A-LEVEL COMPUTER SCIENCE • EXERCISE SHEET

18.1 Artificial Intelligence (AI)

Name: _____ Class: _____ Date: _____ **Total: 32 marks**

KEY WORDS artificial intelligence 人工智能 • deep learning 深度学习 • machine learning 机器学习 • neural networks 神经网络 • deep neural network 深度神经网络

Objective

Build the skills to answer exam questions on **Artificial Intelligence** 人工智能 — graphs in AI, neural networks, and the types of machine learning.

You must be able to:

- explain how **graphs** help AI and how an **artificial neural network** works
- compare **supervised**, **unsupervised** and **reinforcement** learning
- explain **deep learning** and **backpropagation**

1 Worked examples

Study these first. Each one shows the method for a question type used later — follow the steps and you can do the Practice and Exam-style questions yourself.

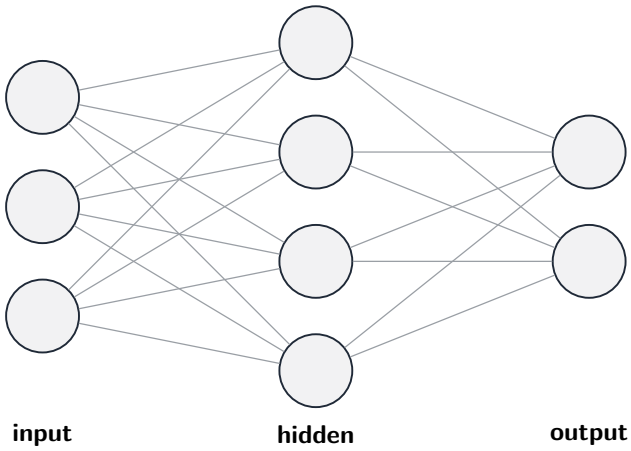
■ AI, ML and graphs

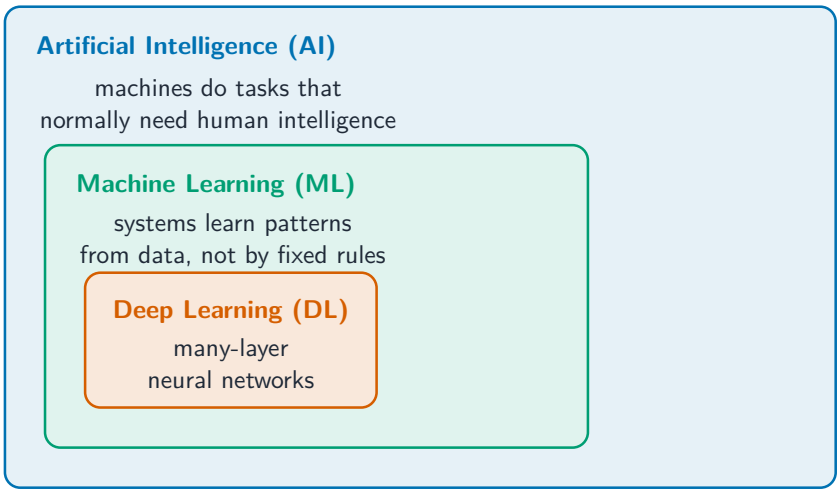
Machine learning (ML) learns patterns **from data** instead of fixed rules; **deep learning (DL)** is ML using deep neural networks. Many AI problems sit on a **graph** — **nodes** (states/places) joined by **edges** (moves, each with a cost). Two algorithms find the **shortest path**:

- **Dijkstra's algorithm** explores outward from the start, always extending the lowest-cost node, until it reaches the goal.
- **A*** does the same but adds a **heuristic** estimate of the distance **still to go**, which steers the search **toward the goal** — so it usually checks **fewer nodes** and is faster.

■ Neural networks

A network has an **input** layer, one or more **hidden** layers, and an **output** layer. Each connection has a **weight**; a neuron sums its weighted inputs (plus a bias), applies an **activation function**, and passes the result on:





How AI, machine learning and deep learning relate

Types of machine learning

- **supervised** — data has **labels**; learns input → label (classification, regression).
- **unsupervised** — **no labels**; finds structure (e.g. **clusters**).
- **reinforcement** — an **agent** acts, gets a **reward**, and learns a **policy** that maximises total reward by trial and error.

Training adjusts the weights by **backpropagation**: a forward pass computes the output, a **loss function** measures the error, and the error is propagated **backwards** to update each weight (gradient descent). Deep learning needs **lots of data and compute**.

2 Practice

Now apply the methods above.

2.1 State what **machine learning** is. [1]

2.2 Name the **three** kinds of layer in a neural network. [2]

2.3 In a graph used for **pathfinding**, state what an **edge** represents. [1]

2.4 State what a **weight** is in an artificial neuron. [1]

3 Exam-style questions

3.1 Describe the difference between **supervised** and **unsupervised** learning. [2]

3.2 Explain how **reinforcement learning** works, using the terms **agent**, **reward** and **policy**. [3]

3.3 Explain why **deep learning** needs a large amount of **data** and **computing power**. [2]

3.4 State the role of **backpropagation** in training a neural network. [2]

3.5 State the purpose of **Dijkstra's algorithm** and **A***, and outline **one** difference

between them. [3]

3.6 A delivery company is building software to plan its routes and to read handwritten address labels.

(a) **Identify** one Artificial Intelligence algorithm suitable for finding the shortest distance between two depots on a road network, and **state** what the nodes and the edge weights represent. [3]

(b) **Describe** how that algorithm proceeds from the starting node. [3]

(c) The company also considers the **A*** algorithm. **State** the one thing A* uses that the algorithm in (a) does not, and **explain** the benefit. [3]

(d) **Describe** what is meant by **Deep Learning**, and **state** one example of where it is used. [3]

(e) **Explain** why Deep Learning suits the handwriting-reading task better than a set of hand-written rules. [3]

Solutions

2.1 Algorithms that **learn patterns from data** (and improve with experience) instead of being programmed with explicit rules.

2.2 Input layer, **hidden** layer(s), **output** layer.

2.3 A **move/connection** between states (e.g. a road between two places), often with a **cost/weight**.

2.4 A value each input is **multiplied by**, setting how strongly that input affects the neuron's output.

3.1 Supervised learning trains on **labelled** data (each input has a known correct output); **unsupervised** learning uses **unlabelled** data and finds structure/clusters itself.

3.2 An **agent** takes an **action** in an environment, which changes the state and returns a **reward**; over many trials it learns a **policy** (strategy) that **maximises total reward** by trial and error.

3.3 Deep networks have **very many weights** to learn, so they need large datasets to learn good patterns and heavy compute (GPUs) to train in reasonable time.

3.4 It propagates the output **error backwards** through the layers to **adjust each weight** so the network's output gets closer to the target (gradient descent).

3.5 Both find the **shortest path** between two nodes in a weighted graph. Difference: **A*** uses a **heuristic** to estimate the remaining distance and head toward the goal, so it usually explores **fewer nodes / is faster**; **Dijkstra's** has no heuristic and explores outward uniformly.

3.6 (a) **Dijkstra's algorithm**. The **nodes** are the depots and road junctions; the **edge weights** are the distances, or the travel times, along each road between them.
(b) It gives the start node a distance of zero and every other node infinity. It then repeatedly takes the **unvisited node with the smallest known distance**, and for each of its neighbours checks whether going via this node is shorter than the distance recorded so far, updating it if so. The node is then marked visited and never reconsidered; the process ends when the destination is visited.
(c) A* also uses a **heuristic** — an estimate of the remaining distance from each node to the goal, such as the straight-line distance. It therefore explores nodes that head **towards** the destination first, so it usually examines far fewer nodes than Dijkstra while still finding the shortest route, provided the heuristic never overestimates.
(d) Deep Learning uses an **artificial neural network with many hidden layers**, trained on large amounts of data, where each layer learns features built from the layer before it. Examples: speech recognition, image and face recognition, machine translation, self-driving vehicles.
(e) Handwriting varies enormously between people, so the rules that distinguish a `7` from a `1` cannot be written out completely. A network **learns** the distinguishing features from thousands of labelled examples instead of being told them, and it generalises to handwriting it has never seen, which a fixed rule set cannot do.