Solutions

Each answer shows the steps, then the **result**.

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
- **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