

UNIT 1

Introduction

(6 Hours) — Artificial Intelligence (CACS410)

⚠ Please Note

These notes were prepared for quick revision and **may contain errors or omissions**. Please take care: cross-check important definitions, formulas, algorithms and worked examples against your prescribed textbook, class notes and teacher's instructions before relying on them (especially for exams).

1.1 Intelligence and Artificial Intelligence

Intelligence and Intelligent Behavior

Intelligence is the ability to learn from experience, understand and apply knowledge, reason, solve problems, adapt to new situations and make decisions. **Intelligent behavior** is behavior that shows these abilities: the entity perceives its surroundings, chooses actions that move it toward a goal, and improves with experience.

Aspect of intelligent behavior	Meaning	Example
Perception	Sensing and interpreting the environment	Recognizing a face in a photo
Learning	Improving performance from experience	A spam filter getting more accurate
Reasoning	Drawing conclusions from known facts	If it rains, the road is wet; it rains, so the road is wet
Problem solving	Finding a sequence of steps to reach a goal	Finding a route between two cities
Language use	Understanding and generating language	Machine translation
Adaptation	Coping with new or changing situations	A robot re-planning around an obstacle

Definition of Artificial Intelligence

Artificial Intelligence (AI) is the branch of computer science concerned with building machines and software that perform tasks which normally require human intelligence. The term was coined by **John McCarthy** for the 1956 **Dartmouth Conference**, which is regarded as the birth of AI as a field. A popular working description (Elaine Rich) is that AI studies how to make computers do things at which, at the moment, people are better.

Understanding AI by Thought Process and Behavior

AI can be defined along two dimensions: whether a system is judged on *thought processes* or on *behavior*, and whether the yardstick is *human performance* or an ideal concept of *rationality*. This gives four approaches.

	Human-like	Rational (ideal)
Thinking	Thinking humanly — cognitive modelling: make programs that think the way the human mind does (cognitive science, GPS).	Thinking rationally — the “laws of thought” approach: correct reasoning using logic (Aristotle's syllogisms).
Acting	Acting humanly — the Turing Test approach: behave so that a human cannot tell you from a person.	Acting rationally — the rational agent approach: act to achieve the best (expected) outcome.

Modern AI mostly follows the **rational agent** approach because it is more general than the “laws of thought” approach and more mathematically precise than imitating humans.

The Turing Test

Proposed by **Alan Turing** in 1950 (*Computing Machinery and Intelligence*). A human interrogator types questions to two hidden parties, one human and one computer. If the interrogator cannot reliably tell which is the machine, the machine is said to pass. To pass, a computer needs: **natural language processing**, **knowledge representation**, **automated reasoning** and **machine learning**. The *Total Turing Test* adds **computer vision** and **robotics**.

Strong (Hard) AI vs. Weak (Soft) AI

Point	Strong / Hard AI	Weak / Soft AI
Claim	A properly programmed computer can really think, understand and have a mind.	A machine can only act as if it were intelligent (simulate intelligence).
Scope	General intelligence (AGI) across many tasks	Narrow intelligence for specific tasks (narrow AI)
Status	Hypothetical; does not yet exist	Exists everywhere today

Point	Strong / Hard AI	Weak / Soft AI
Examples	Human-level reasoning, consciousness (science fiction so far)	Chess programs, voice assistants, recommendation systems, spam filters
Debate	Opposed by Searle's Chinese Room argument	Widely accepted as a practical engineering goal

*Note: John Searle's **Chinese Room** argument says that following symbol-manipulation rules (what a program does) is not the same as understanding, so a program that behaves intelligently need not actually think.*

1.2 Foundations of AI

AI draws on ideas from many older disciplines.

Discipline	Contribution to AI
Philosophy	Logic and reasoning, mind–body question, rationality, source of knowledge (Aristotle, Descartes)
Mathematics	Formal logic, computation and algorithms, probability and statistics (Boole, Gödel, Turing, Bayes)
Economics	Utility theory, decision theory and game theory for making rational choices
Neuroscience	How the brain's neurons process information; inspiration for neural networks
Psychology	How humans think, learn and perceive; cognitive models
Computer engineering	Fast hardware and software tools needed to run AI programs
Control theory & cybernetics	Feedback systems that regulate themselves to reach goals
Linguistics	Structure of language; basis of natural language processing and knowledge representation

A Brief History of AI

Year	Event
1943	McCulloch and Pitts propose a mathematical model of the artificial neuron
1950	Turing publishes “Computing Machinery and Intelligence” (Turing Test)
1956	Dartmouth Conference; the term “Artificial Intelligence” is adopted
1957	Rosenblatt's Perceptron
1966	ELIZA, an early chatbot

Year	Event
1970s–80s	Expert systems (DENDRAL, MYCIN, XCON); “AI winters” followed periods of unmet expectations
1997	IBM Deep Blue defeats world chess champion Garry Kasparov
2012 →	Deep learning breakthroughs in image recognition; later AlphaGo (2016) and large language models

1.3 Applications of AI

Domain	Typical applications
Healthcare	Disease diagnosis, medical image analysis, drug discovery
Finance & business	Fraud detection, credit scoring, algorithmic trading, customer chatbots
Transportation	Self-driving vehicles, route planning, traffic prediction
Language (NLP)	Machine translation, speech recognition, voice assistants, sentiment analysis
Vision	Face recognition, OCR, quality inspection, surveillance
Education	Intelligent tutoring systems, automatic grading, personalised learning
Robotics & industry	Assembly-line robots, warehouse automation, planetary rovers
Games	Chess, Go and video-game opponents
E-commerce & media	Product and movie recommendation
Agriculture	Crop monitoring, yield and disease prediction
Security	Intrusion detection, spam and malware filtering
Expert systems	Decision support in medicine, geology, engineering

1.4 Intelligent Agents

Introduction to Agents

An **agent** is anything that **perceives** its environment through **sensors** and **acts** upon it through **actuators**. A human agent has eyes and ears as sensors and hands and legs as actuators; a robot has cameras and motors; a software agent receives keystrokes or network packets and acts by displaying output or sending messages.

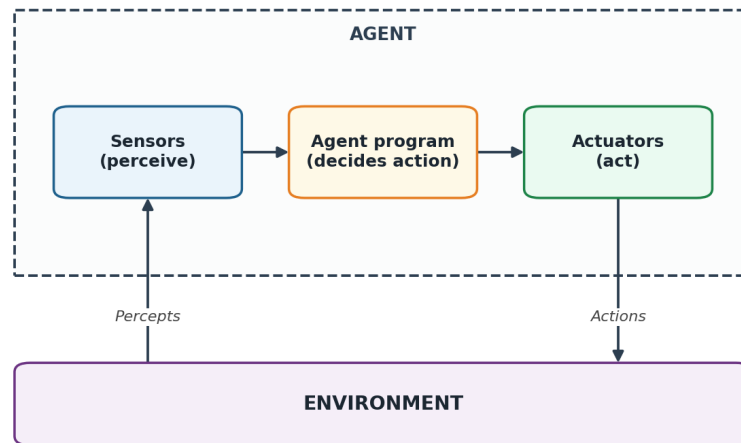


Figure 1.1 — An agent interacting with its environment

- **Percept**: the agent's input at one instant. **Percept sequence**: the complete history of percepts.
- **Agent function**: a mapping from percept sequences to actions. **Agent program**: its concrete implementation.
- A **rational agent** chooses, for every percept sequence, the action expected to maximize its **performance measure**, given the evidence from percepts and its built-in knowledge.

Structure of an Intelligent Agent

Agent = Architecture + Program. The architecture is the computing device with its sensors and actuators; the program implements the agent function. The agent program takes the current percept and returns an action.

Properties of Intelligent Agents

Property	Meaning
Autonomy	Operates without constant human guidance and controls its own actions
Reactivity	Perceives the environment and responds in time to changes
Pro-activeness	Takes the initiative and acts to achieve goals, not just in response to events
Social ability	Communicates and cooperates with other agents or humans
Rationality	Acts to maximize expected performance
Learning / adaptivity	Improves its behavior with experience
Mobility	Ability of some software agents to move across a network

PEAS Description

To design an agent we first specify its task environment as **PEAS**: **P**erformance measure, **E**nvironment, **A**ctuators and **S**ensors.

Agent	Performance	Environment	Actuators	Sensors
Automated taxi driver	Safe, fast, legal, comfortable trip; profit	Roads, traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn	Cameras, GPS, speedometer, sonar
Medical diagnosis system	Healthy patient, low cost	Patient, hospital, staff	Display of questions, tests, diagnoses	Keyboard entry of symptoms and findings
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts, bins	Jointed arm and hand	Camera, joint-angle sensors
Vacuum cleaner	Cleanliness, time, energy used	Rooms, dirt, furniture	Wheels, brushes, suction	Dirt, bump and cliff sensors

Types of Agents

(a) Simple Reflex Agent

Selects actions on the basis of the **current percept only**, using condition–action (IF–THEN) rules. It has no memory, so it works properly only when the environment is fully observable. Example: a thermostat, or “if dirty then suck”.

```
function SIMPLE-REFLEX-AGENT(percept)
  state <- INTERPRET-INPUT(percept)
  rule <- RULE-MATCH(state, rules)
  return rule.ACTION
```

(b) Model-Based Reflex Agent

Maintains an **internal state** — a model of how the world evolves and how its own actions affect it — so it can handle **partially observable** environments. It updates the state from the latest percept and action, then applies rules. Example: a car that remembers a vehicle in its blind spot.

(c) Goal-Based Agent

Has explicit **goals**. It considers “what will happen if I do action A?” and uses **search and planning** to find action sequences that reach the goal. More flexible than reflex agents because the behavior changes when the goal changes. Example: a GPS navigator finding a route to a destination.

(d) Utility-Based Agent

Goals give only a yes/no distinction (goal achieved or not). A **utility function** maps each state to a real number measuring how desirable it is, so the agent can choose the action with **maximum expected utility**. It handles conflicting goals (speed vs. safety) and uncertainty. Example: a taxi picking the route that best balances time, cost and safety.

(e) Learning Agent

Can start with little knowledge and improve through experience. It has four conceptual components: the **performance element** (selects external actions), the **critic** (judges how well the agent is doing against a performance standard), the **learning element** (makes improvements using the critic's feedback) and the **problem generator** (suggests exploratory actions that lead to new experiences).

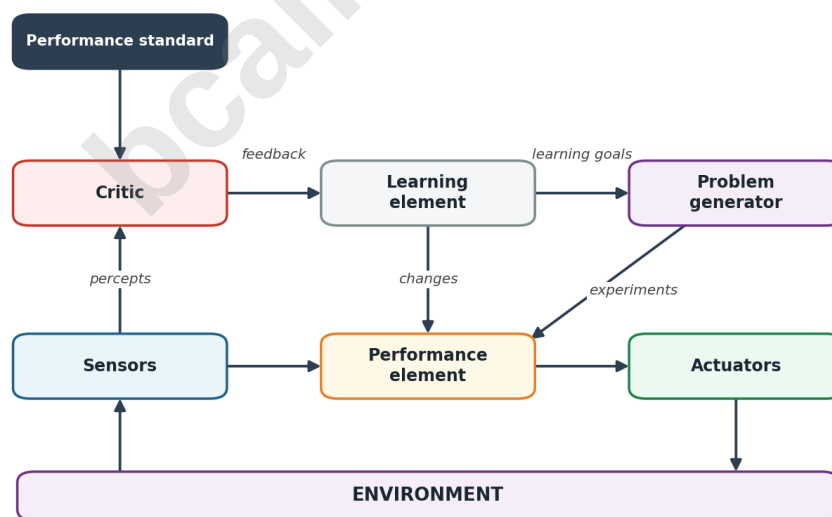


Figure 1.2 — General structure of a learning agent

Agent type	Uses	Key idea	Limitation
Simple reflex	Current percept	Condition–action rules	Needs fully observable environment
Model-based	Percept + internal state	Keeps a world model	No notion of goals

Agent type	Uses	Key idea	Limitation
Goal-based	Model + goals	Search / planning	Cannot compare how good different goal paths are
Utility-based	Model + utility function	Maximize expected utility	Designing utility is hard
Learning	Experience	Critic + learning element	Needs feedback and time

Environment Types

Property	Meaning
Fully observable vs. Semi (partially) observable	Sensors give the complete relevant state of the environment at each time vs. only part of it (noisy or missing data)
Deterministic vs. Stochastic	The next state is completely determined by the current state and action vs. involves randomness or uncertainty
Static vs. Dynamic	The environment does not change while the agent is thinking vs. it can change meanwhile (the agent must keep up)
Single-agent vs. Multi-agent	Only one agent operates in the environment vs. several agents (cooperative or competitive) interact

Other useful distinctions are **episodic vs. sequential** (do current decisions affect future ones?), **discrete vs. continuous** (states, time, actions) and **known vs. unknown** (does the agent know the “laws” of the environment?).

Task	Observable	Deterministic	Static / Dynamic	Agents
Crossword puzzle	Fully	Deterministic	Static	Single
Chess with a clock	Fully	Deterministic (strategic)	Semi-dynamic	Multi
Poker	Partially	Stochastic	Static	Multi
Taxi driving	Partially	Stochastic	Dynamic	Multi
Medical diagnosis	Partially	Stochastic	Dynamic	Single
Image analysis	Fully	Deterministic	Semi-dynamic	Single

*Note: The hardest case is **partially observable, stochastic, dynamic, multi-agent** (e.g., real-world driving).*

Key Takeaways

- ✓ Intelligence = learning, reasoning, perception, problem solving and adaptation; AI builds machines that show it.
- ✓ Four approaches: thinking/acting humanly, thinking/acting rationally; modern AI prefers the **rational agent** view.
- ✓ Strong AI claims machines can really think; Weak AI only simulates intelligence. Today's systems are weak (narrow) AI.
- ✓ Agent = sensors + program + actuators; describe a task using **PEAS**.
- ✓ Agent types: simple reflex, model-based, goal-based, utility-based, learning (in increasing power).
- ✓ Environments: observable/semi-observable, deterministic/stochastic, static/dynamic, single/multi-agent.

Review Questions

- Define AI. Explain the four approaches to AI with examples.
- Explain the Turing Test. What capabilities must a machine have to pass it?
- Differentiate between strong AI and weak AI.
- Give the PEAS description of an automated taxi driver and a medical diagnosis system.
- Explain the five types of agents with diagrams. How does a learning agent differ from the others?
- Explain the different types of environments with examples.

— *End of Unit 1* —

Reminder: these notes may contain errors — verify with your textbook and teacher.