

Artificial Intelligence

A-Level Computer Science 9618 · Topic 18 revision notes

Everything from the 3 lessons of this topic on a few pages: the key idea of each lesson, the diagram to remember, the points to learn and the traps examiners set. Read it, cover it, test yourself.

1. Graphs in AI: Dijkstra's and A* algorithms
2. Machine learning: supervised, unsupervised and reinforcement
3. Neural networks, deep learning and back propagation

thelucidstem.com

1 Graphs in AI: Dijkstra's and A* algorithms

Syllabus 18.1

A **graph** is a set of **nodes** (vertices) joined by **edges** (arcs). Edges can carry **weights** (a cost such as distance or time) and can be **directed** or undirected. AI uses graphs to model states and the moves between them, then searches them. **Dijkstra's algorithm** always visits the unvisited node with the **smallest distance from the start**, so it finds the shortest path to every node. **A*** chooses the node with the smallest $f = g + h$, where g is the cost so far and h is a **heuristic** estimate of the cost still to go, so it usually reaches the goal after visiting far fewer nodes.

A weighted, undirected graph (weights are distances in km)

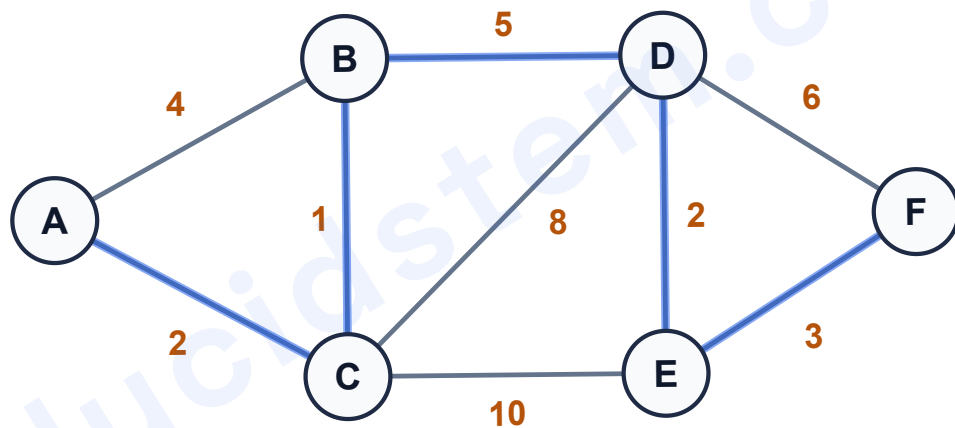


Fig. 1: a weighted graph; the highlighted route A-C-B-D-E-F is the shortest path from A to F (13 km)

Worked example: Trace on Fig. 1

Use Dijkstra's algorithm on the graph in Fig. 1 to find the shortest distance and route from A to F.

1. Start: $A = 0$, all others ∞ . Visit A: B becomes $0 + 4 = 4$ and C becomes $0 + 2 = 2$.
2. Smallest unvisited is C (2). Via C: $B = 2 + 1 = 3$ (better than 4, so B's previous node becomes C), $D = 2 + 8 = 10$, $E = 2 + 10 = 12$.
3. Smallest unvisited is B (3). Via B: $D = 3 + 5 = 8$ (better than 10).
4. Visit D (8). Via D: $E = 8 + 2 = 10$ (better than 12), $F = 8 + 6 = 14$.
5. Visit E (10). Via E: $F = 10 + 3 = 13$ (better than 14). Finally visit F (13).

6. Visit order A, C, B, D, E, F. Follow previous nodes back from F: $F \leftarrow E \leftarrow D \leftarrow B \leftarrow C \leftarrow A$.

Answer: Shortest distance 13 km, route $A \rightarrow C \rightarrow B \rightarrow D \rightarrow E \rightarrow F$

Smallest first.

- ✓ **Dijkstra:** visit the unvisited node with the smallest distance from the start; update neighbours if current + weight is smaller.
- ✓ **A*:** $f = g + h$; expand the open node with the smallest f ; stop when the goal is chosen.
- ✓ **Record previous nodes:** then read the route backwards from the goal.
- ✓ **Show every update:** a full working table earns method marks even if one sum slips.

Watch out!

The syllabus does not ask you to write code for a graph or a search. It asks you to explain how graphs help AI and to trace Dijkstra and A* by hand, so practise clear, complete working tables.

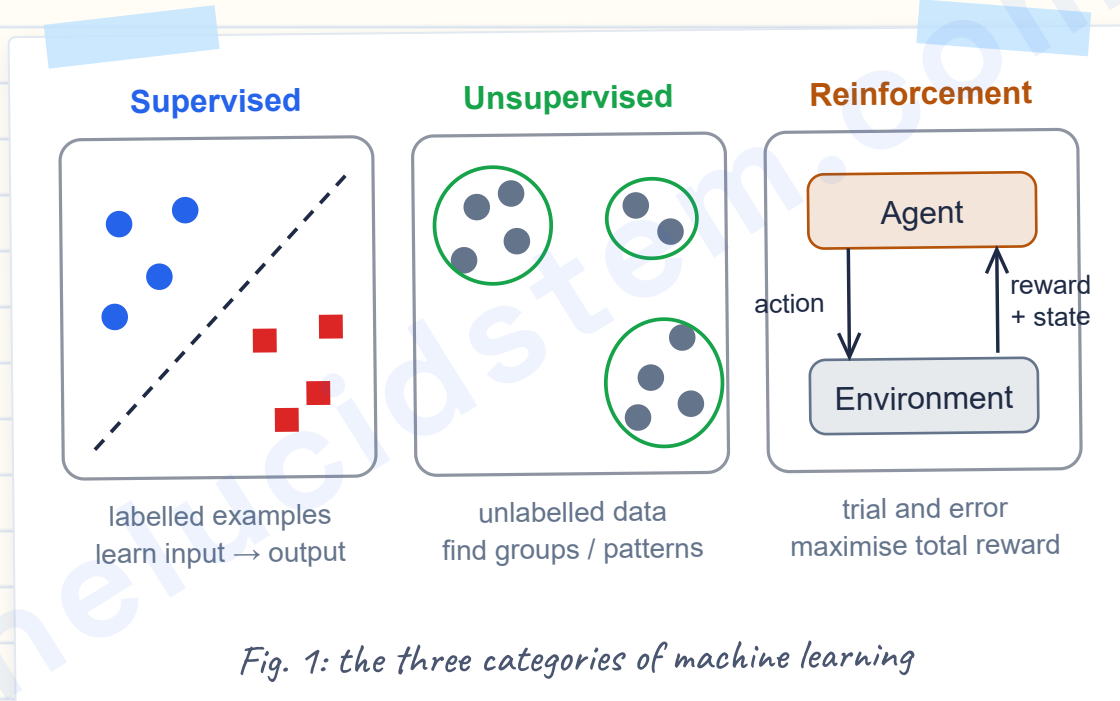
The edge A to B is the most direct way to B, but it is not the shortest: 4 is beaten by $A \rightarrow C \rightarrow B = 3$. Always compare and update; never treat a distance as final until you visit that node.

Do not stop A* when the goal first appears in the open list. Stop when the goal is chosen (has the smallest f): a cheaper route to it may still be found before then.

2 Machine learning: supervised, unsupervised and reinforcement

Syllabus 18.1

Machine learning is a branch of AI in which a program improves its performance on a task by learning patterns from **data**, instead of following rules a programmer has written. In **supervised learning** the training data is **labelled** with the correct output, and the model learns to map inputs to outputs (classification or regression). In **unsupervised learning** the data is **unlabelled** and the model finds its own structure, such as **clusters**. In **reinforcement learning** an **agent** acts in an **environment** and learns by trial and error to maximise its **reward**. **Regression** predicts a continuous value by fitting a line or curve that minimises the error.



Worked example: Least squares by hand

Find the least squares regression line for the data in the table and use it to predict the score after 6 hours of revision.

- Means: $\bar{x} = 15 \div 5 = 3$ and $\bar{y} = 300 \div 5 = 60$.
- Deviations $x - \bar{x}$: $-2, -1, 0, 1, 2$. Deviations $y - \bar{y}$: $-8, -5, 1, 4, 8$.
- $\Sigma(x - \bar{x})(y - \bar{y}) = 16 + 5 + 0 + 4 + 16 = 41$ and $\Sigma(x - \bar{x})^2 = 4 + 1 + 0 + 1 + 4 = 10$.
- $m = 41 \div 10 = 4.1$ and $c = 60 - 4.1 \times 3 = 47.7$, so $y = 4.1x + 47.7$.
- Predictions 51.8, 55.9, 60.0, 64.1, 68.2 give residuals 0.2, -0.9 , 1.0, -0.1 , -0.2 ; $MSE = 1.9 \div 5 = 0.38$.
- For $x = 6$: $y = 4.1 \times 6 + 47.7 = 72.3$.

Answer: $y = 4.1x + 47.7$; predicted score 72.3

Start from the data.

- ✓ **Labelled data:** supervised (classification for categories, regression for numbers).
- ✓ **Unlabelled data:** unsupervised (clustering, associations).
- ✓ **Score actions, no data set:** reinforcement (agent, environment, reward, policy).
- ✓ **Regression:** $m = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sum(x - \bar{x})^2}$, $c = \bar{y} - m\bar{x}$.

Watch out!

Unsupervised does not mean "no human involved". It means the training data has no labels, so the algorithm must find the groups itself; a human still chooses the data and interprets the clusters.

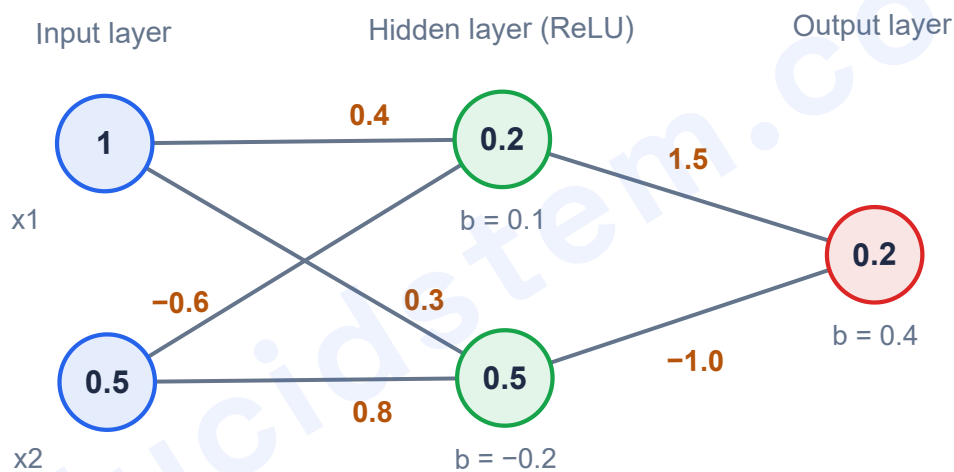
A regression model is only trustworthy inside the range of its training data. Predicting for 40 hours gives 211.7, an impossible score: extrapolating far beyond the data breaks the model.

"Supervised" refers to labelled training data, not to a person watching the program run. Justify your choice by describing the data.

3 Neural networks, deep learning and back propagation

Syllabus 18.1

An **artificial neural network** (ANN) is made of **nodes** (artificial neurons) arranged in an **input layer**, one or more **hidden layers** and an **output layer**. Each connection has a **weight**. A node calculates the **weighted sum** of its inputs plus a **bias** and passes it through an **activation function**. **Deep learning** uses ANNs with **many hidden layers** that learn features from raw data by themselves. The network is trained by **back propagation of errors**: compare the output with the target, pass the error backwards through the layers, and adjust every weight a little to reduce the error.



each node: weighted sum of inputs + bias, then activation

Fig. 1: a small neural network with weights and biases; the values in the nodes are a forward pass for inputs 1 and 0.5

Worked example: A forward pass

Use the weights and biases in Fig. 1 to calculate the output of the network for inputs $x_1 = 1$ and $x_2 = 0.5$. The hidden nodes use ReLU; the output node uses no activation.

1. Top hidden node: $0.4 \times 1 + (-0.6) \times 0.5 + 0.1 = 0.4 - 0.3 + 0.1 = 0.2$. $\text{ReLU}(0.2) = 0.2$.

2. Bottom hidden node: $0.3 \times 1 + 0.8 \times 0.5 + (-0.2) = 0.3 + 0.4 - 0.2 = 0.5$. $\text{ReLU}(0.5) = 0.5$.

3. Output node: $1.5 \times 0.2 + (-1.0) \times 0.5 + 0.4 = 0.3 - 0.5 + 0.4 = 0.2$.

Answer: The network outputs 0.2

Forward, then back.

- ✓ **Node:** weighted sum of inputs + bias, then the activation function.
- ✓ **Deep learning:** many hidden layers that learn features; needs lots of data and processing power.
- ✓ **Back propagation:** output error, passed backwards, each weight adjusted by its contribution.
- ✓ **Update:** new weight = old weight - learning rate $\times$ gradient.

Watch out!

Do not forget the bias. It is added to the weighted sum before the activation function, and leaving it out changes the answer.

Deep learning is not a separate fourth category alongside supervised, unsupervised and reinforcement learning. It is a technique (deep neural networks) that can be used within any of them.

"The error is sent back and the weights are changed" is too vague for full marks. Say that the error is calculated at the output, propagated backwards layer by layer, and each weight is adjusted in proportion to its contribution to the error.