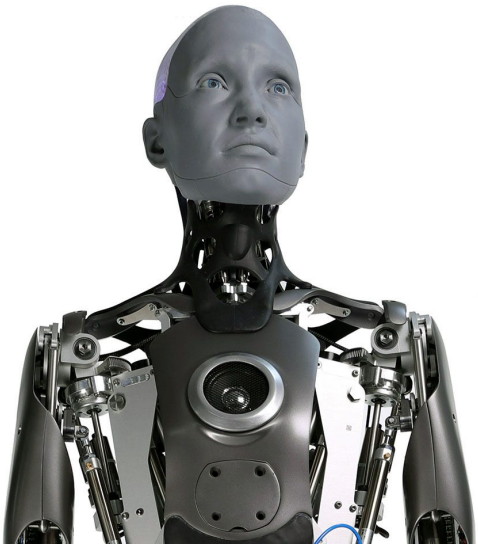
Artificial Intelligence (AI)

A-Level Computer Science

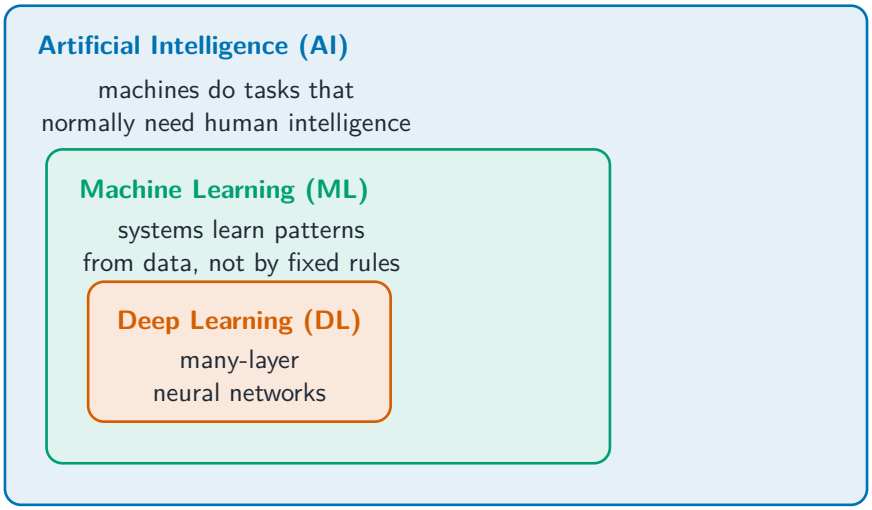
What AI is

Artificial intelligence 人工智能 (AI) builds systems that do tasks normally needing human intelligence —recognising speech and images, translating, playing games, driving, generating text. Most modern AI uses **machine learning** 机器学习—algorithms that learn patterns from data instead of being programmed step by step. Within it, **deep learning** 深度学习, using **neural networks** 神经网络 with many layers, has been dominant since the 2010s.

A **humanoid robot** 人形机器人 puts many of these abilities into one body: it uses AI to see faces, understand speech and move its face and arms in a lifelike way.



A humanoid robot uses AI to see, listen and respond like a person

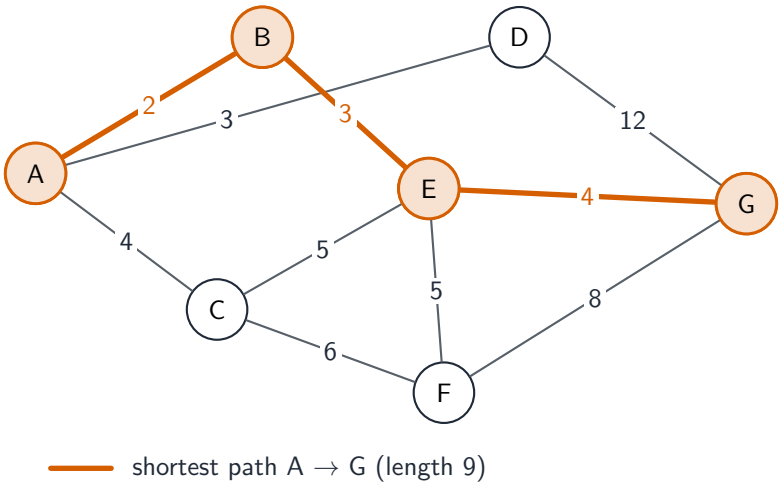


Deep learning is part of machine learning, which is part of AI

Graphs in AI

Many AI problems sit on a **graph** 图—**nodes** 节点 (states, places) joined by **edges** 边 (moves, relationships).

- **pathfinding**: roads form a graph; the shortest route is a graph search (Dijkstra, A*).
- **game playing**: each board position is a node, each move an edge; **minimax** 极小化极大 with alpha-beta pruning searches the game tree.
- **state-space search**: a planning problem is moving between states by applying operators to reach a goal.
- **knowledge representation**: a **semantic network** 语义网络 has concepts as nodes and relationships as edges ("dog IS-A animal"); a **knowledge graph** 知识图谱 stores facts about the world for search engines and assistants.



AI problems often sit on a graph; here the shortest path is highlighted

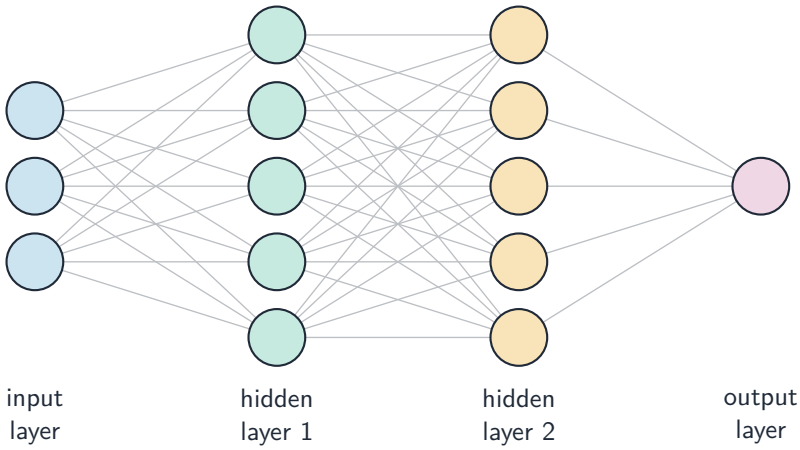
Standard tools for navigating graphs include **breadth-first search** 广度优先搜索 and **depth-first search** 深度优先搜索.

Artificial neural networks (ANNs)

An ANN is inspired by the brain's neurons. An **artificial neuron** 人工神经元:

- takes several **input values**, multiplies each by a **weight** 权重, and adds them up with a **bias term** 偏置项.
- applies an **activation function** 激活函数 (a non-linear function such as ReLU) to the sum.
- outputs the result, which feeds neurons further on.

Neurons sit in layers: an **input layer**, one or more **hidden layers** 隐藏层 (where useful internal patterns are learned), and an **output layer**. With many hidden layers it is a **deep neural network** 深度神经网络, and training it is deep learning.



A neural network with an input layer, two hidden layers and an output layer

ANNs let models learn complex patterns straight from raw data (pixels, audio, text) without hand-designed features —driving breakthroughs in **image recognition** 图像识别, **speech recognition** 语音识别, **machine translation** 机器翻译, and game playing. They do well with large data, noisy/high-dimensional input, and patterns too complex for explicit rules.

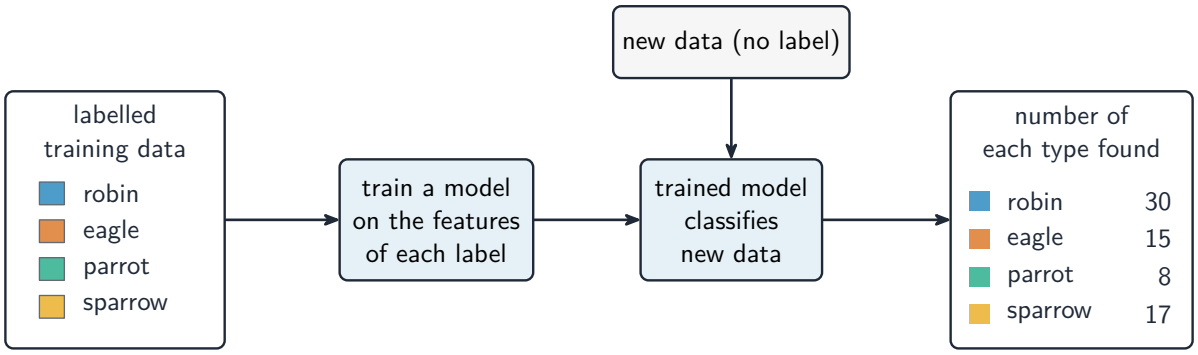
Machine learning, deep learning, reinforcement learning

Machine learning

The umbrella term —any algorithm that learns from data. Three paradigms:

- **supervised learning** 监督学习—the data has **labels** 标签 (images tagged "cat"/"dog"); the algorithm learns input → label. Used for **classification** 分类 (a category) and regression.
- **unsupervised learning** 无监督学习—no labels; the algorithm finds structure, e.g. a **cluster** 聚类 of similar customers.
- reinforcement learning (below).

Use ML when explicit rules would be impractical (spam filters, recommendations, fraud detection).



Supervised learning: a model is trained on labelled data, then recognises new data

Deep learning

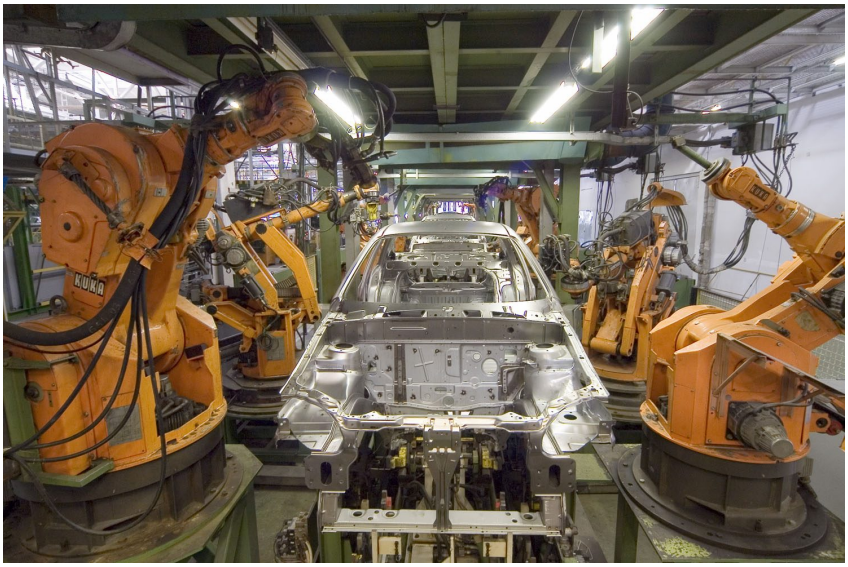
A subset of ML using deep neural networks. Lower layers learn simple patterns (edges, phonemes), higher layers combine them into abstract concepts. It needs **lots of data** and **lots of compute** (GPUs); for small datasets, simpler ML methods often do better.

Reinforcement learning

In **reinforcement learning** 强化学习, an **agent** 智能体 acts in an environment; each action changes the state and returns a **reward** 奖励. The agent learns a **policy** 策略 (a strategy) that maximises the total reward over time, by **trial and error** with no labels up front. Used for sequential-decision problems —games, robot control, autonomous driving.

A **self-driving car** 自动驾驶汽车 is a real example. **Lidar** 激光雷达 and camera sensors (the spinning unit on the roof) build a live picture of the road, and a learned policy decides how to steer, speed up and brake safely.

A self-driving car uses cameras and lidar sensors to see the road around it



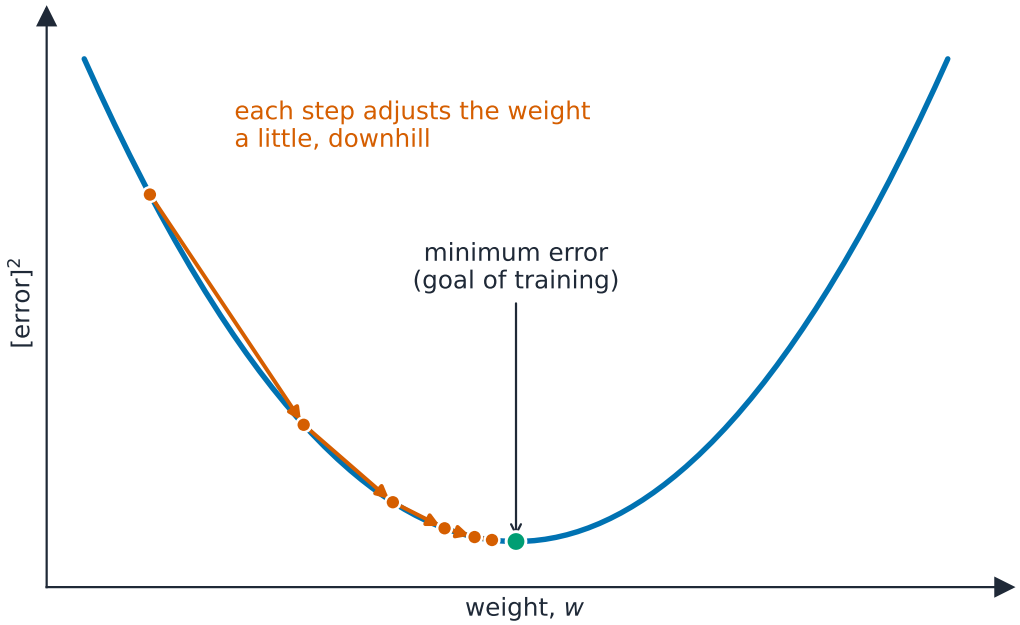
Industrial robot arms on a production line: reinforcement learning can teach a robot to control its movements

Training an ANN: backpropagation

Training adjusts the **weights** so outputs match the targets. The standard method is **backpropagation** 反向传播 with **gradient descent** 梯度下降. For each training example:

1. **forward pass** —feed the input through to the output.
2. **compute the error** with a **loss function** 损失函数 (a single number for how wrong the output is).
3. **backward pass** —propagate the error **backwards**, finding each weight's **gradient** (how much it contributed to the error) using the chain rule.
4. **update the weights** by a small step (set by the **learning rate** 学习率) that reduces the error.

Repeat over many examples and many passes (**epochs** 训练轮次) until the error stops shrinking. After training, a new input needs only one forward pass to get a prediction — which is why "back": the error flows from the output back to the input, so all gradients are found in one efficient backward sweep.



Training adjusts the weights to reach the minimum error

Regression

Some tasks predict a **number** (a house price, tomorrow's temperature) —**regression** 回归, as opposed to classification (a category).

Linear regression 线性回归 fits a straight line (or hyperplane):

$$y = m_1x_1 + m_2x_2 + \ldots + m_nx_n + c.$$

Choose the coefficients to **minimise the sum of squared errors** against the training data. Use it when the relationship looks roughly linear and you want an interpretable model. For curved data, use polynomial, decision-tree, or neural-network regression —same idea: define a model, define a loss, and adjust the parameters to minimise it. Regression and classification are both supervised; the choice depends on whether the answer is a number or a category.

How AI is used in a real scenario

Many exam scenarios use the same pattern —a **deep-learning model trained on labelled data**, often several combined into a pipeline:

- **customer identification** at an automated shop: the system is trained on labelled face images; a camera captures a face; image recognition extracts a representation; it is matched against registered customers; the closest match identifies the person.
- **reading text from images**: image recognition finds text regions; **optical character recognition** 光学字符识别 extracts the characters; machine translation converts them; **text-to-speech** 文本转语音 reads them aloud.
- **checkout item-detection**: object-detection AI, trained on labelled product images, sees which items go into a basket and charges the account.

By the time a user interacts with the system, the model is fast —it only does forward-pass inference; the intelligence is in the patterns learned during training.