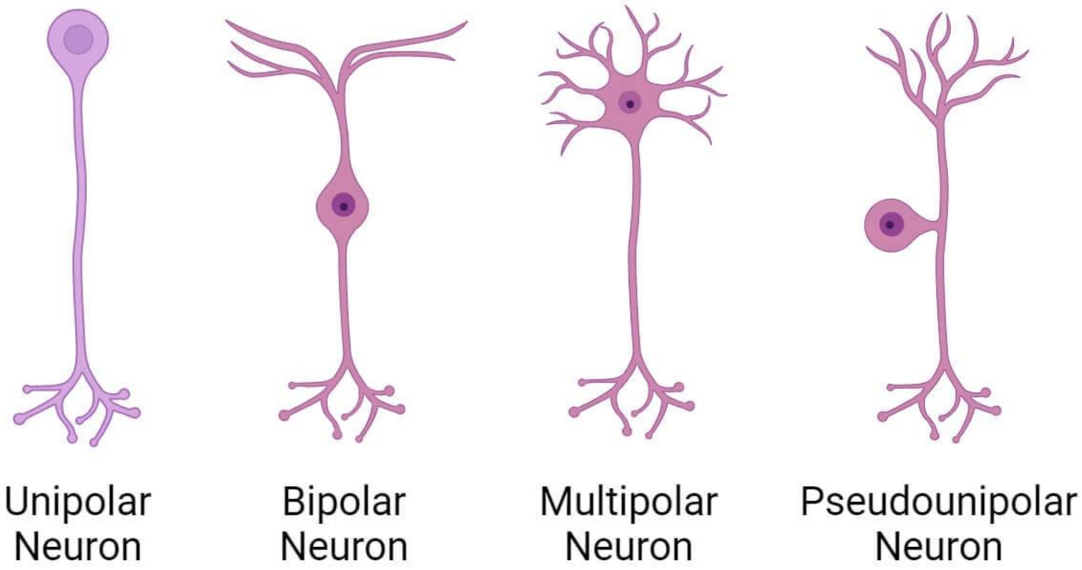
Types of Nerve Cells (Neurons)

- Nerve cells can also be classified into four types based on their structure.

1. **Unipolar neurons** have a single structure extending from the cell body which contains one axon with dendrites. They are commonly found in invertebrates.
2. **Bipolar neurons** have two structures – one axon and one dendrite, extending from the cell body.
3. **Multipolar neurons** contain one axon and multiple symmetrical dendrites extending from the cell body. This is the most common type of neuron.

Types of Nerve Cells (Neurons)

4. **Pseudounipolar neurons** have only one process that extends from the cell body, which separates into two structures. These nerve cells do not have dendrites.



Nerve Impulses and Action Potential

- Nerve impulses are the signals transmitted through the nervous system which allows communication between neurons.
- Nerve cells receive and transmit information in the form of electrical signals throughout the body.
- Nerve impulses from one neuron are passed onto another through sites called **synapses**.
- Before a nerve impulse is transmitted through a synapse, an action potential must be generated in the presynaptic neuron.
- The action potential refers to a **rapid change** in the the **membrane potential**.

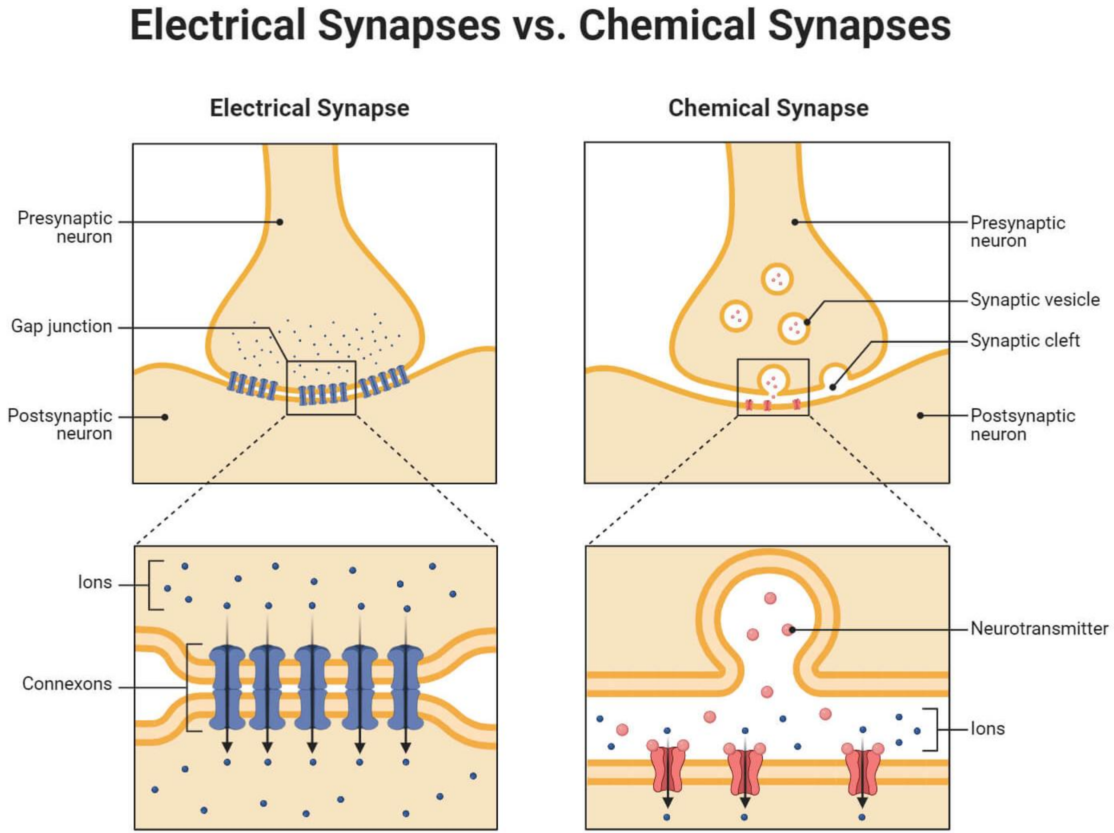
Nerve Impulses and Action Potential

- Neurons allow communication within the nervous system which is vital for different body functions.
- Sensory neurons receive stimuli from the external environment. These nerve cells detect changes in the surroundings and transmit these signals to the central nervous system. This allows us to respond to our surroundings.
- Motor neurons receive signals from the brain and transmit these signals to muscles. They control muscle movements and allow voluntary movements like walking and running.

Nerve Impulses and Action Potential

- Interneurons act as mediators and transmit signals between other neurons.
- Nerve cells are also involved in maintaining the concentrations of ions inside and outside their membranes which is important for generating and transmitting nerve impulses.

Nerve Impulses and Action Potential



The Evolution of Artificial Neurons

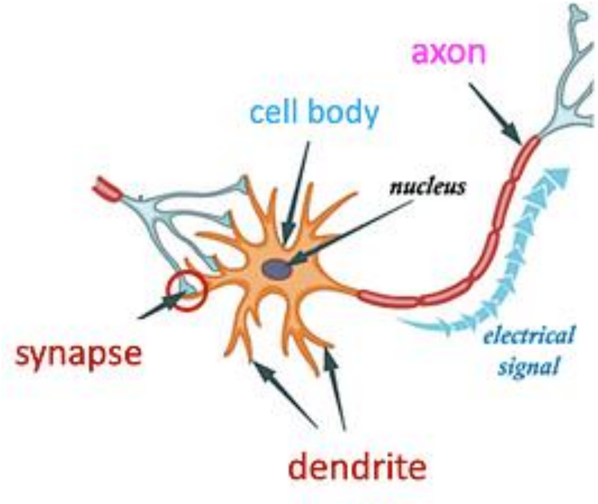
- The evolution of artificial neurons witnessed transition from **McCullock-Pitts neuron** to **Perceptron** and **ADALINE**.
- Artificial neurons, despite their simplicity, can **mimic** the information processing capabilities of biological neurons.
- The development of these neurons, along with that of perceptron, represents a significant turning point in the history of artificial intelligence and neural networks.

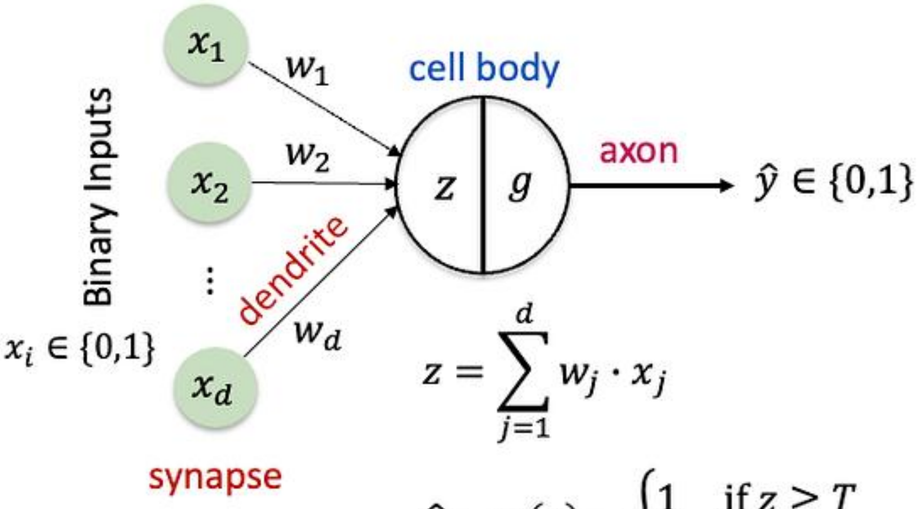
The Evolution of Artificial Neurons

- The evolution of artificial neurons witnessed transition from **McCullock-Pitts neuron** to **Perceptron** and **ADALINE**.
- Artificial neurons, despite their simplicity, can **mimic** the information processing capabilities of biological neurons.
- The development of these neurons, along with that of perceptron, represents a significant turning point in the history of artificial intelligence and neural networks.

McCulloch-Pitts Neuron

- The origins of neural networks can be traced back to the **McCulloch-Pitts (MP) neuron**, a groundbreaking concept introduced by Warren McCulloch and Walter Pitts in 1943.





$$\hat{y} = g(z) = \begin{cases} 1 & \text{if } z \geq T \\ 0 & \text{otherwise} \end{cases}$$

McCulloch-Pitts Neuron

- MP neuron is a highly simplified computational model, which aggregates inputs with a linear function $z(\cdot)$ and makes a decision with a threshold function $g(\cdot)$.

Key Features:

- **Binary Input & Output:** The model operates only with **binary values** (0 or 1).
- **Weighted Inputs:** Each input is multiplied by a weight w_j before being summed. This weighting mechanism allows the model to assign different levels of importance to each input, mimicking the synaptic strengths and dendrite in biological neurons.

McCulloch-Pitts Neuron

- **Fixed Threshold for Activation:** The MP neuron activates if the sum of its weighted inputs exceeds a predefined **threshold** T. This threshold acts as a decision boundary, determining whether the neuron will fire or not.
- **All-or-Nothing Activation Function:** The neuron either fires (**output = 1**) or does not fire (**output = 0**).

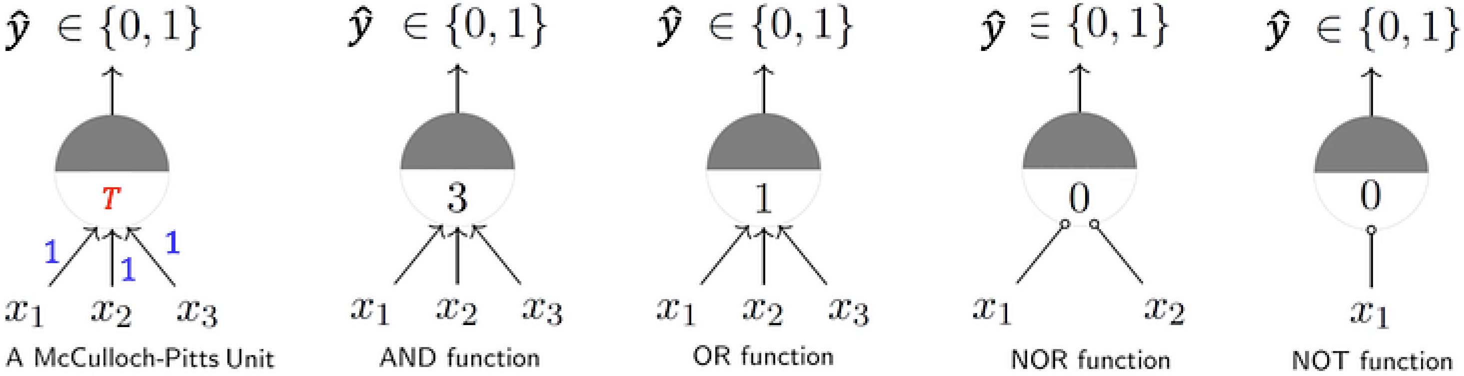
McCulloch-Pitts Neuron

- The MP neuron can compute simple Boolean functions $f : \{0,1\} \rightarrow \{0,1\}$ by selecting appropriate threshold parameters and weights.
- This capability allows the model to perform basic logical operations such as AND, OR, and NOT, which are fundamental to digital computation.
- However, the weights and threshold in the MP neuron were manually set for specific functions.
- Moreover, no automated learning method was developed to identify these parameters for desired functions, which greatly restricted its practical applications.

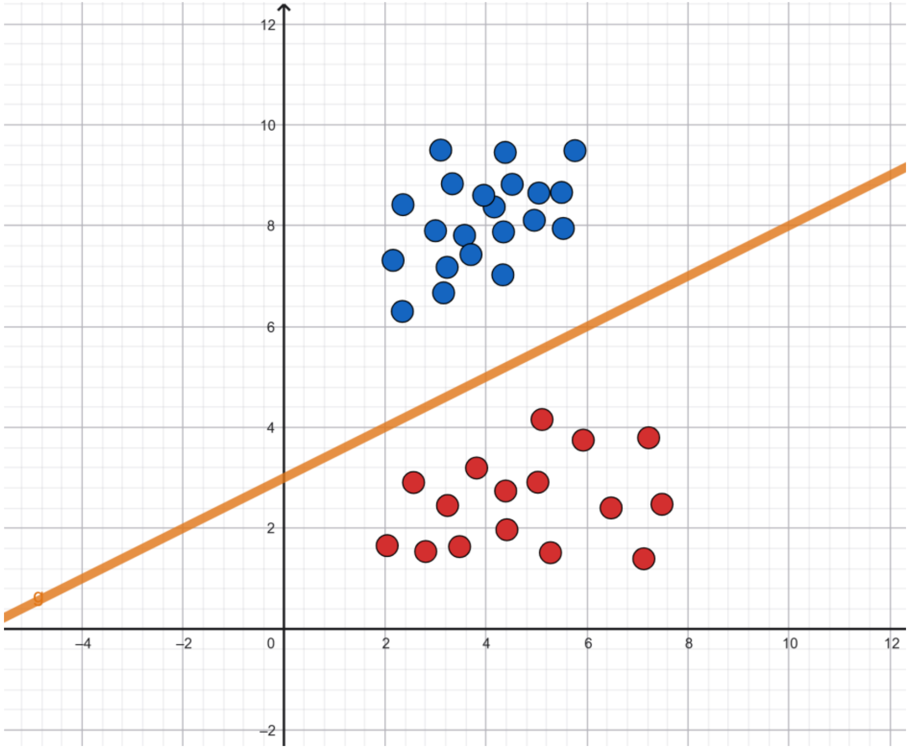
McCulloch-Pitts Neuron

$$\hat{y} = f(\mathbf{x}) = g(z(\mathbf{x})) = \begin{cases} 0 & \text{if } z(\mathbf{x}) < T \\ 1 & \text{if } z(\mathbf{x}) \geq T \end{cases}$$

For all $w_j = 1$,



McCulloch-Pitts Neuron



- A single MP-Neuron **linearly separates** the data with a straight line.
- The line can be modeled as:

$$y = mx + b$$

$$y = 0.5x + 3$$

m = slope of line

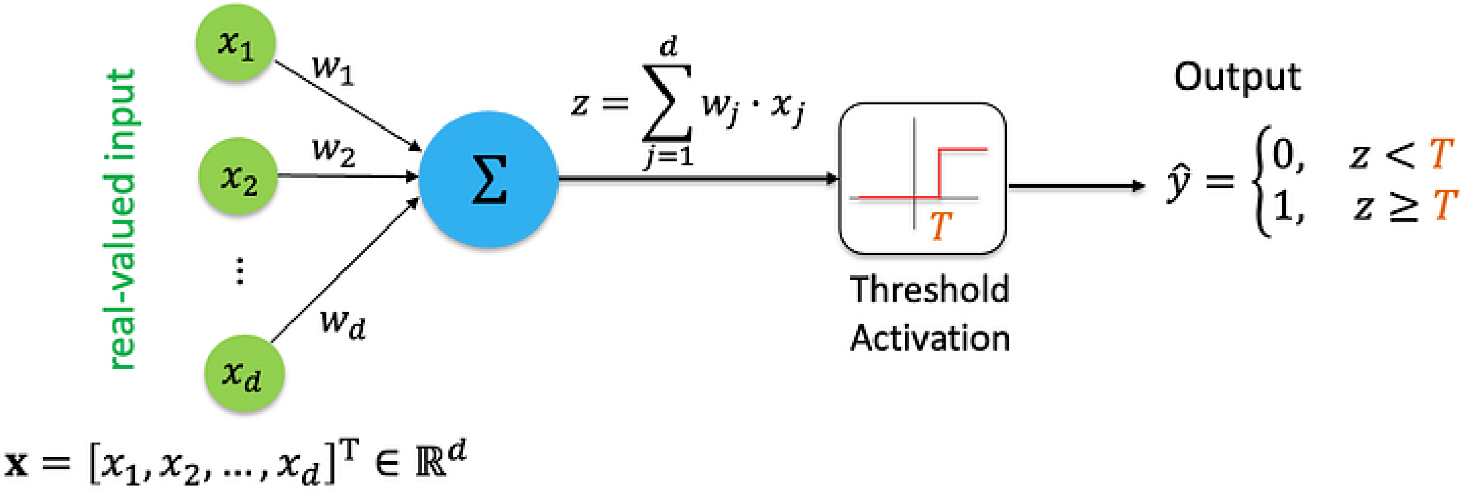
b = y-intercept

- m and b are the two **learnable parameters** of the MP-Neuron.

Rosenblatt’s Perceptron

- Building upon the MP neuron model, Frank Rosenblatt’s perceptron model in 1957 marked a major advancement.
- Real-valued Inputs: one of the first key innovations of the perceptron is its ability to handle real-valued inputs, enabling it to approximate a wide range of complex functions.
- The modern perceptron model builds upon the foundational concepts of the MP-neuron and Rosenblatt’s perceptron, but adds a bias term to simplify the model. This enhancement increases the model’s flexibility and ease of use.

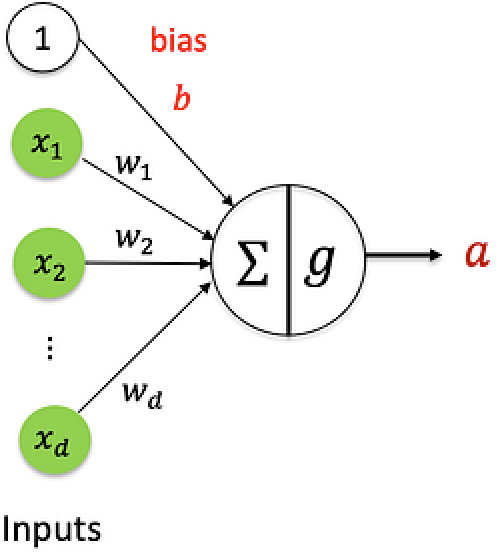
Rosenblatt's Perceptron



Rosenblatt's Perceptron

- In the original perceptron model, a threshold T is used to determine whether the neuron activates.
- To simplify the model, the threshold can be replaced by a bias term b, where $b=-T$.
- This modification allows the activation function to be represented as a unit step function $u(z)$, where z is the net input to the neuron.

Rosenblatt’s Perceptron



$\mathbf{x} = [x_1, x_2, \dots, x_d]^{\mathrm{T}}$ $\quad$ $\mathbf{w} = [w_1, w_2, \dots, w_d]^{\mathrm{T}}$

Net Input

$$z = b + \sum_{j=1}^{d} w_j\, x_j = b + \mathbf{w}^{\mathrm{T}}\mathbf{x} \qquad \text{where } b = -T$$

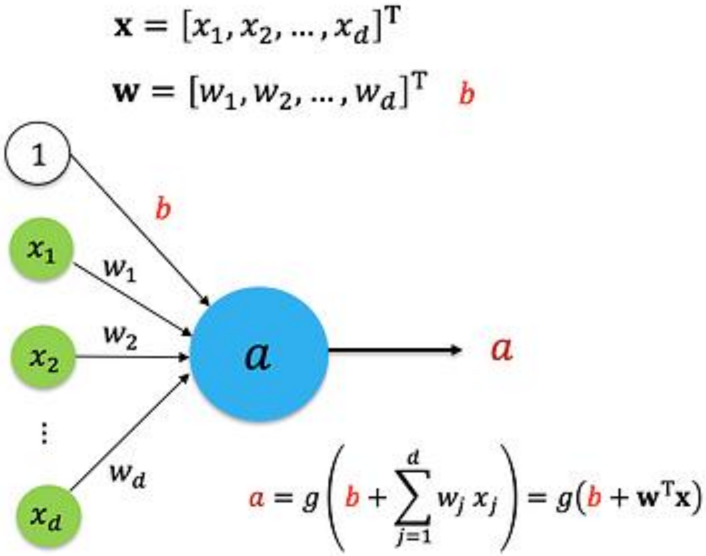
$$a = g(z) = g\left(b + \mathbf{w}^{\mathrm{T}}\mathbf{x}\right)$$

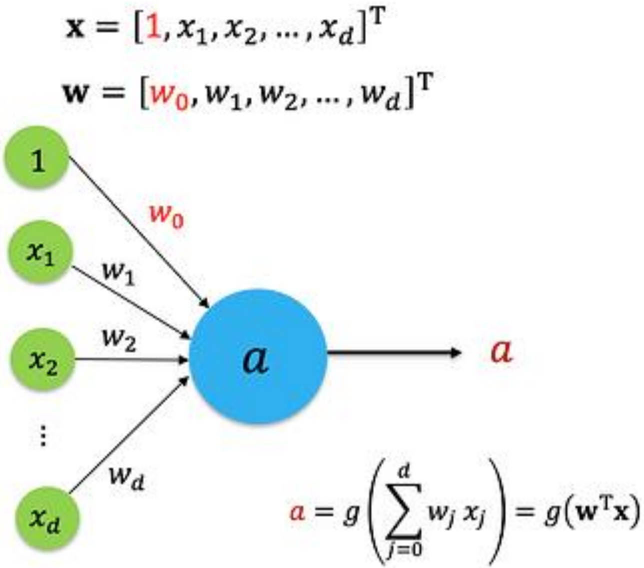
In **original Perception**, the activation function is a **Binary Step function** $u(z)$:

$$a = u(z) = \begin{cases} 0, & \text{for } z < 0 \\ 1, & \text{for } z \geq 0 \end{cases}$$

Rosenblatt’s Perceptron

- The two most used modern perceptron model representations are:





Perceptron Learning Rule

- Frank Rosenblatt also introduced a groundbreaking perceptron learning rule.
- This rule allows the model to adjust its weights iteratively based on the error in its predictions of the training examples.
- Unlike the static nature of the MP neuron, the perceptron learning rule enables the model to learn from data, making it more adaptable and capable of improving its performance over time.
- The learning rule works by comparing the perceptron's output to the desired output and updating the weights to minimize the difference, thereby enhancing the model's accuracy and effectiveness.

Perceptron Learning Rule

- This iterative learning process laid the foundation for more advanced learning algorithms in neural networks.
- The perceptron learning rule is a supervised learning algorithm used to train the perceptron.

Perceptron Learning Rule

1. Initialization: Start with random weights w_j and a bias term b.

2. Forward Pass: For each training example $\mathbf{x} = [x_1, x_2, \dots, x_d]^{\mathrm{T}}$ with label $y \in \{0,1\}$, compute the predicted output $\hat{y}$ as follows:

$$z = b + \sum_{j=1}^{d} w_j\, x_j \quad \text{and} \quad \hat{y} = \begin{cases} 0, & \text{for } z < 0 \\ 1, & \text{for } z \geq 0 \end{cases}$$

3. Error Calculation: Calculate the error as the difference between the true label y and the predicted label $\hat{y}$:

$$\text{error} = y - \hat{y}$$

4. Weight Update: Update the weights and bias based on the error:

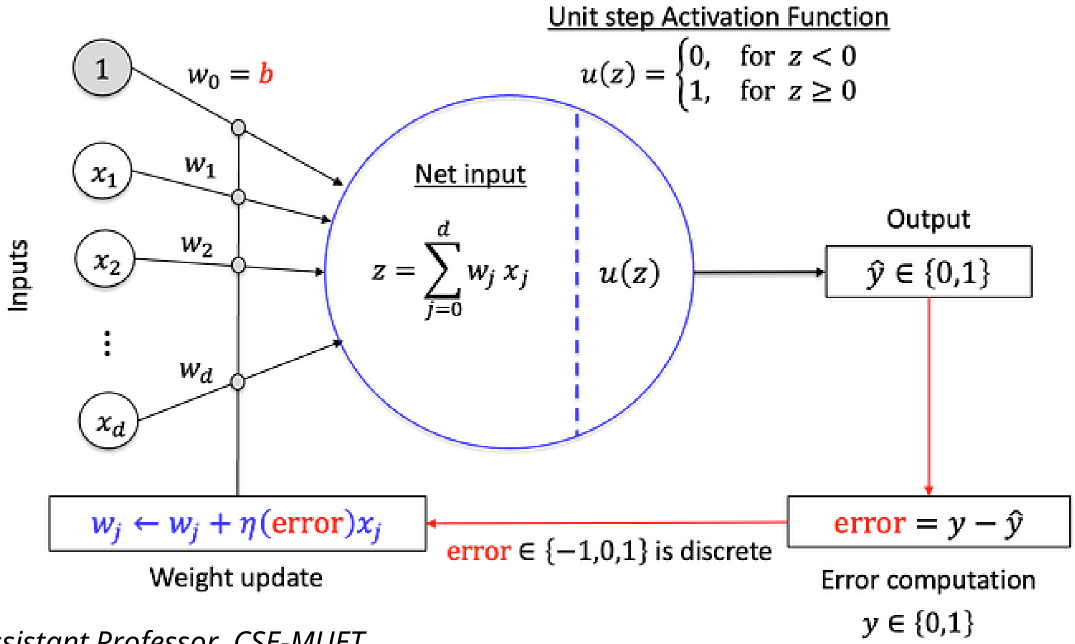
$$w_j = w_j + \eta \cdot \text{error} \cdot x_j$$

$$b = b + \eta \cdot \text{error}$$

5. Iteration: Repeat steps 2–4 for a fixed number of iterations or until the weights converge.

Perceptron Learning Rule

- where η is the learning rate, a small positive constant between 0 and 1 that controls the step size of the weight updates.



Limitations of the Perceptron Learning Rule

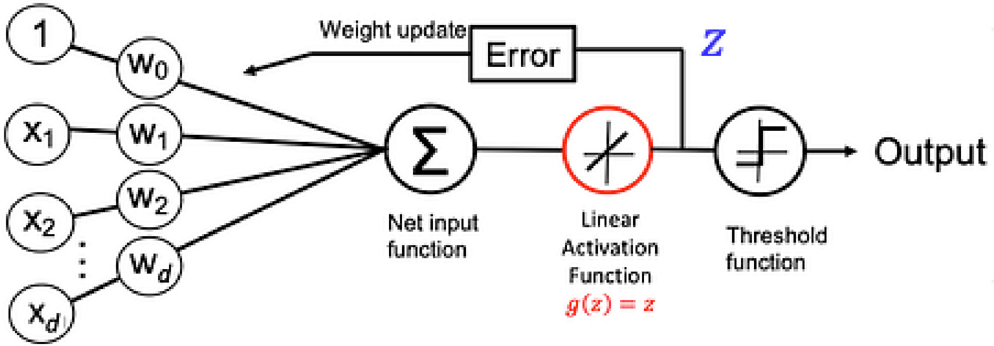
- **Linear Decision Boundaries:** The perceptron is limited to problems where the classes can be separated by a straight line.
- **Convergence Issues:** The perceptron learning rule may not converge if the data is not linearly separable.
- **Non-differentiable of the Perceptron:** In the perceptron, the net input z is computed and passed through a threshold function to produce the output, which is the predicted class label (0 or 1). If the prediction matches the actual label, no action is taken. If it doesn't, an error term is computed, and the weights are updated to adjust the decision boundary. However, the threshold function is not differentiable, preventing the use of calculus for optimization.

ADALINE and the Delta Rule

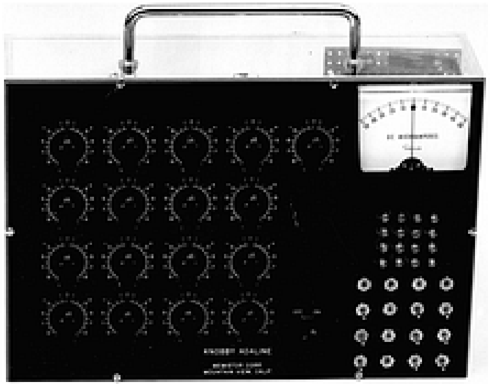
- To address the limitations of the perceptron learning algorithm, Bernard Widrow and Marcian Hoff introduced ADALINE (Adaptive Linear Neuron) in 1960.
- ADALINE uses a different learning rule, known as the delta learning rule or the least mean squares (LMS) algorithm.
- The ADALINE was originally a physical device but can now be implemented in software. It is a nicely differentiable neural model.

ADALINE and the Delta Rule

$$\text{error} = y - g(\mathbf{w}^{\mathrm{T}}\mathbf{x}) = y - \mathbf{w}^{\mathrm{T}}\mathbf{x} = y - z$$



ADALINE Device



ADALINE and the Delta Rule

- Here’s how it works:
- **Net Input Calculation:** Similar to the perceptron, the ADALINE computes the net input.
- **Activation Function:** Initially, the activation function is a linear activation (identity) function , which doesn’t alter the net input. This is a placeholder for future nonlinear activation functions.
- **Threshold Function:** The ADALINE adds a threshold function to turn the linear regression model into a classifier, similar to the perceptron.

ADALINE and the Delta Rule

1. Initialization: Start with random weights w_j and a bias term b.

2. Forward Pass: For each training example $\mathbf{x} = [x_1, x_2, \ldots, x_d]^{\mathrm{T}}$ with label $y \in \{0,1\}$, compute the predicted output $\hat{y}$ as follows:

$$z = b + \sum_{j=1}^{d} w_j \, x_j \quad \text{and} \quad \hat{y} = \begin{cases} 0, & \text{for } z < 0 \\ 1, & \text{for } z \geq 0 \end{cases}$$

3. Error Calculation: Calculate the error as the difference between the true label y and the net input z :

$$\text{error} = y - z$$

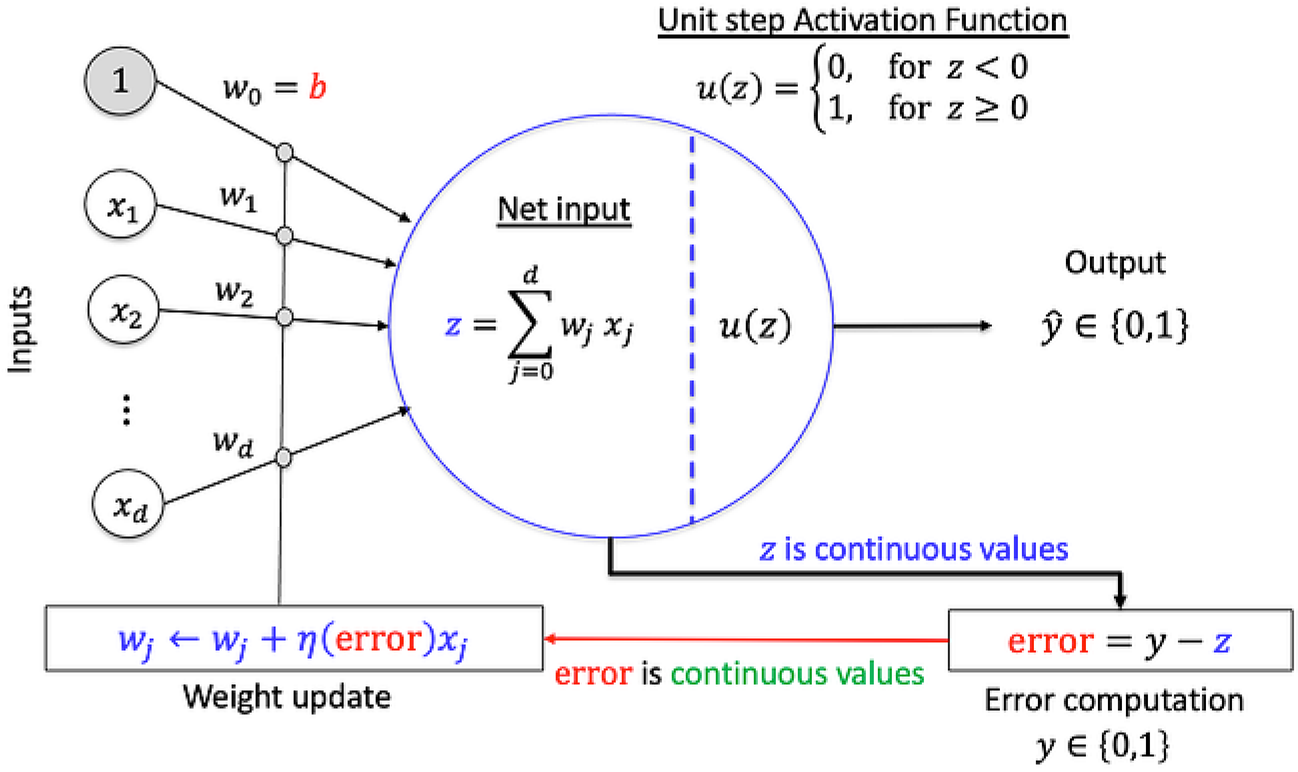
4. Weight Update: Update the weights and bias based on the error:

$$w_j = w_j + \eta \cdot \text{error} \cdot x_j$$

$$b = b + \eta \cdot \text{error}$$

5. Iteration: Repeat steps 2–4 for a fixed number of iterations or until the weights converge.

ADALINE and the Delta Rule



Key Difference: Perceptron & ADALINE

- The main difference between the perceptron and the ADALINE is where the error is computed:
- **Perceptron:** Computes the error after the threshold function.
- **ADALINE:** Computes the error before the threshold function.
- This allows the ADALINE to avoid the non-differentiability issue of the threshold function, as the gradients are computed before applying the threshold.

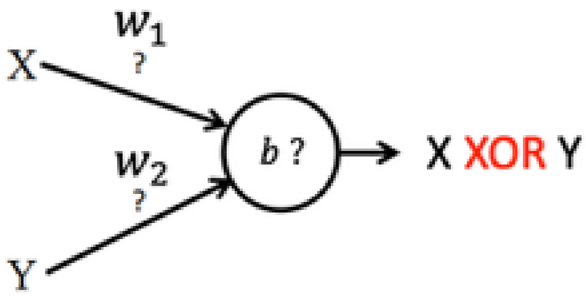
Advantages of ADALINE

- **Convergence:** The delta learning rule is guaranteed to converge to the optimal solution if the learning rate is sufficiently small.
- **Handling Non-Linearly Separable Data:** Although ADALINE still uses a linear decision boundary, its continuous output and MSE-based optimization make it more robust and applicable to a wider range of problems.

Limitations of ADALINE

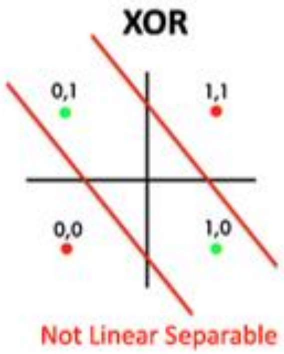
- While groundbreaking, both perceptron and ADALINE were limited to solving linearly separable problems.
- The XOR function takes two binary inputs and returns a binary output.
- The truth table for the XOR function is as follows:

X	Y	X XOR Y
0	0	0
0	1	1
1	0	1
1	1	0



Limitations of ADALINE

1. $b + w_1 \cdot 0 + w_2 \cdot 0 < 0 \Rightarrow b < 0$
2. $b + w_1 \cdot 1 + w_2 \cdot 0 \geq 0 \Rightarrow w_1 \geq -b$
3. $b + w_1 \cdot 0 + w_2 \cdot 1 \geq 0 \Rightarrow w_2 \geq -b$
4. $b + w_1 \cdot 1 + w_2 \cdot 1 < 0 \Rightarrow w_1 + w_2 < -b$



Data Modality in AI

- In artificial intelligence, a **modality** refers to a specific **type of data** input or output that an AI system can process and understand.

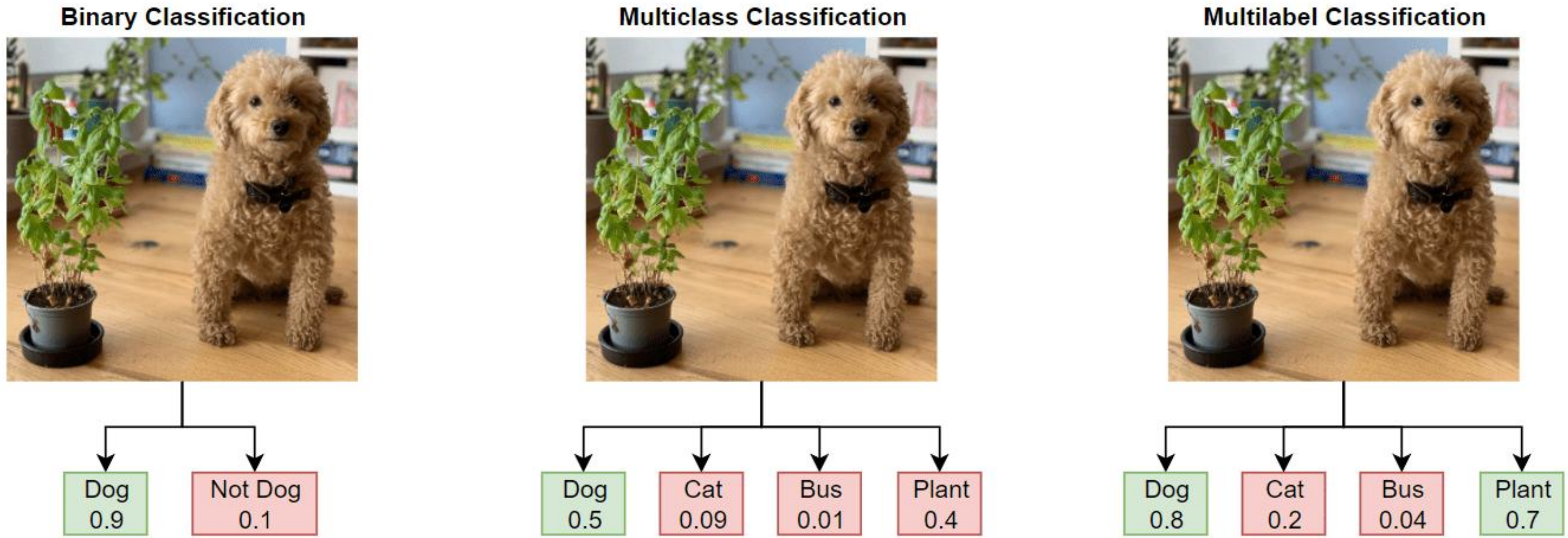
- Visual: Images, videos
- Textual: Text documents, code
- Audio: Speech, music
- Sensor data: Temperature readings, pressure gauges (data from Internet of Things)

Data Modality in AI

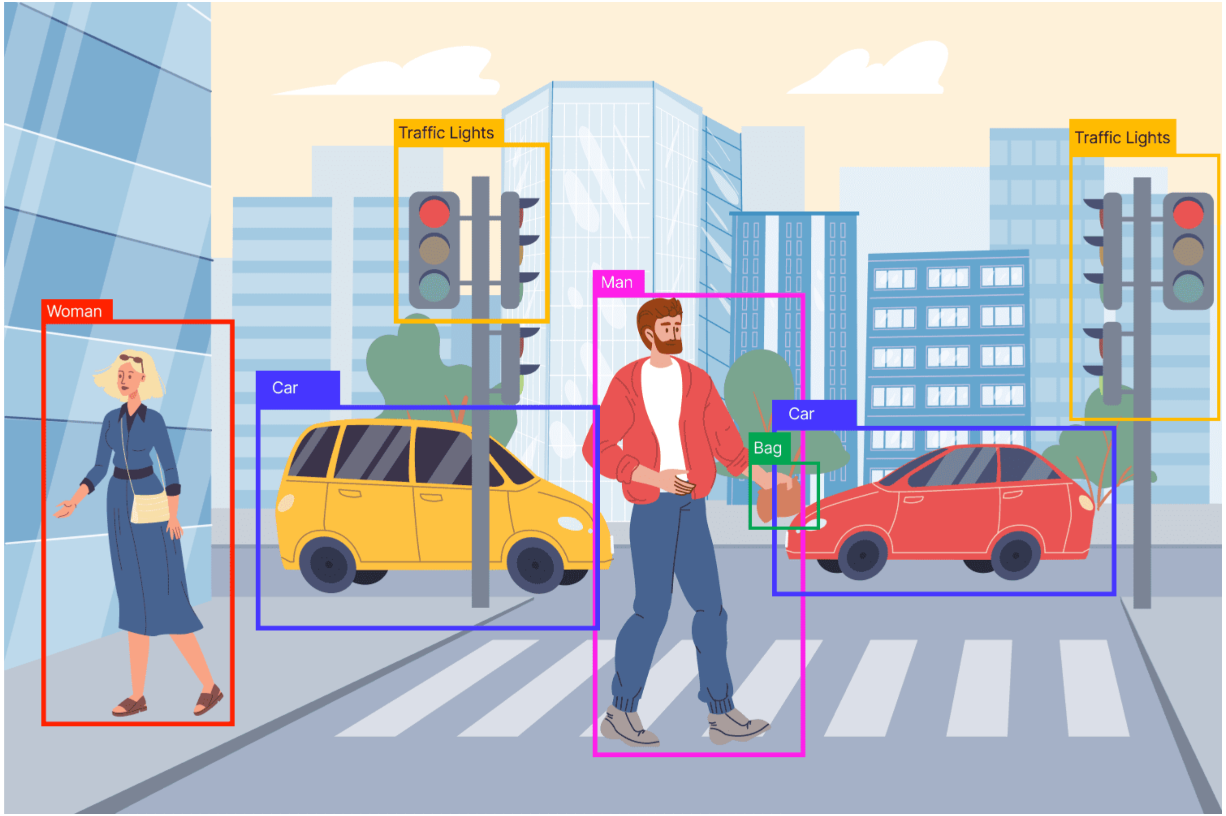
- **Unimodal** AI systems: Specialists in one data type
  - Unimodal AI systems are designed to process and understand data from a single modality.

- **Multimodal** AI systems: Masters of diverse data
  - Multimodal AI systems integrate and process data from multiple modalities simultaneously, enabling more comprehensive understanding and interaction with the environment.

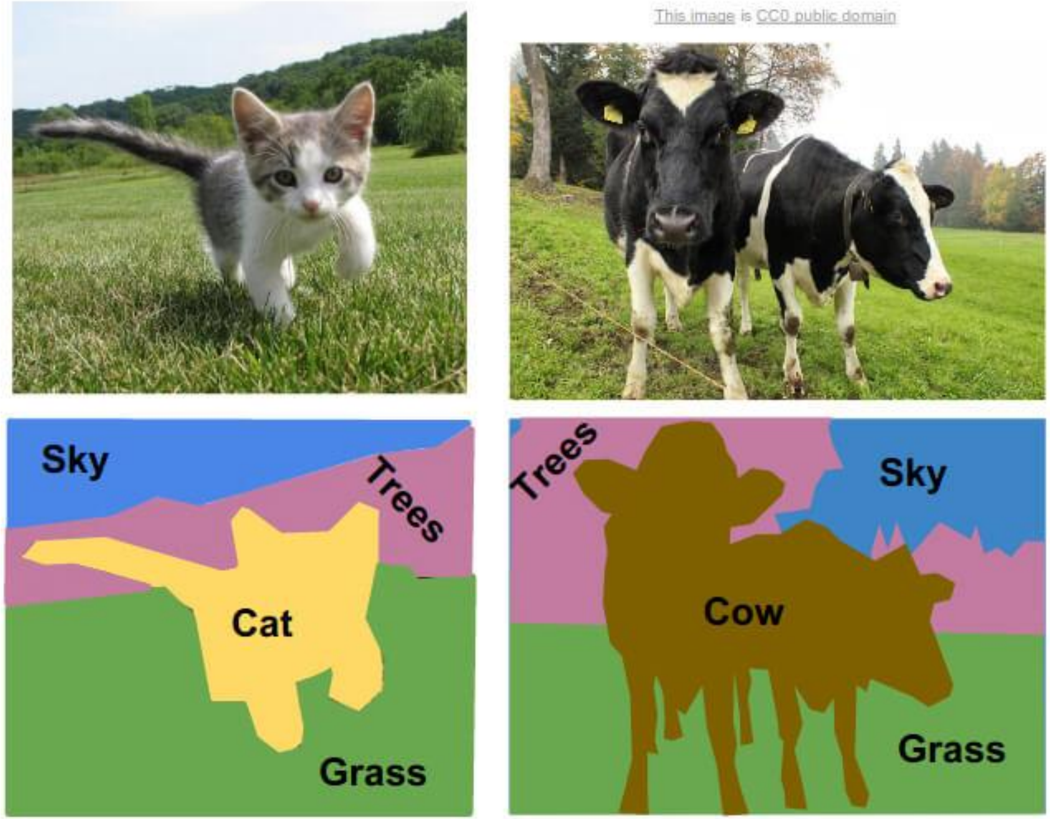
What AI can do? Image Classification



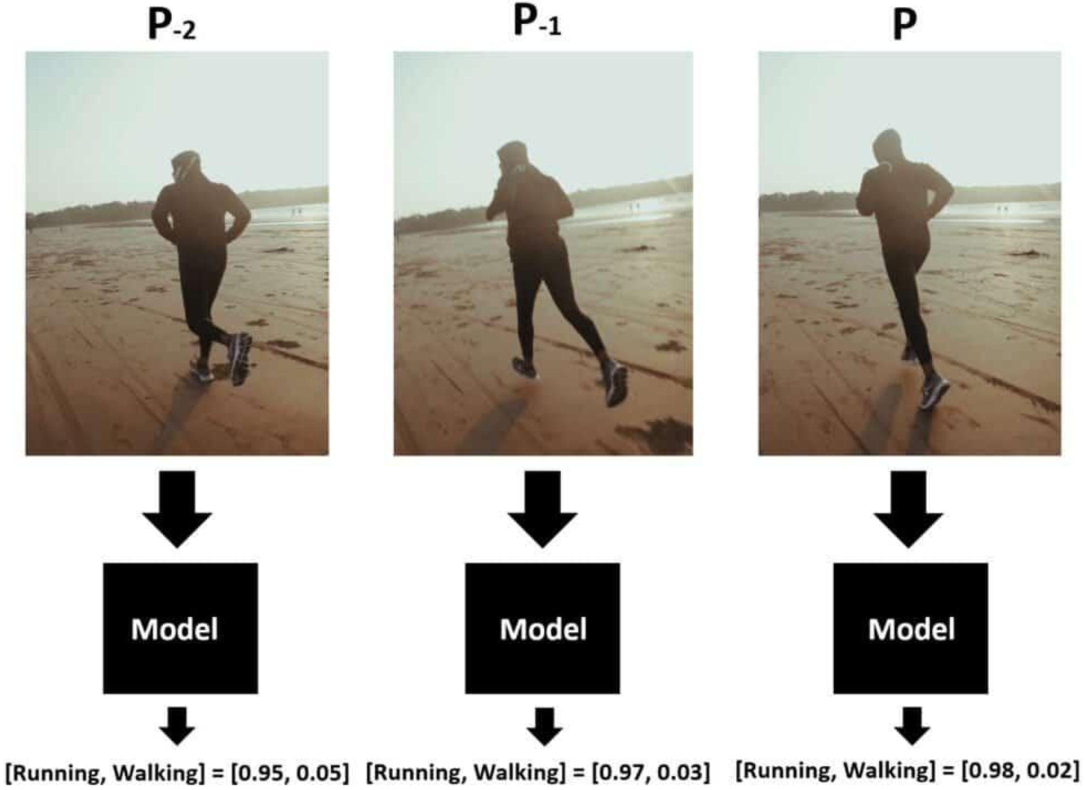
What AI can do? Object Detection



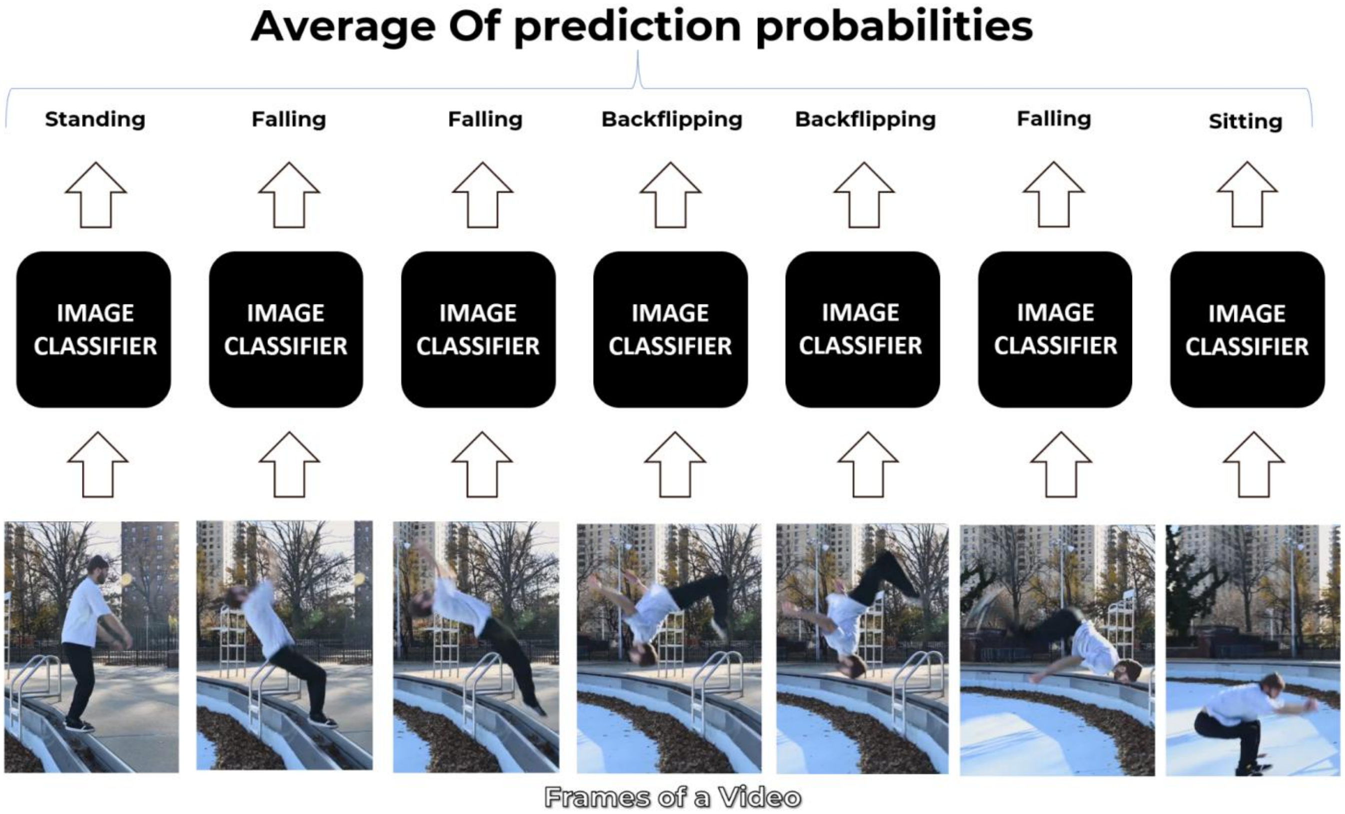
What AI can do? Image Segmentation



What AI can do? Video Classification



What AI can do? Video Classification



What AI can do?

Computer Vision

- Depth Estimation
- Image Classification
- Object Detection
- Image Segmentation
- Text-to-Image
- Image-to-Text
- Image-to-Image
- Image-to-Video
- Unconditional Image Generation
- Video Classification
- Text-to-Video
- Zero-Shot Image Classification
- Mask Generation
- Zero-Shot Object Detection
- Text-to-3D
- Image-to-3D
- Image Feature Extraction
- Keypoint Detection
- Video-to-Video

What AI can do?

Natural Language Processing

- Text Classification
- Token Classification
- Table Question Answering
- Question Answering
- Zero-Shot Classification
- Translation
- Summarization
- Feature Extraction
- Text Generation
- Fill-Mask
- Sentence Similarity
- Text Ranking

What AI can do?

Audio

- Text-to-Speech
- Text-to-Audio
- Automatic Speech Recognition
- Audio-to-Audio
- Audio Classification
- Voice Activity Detection

What AI can do?

Multimodal

- Audio-Text-to-Text
- Image-Text-to-Text
- Visual Question Answering
- Document Question Answering
- Video-Text-to-Text
- Visual Document Retrieval
- Any-to-Any