Artificial Intelligence (AI)

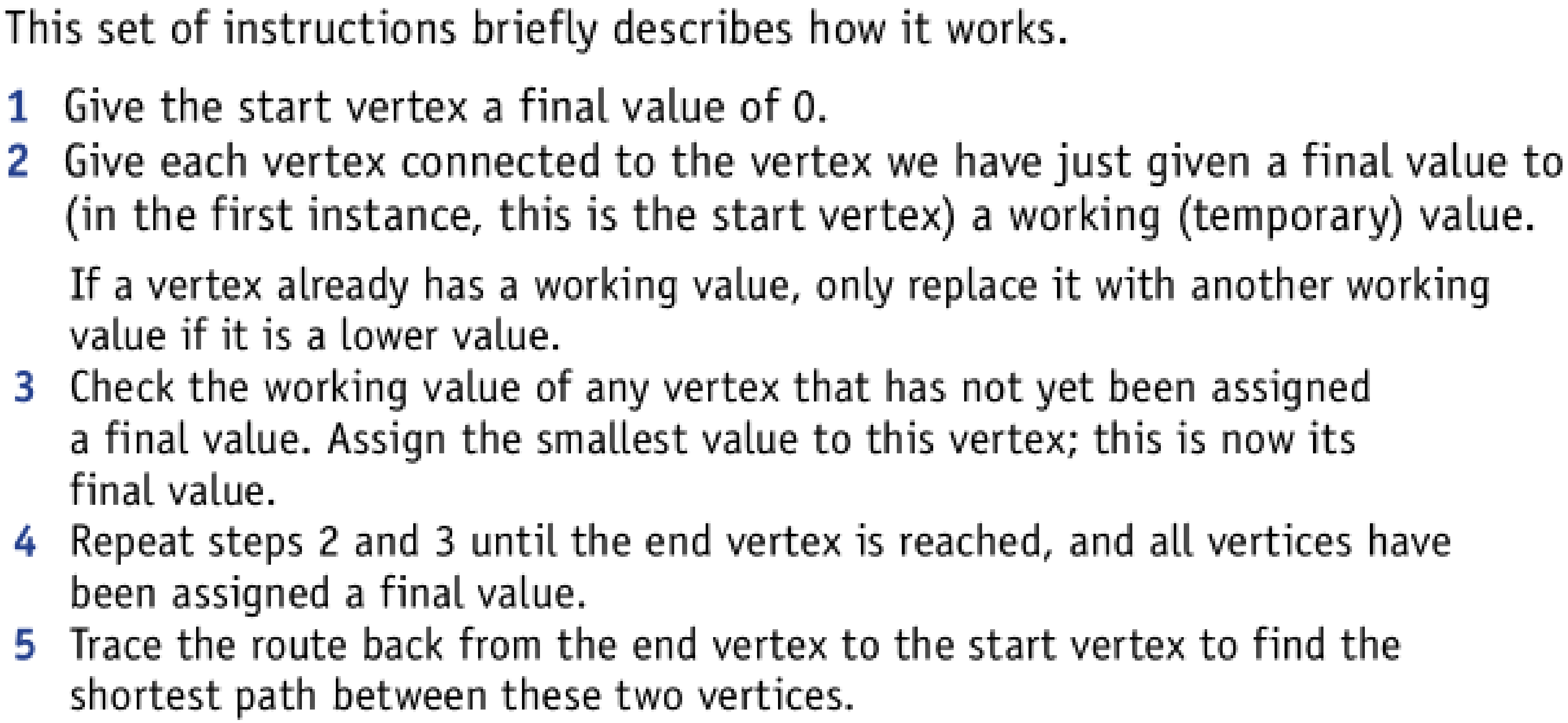
- Difficult to give a precise definition
- An early attempt at an imprecise definition:

  How to make computers do things that, at the moment, people can do better
- AI is a wide and diverse area of study and application

Graphs in AI

- A graph contains nodes connected by edges
- Can represent many different scenarios
- Each edge contains a value representing, in the broadest terms, a ‘cost’ in moving from one of the connected nodes to the other connected node
- Dijkstra’s algorithm and the A* algorithm are used for finding the best path through the graph

Dijkstra’s algorithm (pronounced dyke – strah) is a method of finding the shortest path between two points on a graph. Each point on the graph is called a node or a vertex (for more information on the features and uses of graphs, see Chapter 19). It is the basis of technology such as GPS tracking and, therefore, is an important part of AI.



This set of instructions briefly describes how it works.

1 Give the start vertex a final value of 0.

2 Give each vertex connected to the vertex we have just given a final value to (in the first instance, this is the start vertex) a working (temporary) value.

If a vertex already has a working value, only replace it with another working value if it is a lower value.

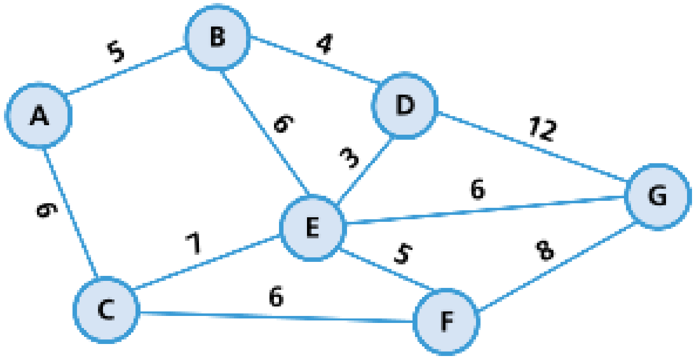
3 Check the working value of any vertex that has not yet been assigned a final value. Assign the smallest value to this vertex; this is now its final value.

4 Repeat steps 2 and 3 until the end vertex is reached, and all vertices have been assigned a final value.

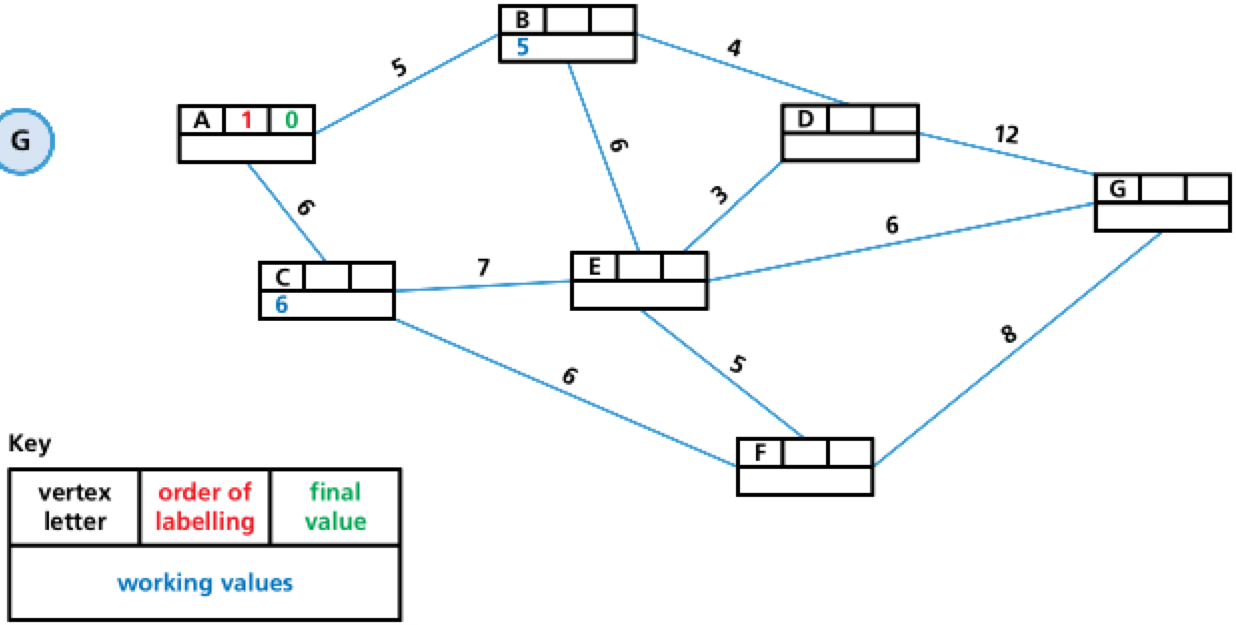
5 Trace the route back from the end vertex to the start vertex to find the shortest path between these two vertices.

Here is a step-by-step example.

Suppose we have the following graph (route map) with seven vertices labelled A to G. We want to find the shortest path between A and G. The numbers show the distance between each pair of vertices.



First, redraw the diagram, replacing the circled letters as per the key:

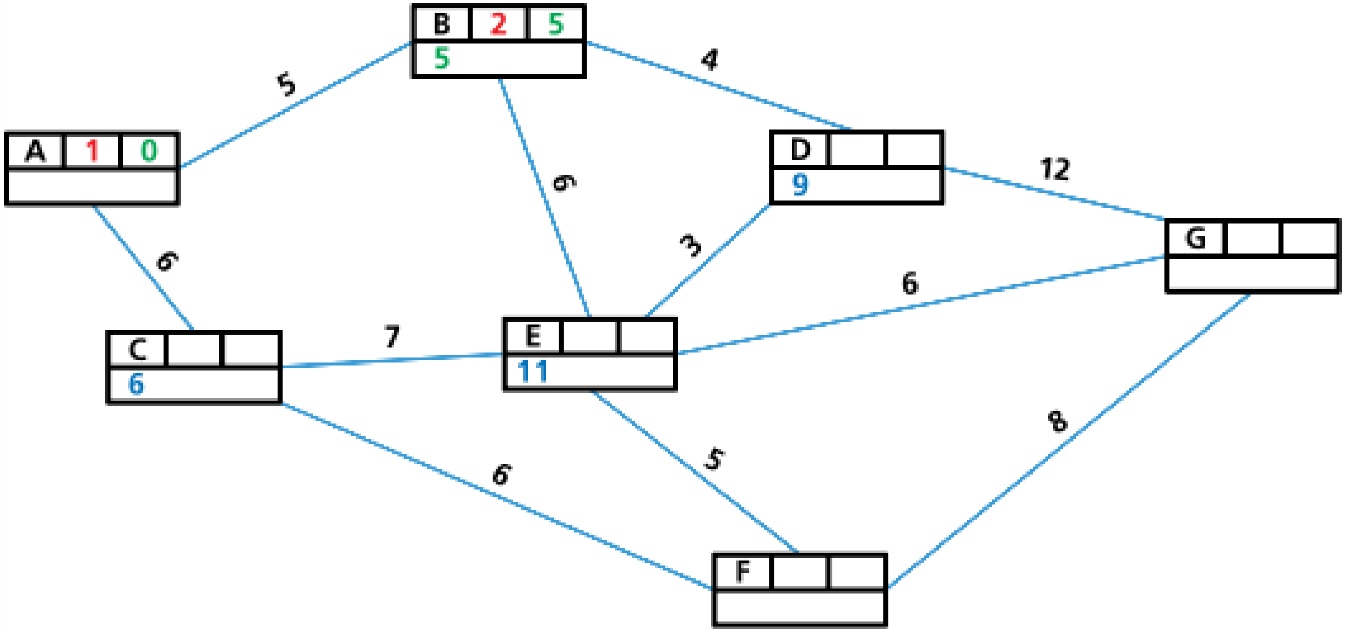


Set the final value of the start vertex (vertex A) to 0 (as per step 1 above).

The two vertices connected to A (B and C) are given the working values 5 and 6 respectively.

Make the smallest working value (vertex B) a final value. Then give the vertices connected to B (D and E) working values based on the original distances. The working value for E is the final value of B plus the value of B to E (5 + 6 = 11). The working value for D is the final value of B plus the value of B to D (5 + 4 = 9).

The diagram now looks like this:



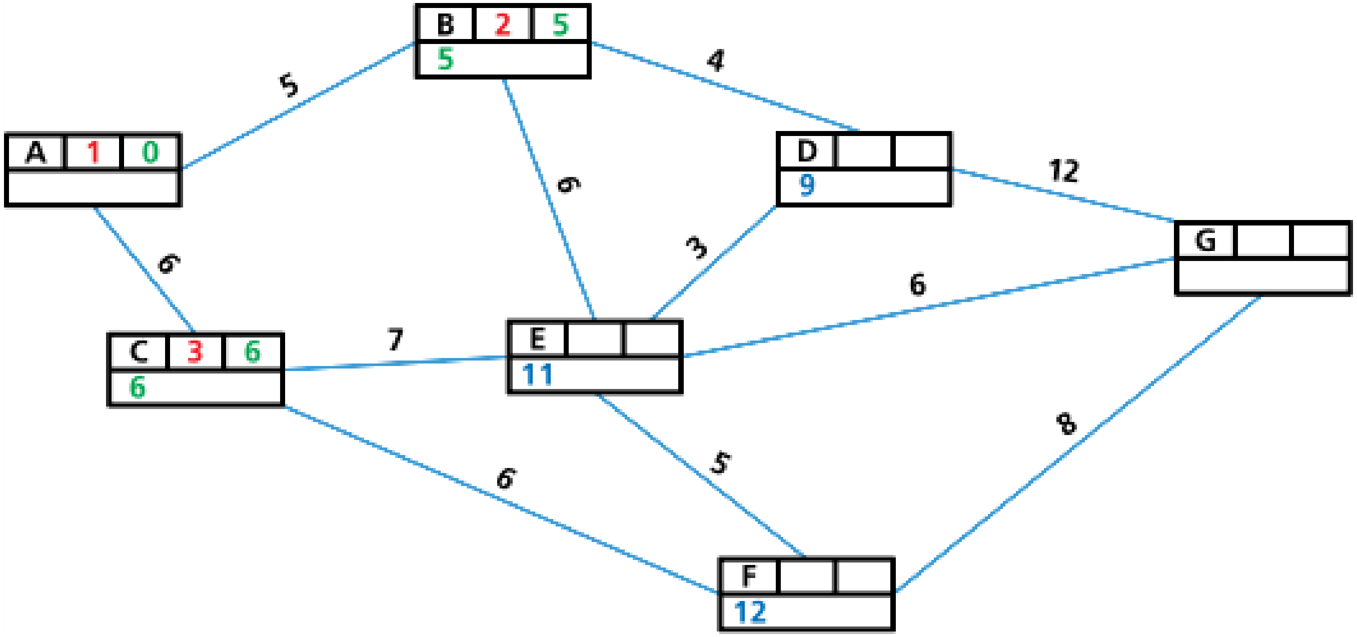
Make the smallest working value a final value: vertex C becomes 6.

Now give working values to all vertices connected to C. Note that the working value for E remains 11 since the final value of C plus the value of C to E is 13, which is greater than 11.

Vertex D retains its working value since it is not connected to C and is not affected.

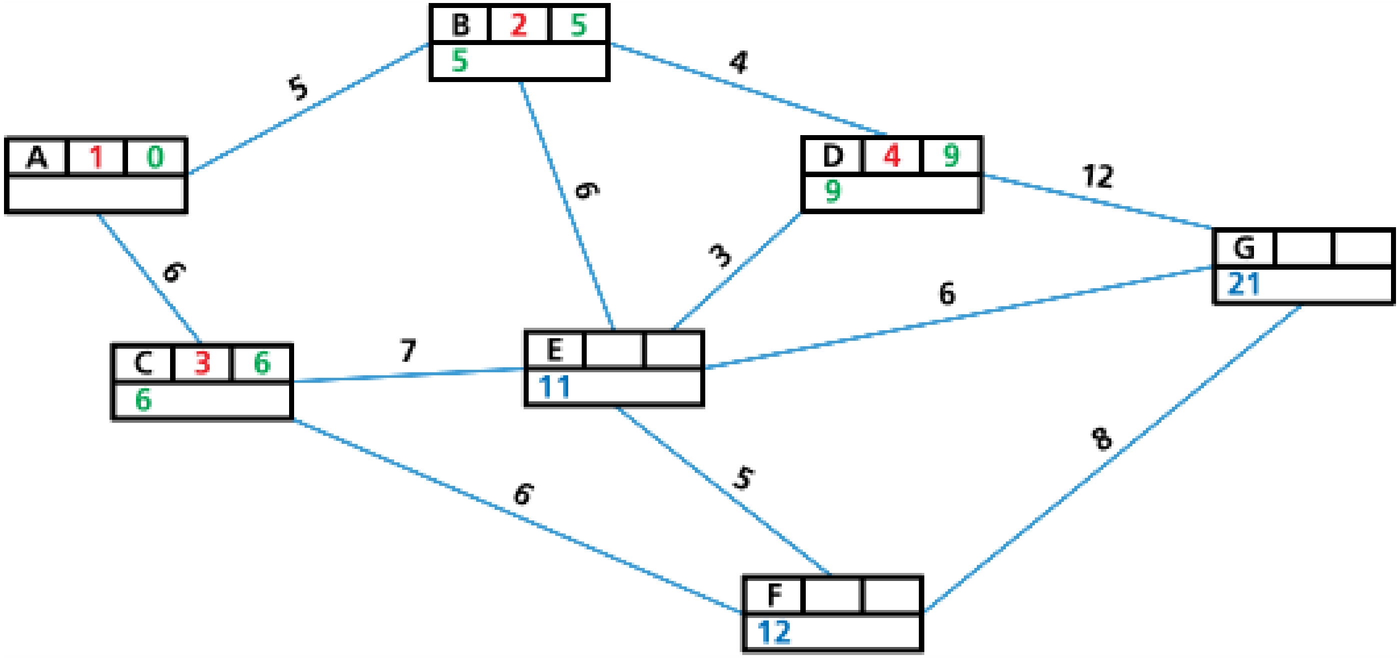
Vertex F takes the working value of C plus the value of C to F (6 + 6 = 12).

The diagram now looks like this:



Vertex D now has the smallest working value (9), so this becomes a final value.

Vertices E and G are connected to D, so these are now assigned working values. Note that G has the working value 21 since it is the final value of D plus the value of D to G (9 + 12 = 21); E keeps the value of 11 since the final value of D plus the value of D to E is greater than 11 (9 + 11 = 20).



Vertex E now has the smallest working value (11), so this becomes a final value.

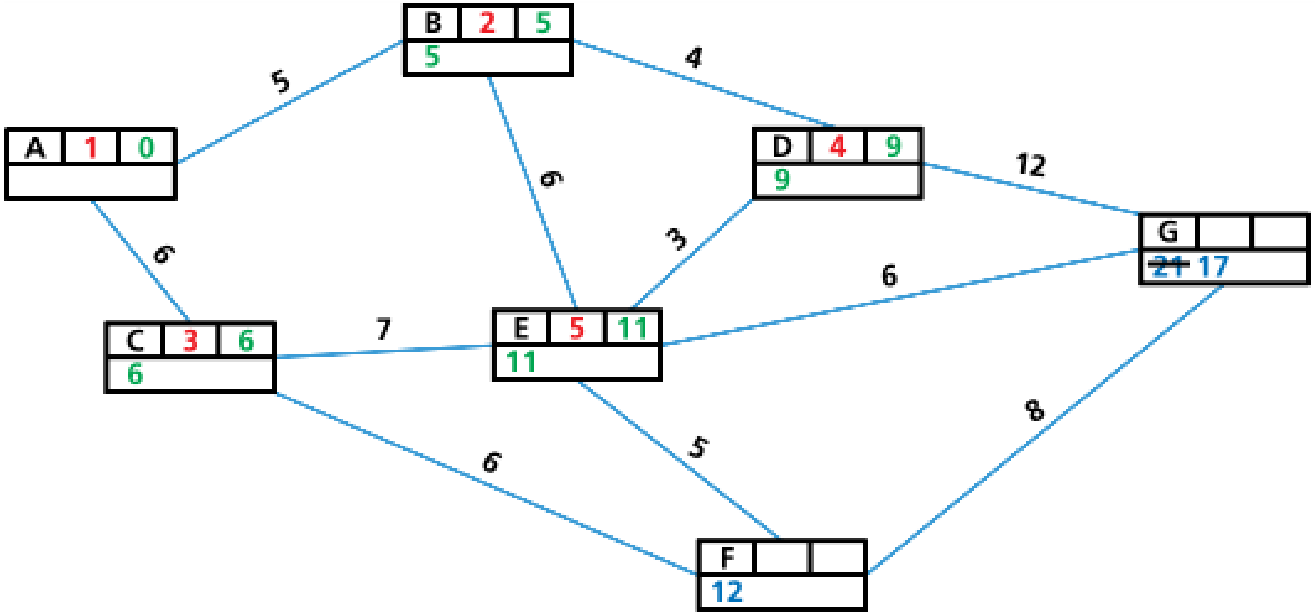
Vertices D, F and G are all connected to E.

D already has a final value so it can be ignored.

F retains its value of 12 since E + E to F = 16 (> 12).

G changes since E + E to G = 17 (< 21).

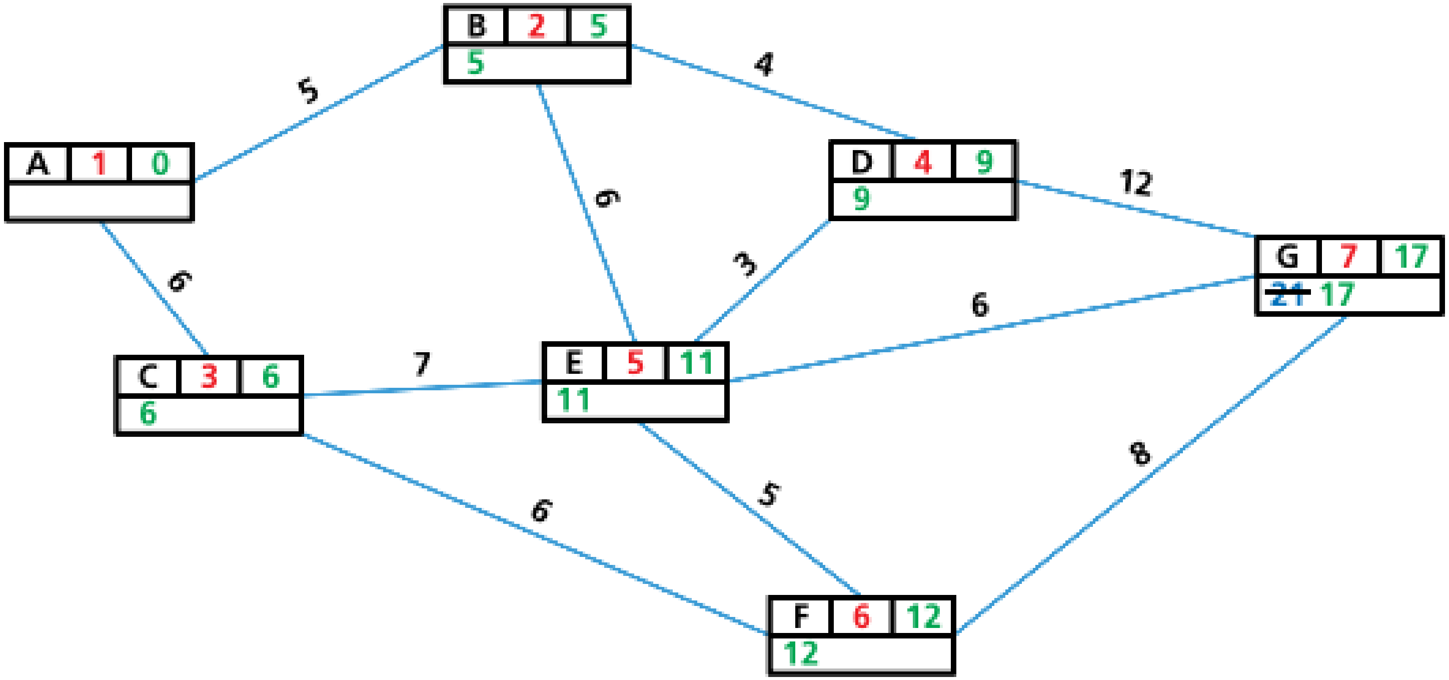
The diagram now looks like this:



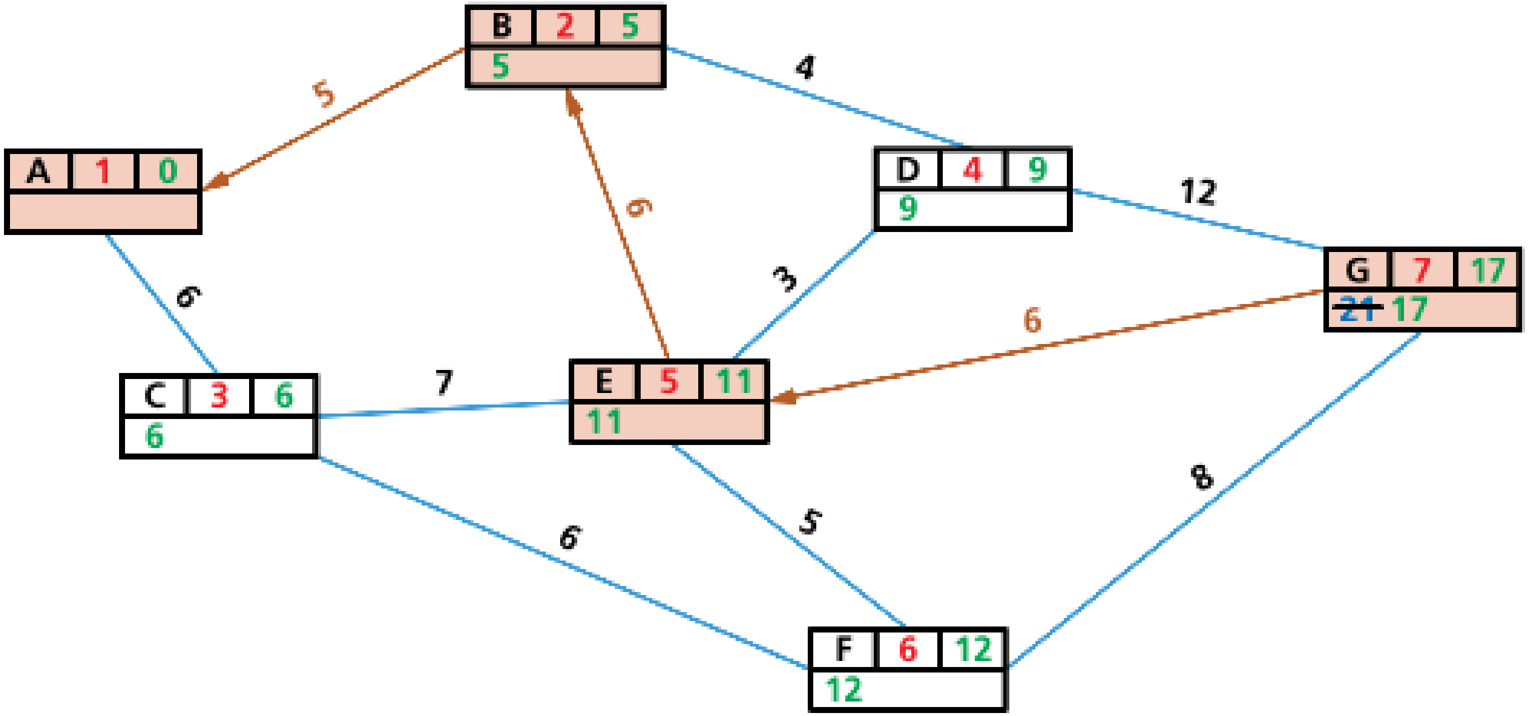
Vertex F now has the smallest working value (12), so this becomes a final value.

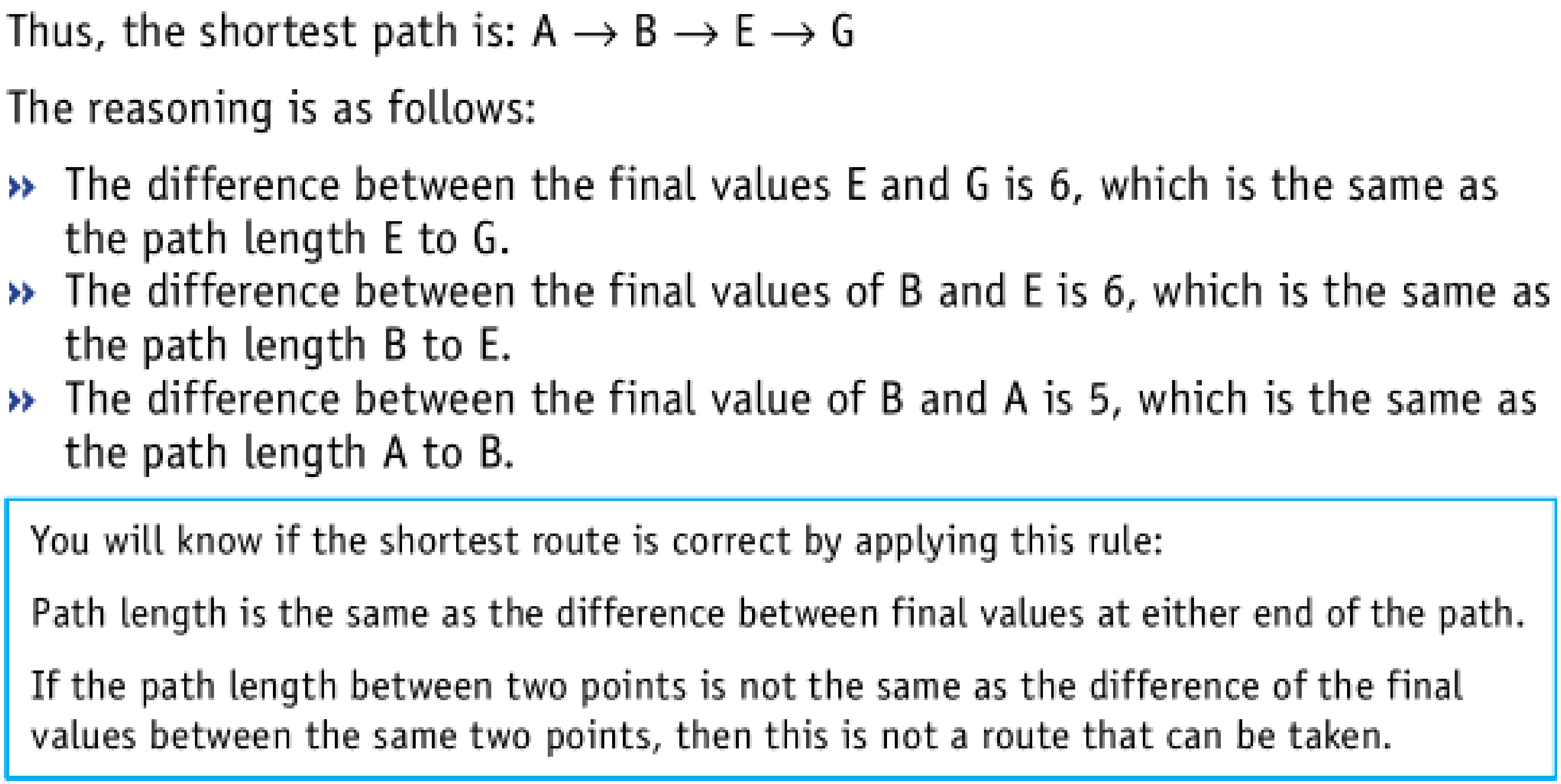
G retains its value of 17 since F + F to G = 20 (> 17).

The final diagram now looks like this:



The final step is to work back from G to A.





Thus, the shortest path is: A → B → E → G

The reasoning is as follows:

- The difference between the final values E and G is 6, which is the same as the path length E to G.
- The difference between the final values of B and E is 6, which is the same as the path length B to E.
- The difference between the final value of B and A is 5, which is the same as the path length A to B.

You will know if the shortest route is correct by applying this rule:

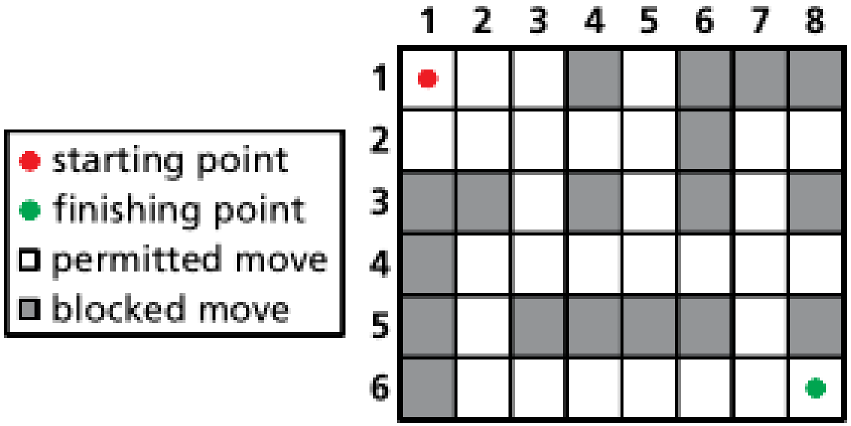
Path length is the same as the difference between final values at either end of the path.

If the path length between two points is not the same as the difference of the final values between the same two points, then this is not a route that can be taken.

A* algorithm

The A* algorithm is a modification of the Dijkstra algorithm designed to improve matters. It's based on Dijkstra, but adds an extra heuristic (h) value – an 'intelligent guess' on how far we have to go to reach the destination most efficiently.

Suppose we have the following graph made up of an 8 × 6 matrix. White squares show permitted moves, and grey squares show blocked moves.



	1	2	3	4	5	6	7	8
1	🔴	□	□	▨	□	▨	▨	▨
2	□	□	□	□	□	▨	□	□
3	▨	▨	□	▨	□	▨	□	▨
4	▨	□	□	□	□	□	□	□
5	▨	□	▨	▨	▨	▨	□	▨
6	▨	□	□	□	□	□	□	🟢

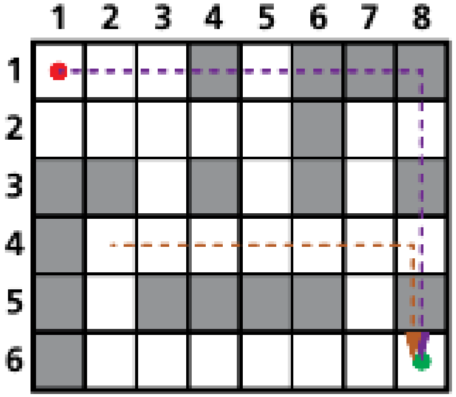
Each of the parts of the graph are called nodes (or vertices). Each node has four values

- h (the heuristic value)
- g (movement cost)
- f (sum of g and h values)
- n (previous node in the path).

Note that the weight of a node usually represents movement cost, which is the distance between the two nodes.

First, find the heuristic values (distances) using the Manhattan method (named after the criss-cross street arrangement in New York). The distance from the starting square (1, 1) to the end square (8, 6) is 12 squares (follow the purple line in the diagram below). Similarly, the distance from square (2, 4) to (8, 6) is eight squares (follow the orange line in the diagram below).

Note: you can ignore the blocked moves when calculating heuristic distance from each node to the final node.

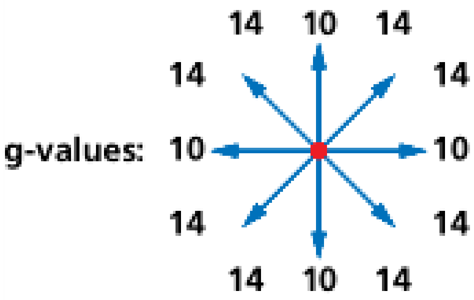


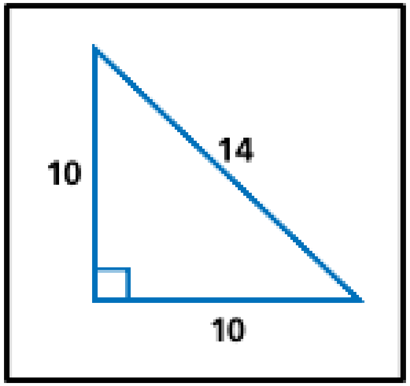
Use this method to find the heuristic distances for all permitted nodes:

h-values:

	1	2	3	4	5	6	7	8
1	12	11	10	▨	8	▨	▨	▨
2	11	10	9	8	7	▨	5	4
3	▨	▨	8	▨	6	▨	4	▨
4	▨	8	7	6	5	4	3	2
5	▨	7	▨	▨	▨	▨	2	▨
6	▨	6	5	4	3	2	1	–

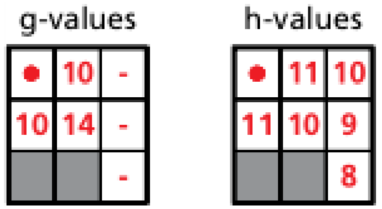
Now, find the g-values (the movement costs). Since we can either move up/down, left/right or diagonally, we can choose our g-values based on a right-angled triangle. To make the maths easy we will use a triangle with sides 10, 10 and 14:





Find the f values using f(n) = g(n) + h(n).

Starting with square (1, 1), look at the surrounding squares:



g-values

●	10	-
10	14	-
▨	▨	-

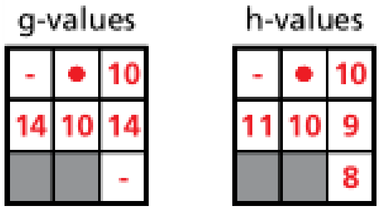
h-values

●	11	10
11	10	9
▨	▨	8

- square (1, 2): f = 10 + 11 = 21
- square (2, 1): f = 10 + 11 = 21
- square (2, 2): f = 14 + 10 = 24

Since 21 < 24, (1, 2) or (2, 1) are the possible directions.

We will choose (2, 1) as the next step:



g-values

-	●	10
14	10	14
▨	▨	-

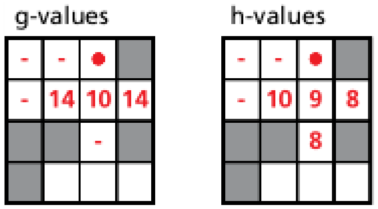
h-values

-	●	10
11	10	9
▨	▨	8

- square (3, 1): f = 10 + 10 = 20
- square (3, 2): f = 14 + 9 = 23
- square (1, 2): f = 14 + 11 = 25
- square (2, 2): f = 10 + 10 = 20

Since 20 is the smallest value, the next stage can be (3, 1) or (2, 2).

We will choose (3, 1) as the next step:



g-values

-	-	●	▨
-	14	10	14
▨	▨	-	▨
▨			

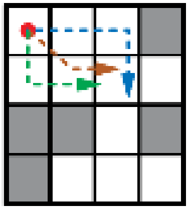
h-values

-	-	●	▨
-	10	9	8
▨	▨	8	▨
▨			

- square (2, 2): f = 14 + 10 = 24
- square (3, 2): f = 10 + 9 = 19
- square (4, 2): f = 14 + 8 = 22

Since square (3, 2) is the smallest value, this must be the next step.

Now look at the possible routes already found to decide where to move next:



Route 1 has the values:
21 + 20 + 19 = 60

Route 2 has the values:
24 + 19 = 43

Route 3 has the values:
21 + 20 + 19 = 60

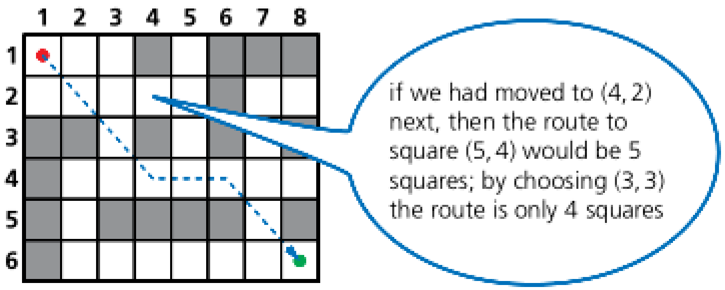
remember each of the values is found by adding the g-value to the h-value

It seems route 2 is the shortest route: (1, 1) → (2, 2) → (3, 2).

When considering the next squares (3, 3) or (4, 2), and applying the above rules, it becomes clear that the next stage in the route is:

(1, 1) → (2, 2) → (3, 3)

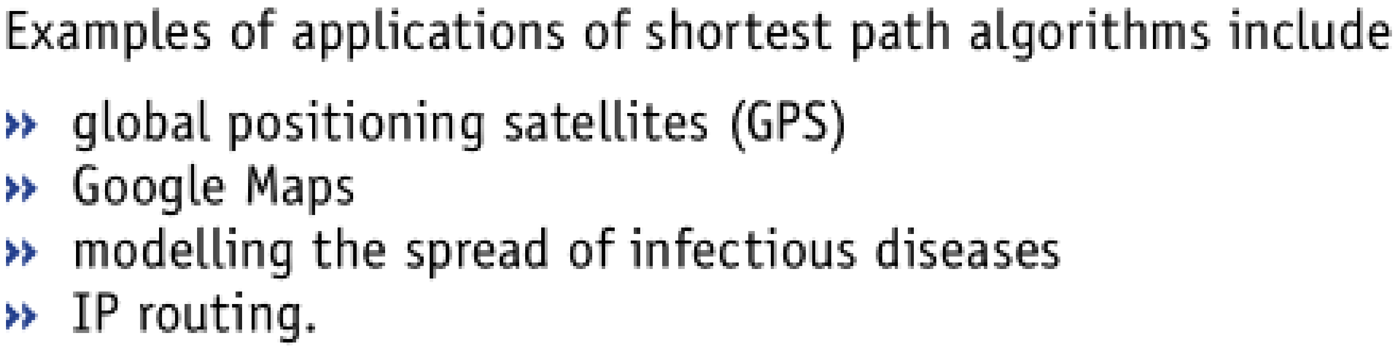
Continue throughout the matrix and produce the following shortest route from (1, 1) to (8, 6):



if we had moved to (4, 2) next, then the route to square (5, 4) would be 5 squares; by choosing (3, 3) the route is only 4 squares

The shortest path is:

(1, 1) → (2, 2) → (3, 3) → (4, 4) → (5, 4) → (6, 4) → (7, 5) → (8, 6)



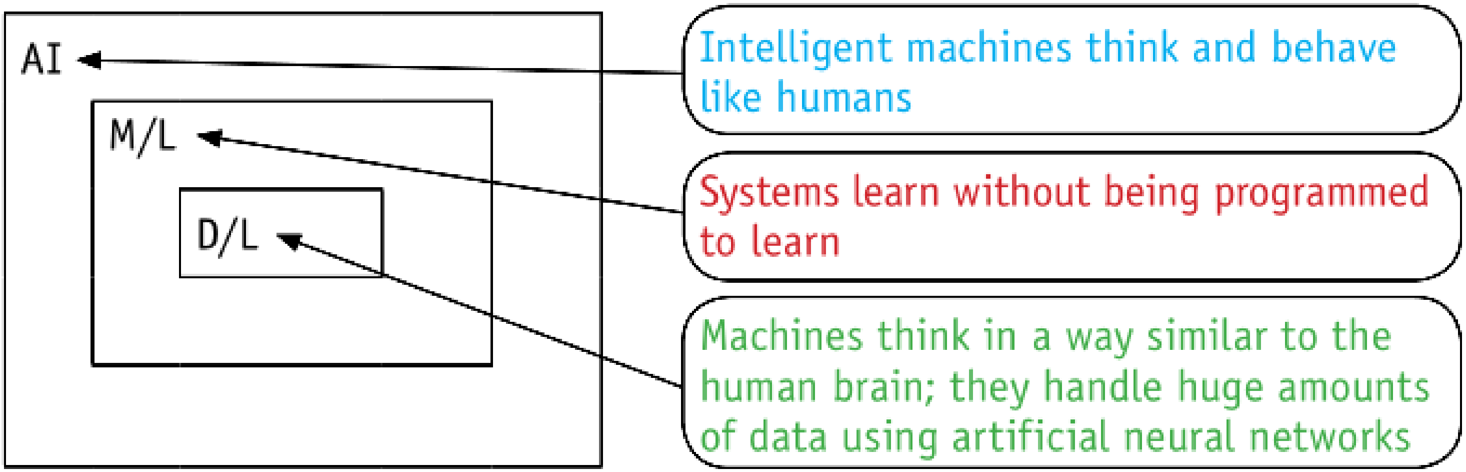
Examples of applications of shortest path algorithms include

- global positioning satellites (GPS)
- Google Maps
- modelling the spread of infectious diseases
- IP routing.

Machine learning

- Concerns the use of an AI system where the aim is for the system to perform better as a result of experience
- Learning can be:
  - supervised
  - unsupervised
  - reinforcement
- Regression methods can be used

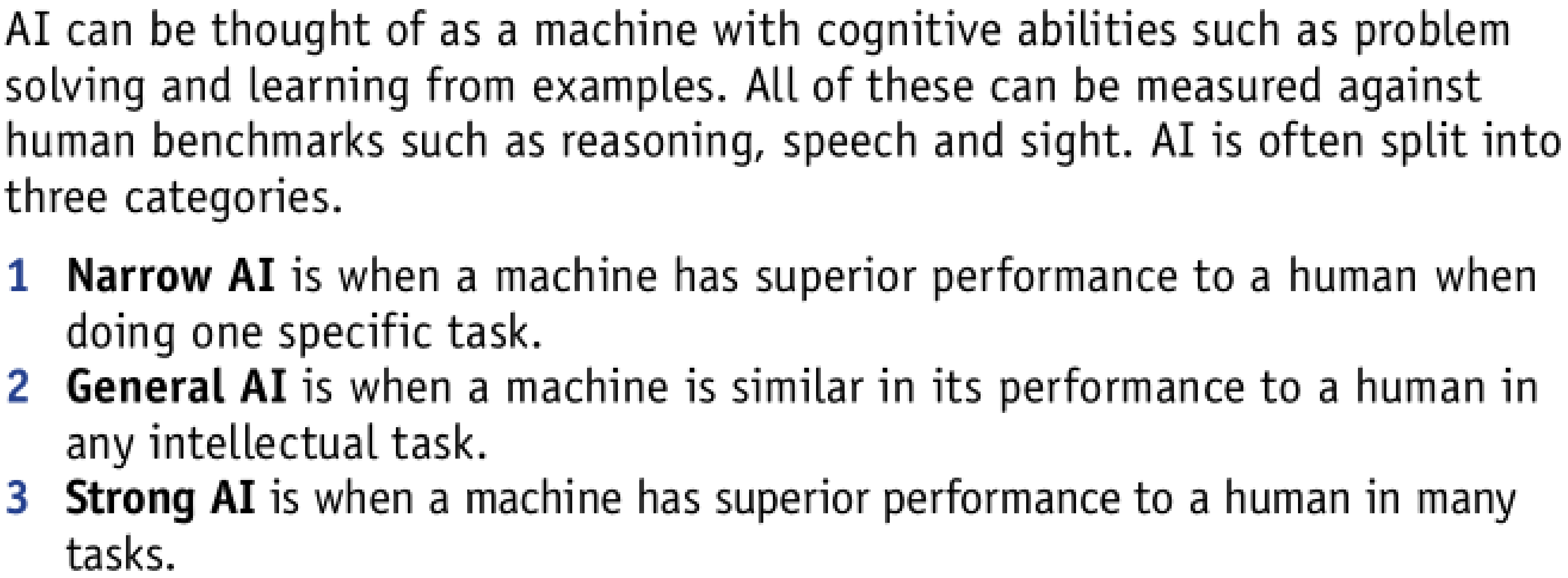
Figure shows the link between artificial intelligence (AI), machine learning and deep learning. Deep learning is a subset of machine learning, which is itself a subset of AI.



Intelligent machines think and behave like humans

Systems learn without being programmed to learn

Machines think in a way similar to the human brain; they handle huge amounts of data using artificial neural networks

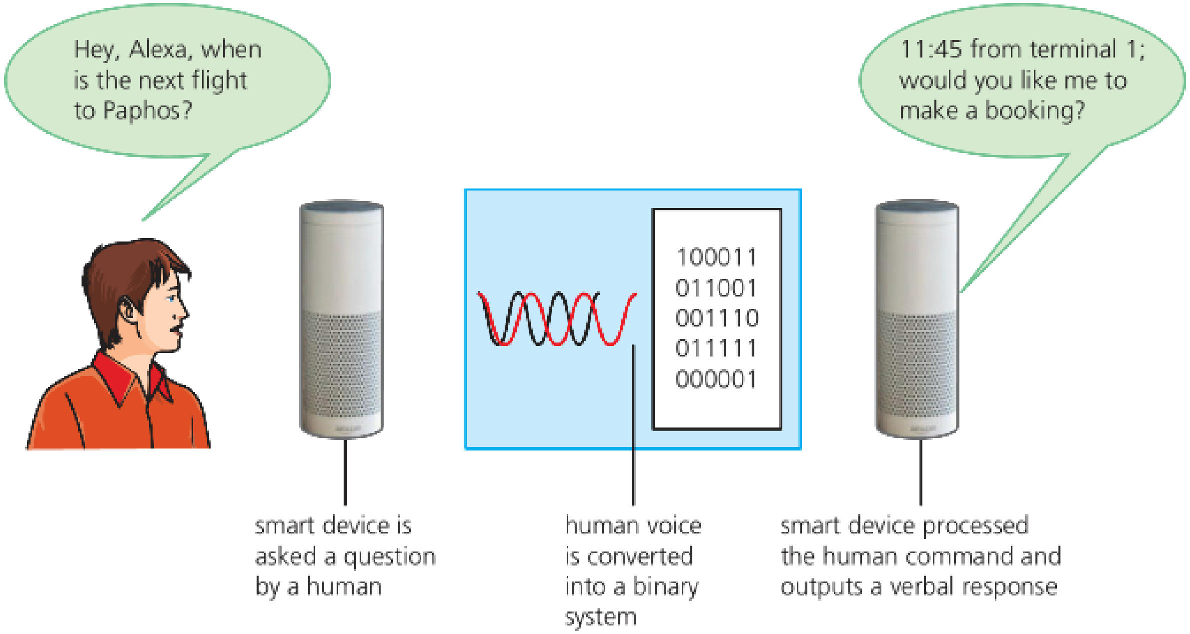


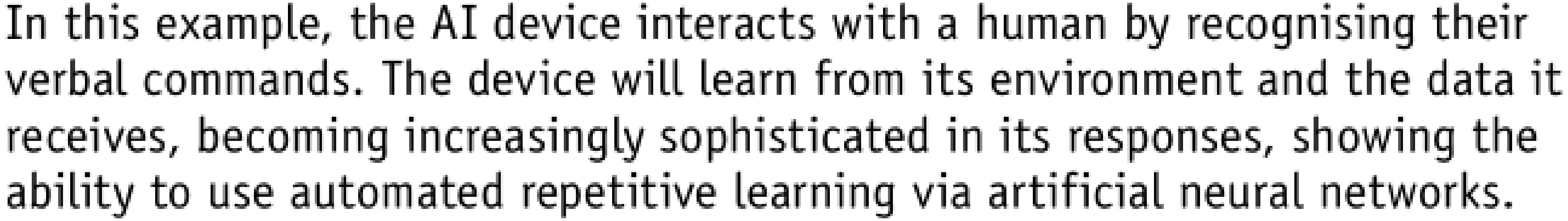
AI can be thought of as a machine with cognitive abilities such as problem solving and learning from examples. All of these can be measured against human benchmarks such as reasoning, speech and sight. AI is often split into three categories.

1. **Narrow AI** is when a machine has superior performance to a human when doing one specific task.
2. **General AI** is when a machine is similar in its performance to a human in any intellectual task.
3. **Strong AI** is when a machine has superior performance to a human in many tasks.

Examples of AI include

- news generation based on live news feeds (this will involve some form of human interaction)
- smart home devices (such as Amazon Alexa, Google Now, Apple Siri and Microsoft Cortana); again these all involve some form of human interaction

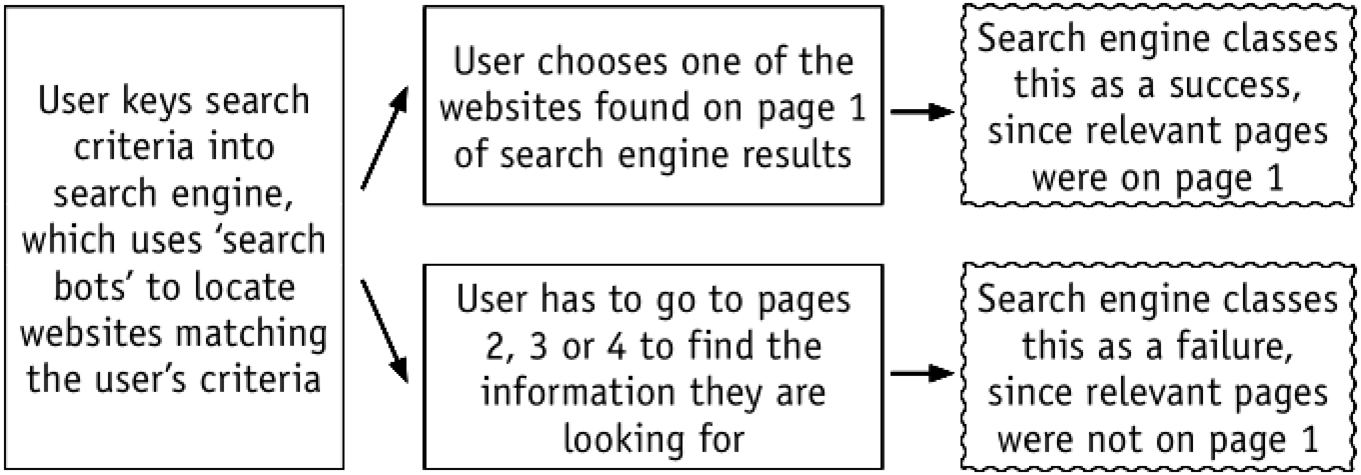

Hey, Alexa, when is the next flight to Paphos?
11:45 from terminal 1; would you like me to make a booking?
100011
011001
001110
011111
000001
smart device is asked a question by a human
human voice is converted into a binary system
smart device processed the human command and outputs a verbal response



In this example, the AI device interacts with a human by recognising their verbal commands. The device will learn from its environment and the data it receives, becoming increasingly sophisticated in its responses, showing the ability to use automated repetitive learning via artificial neural networks.

Machine learning

Machine learning is a subset of AI, in which the algorithms are 'trained' and learn from their past experiences and examples. It is possible for the system to make predictions or even take decisions based on previous scenarios. They can offer fast and accurate outcomes due to very powerful processing capability. One of the key factors is the ability to manage and analyse considerable volumes of complex data – some of the tasks would take humans years, if they were to analyse the data using traditional computing processing methods. A good example is a search engine:



User keys search criteria into search engine, which uses 'search bots' to locate websites matching the user's criteria

User chooses one of the websites found on page 1 of search engine results

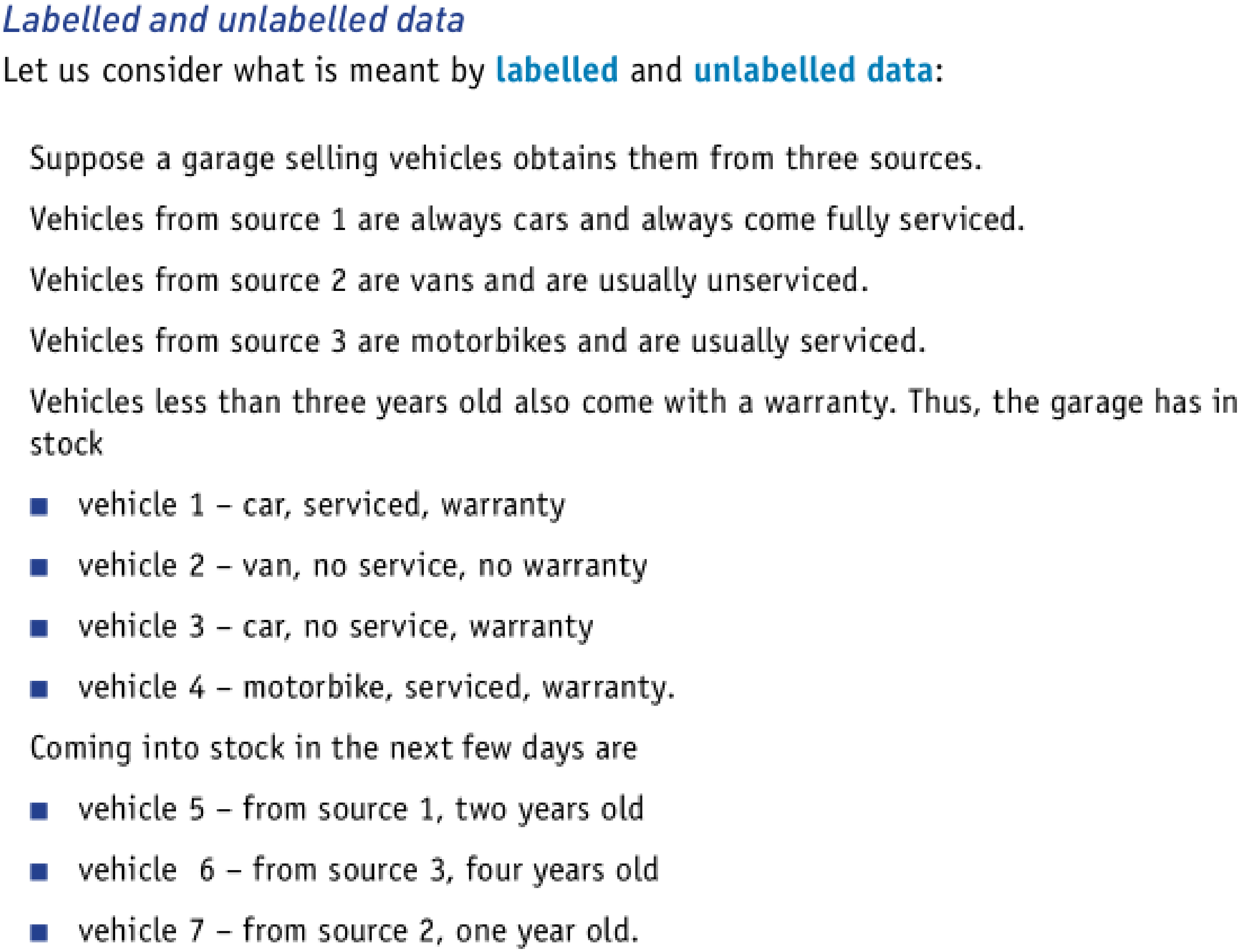
Search engine classes this as a success, since relevant pages were on page 1

User has to go to pages 2, 3 or 4 to find the information they are looking for

Search engine classes this as a failure, since relevant pages were not on page 1

The search engine will learn from its past performance, meaning its ability to carry out searches becomes more sophisticated and accurate.

Machine learning is a key part of AI and the various types of machine learning will be covered later in this chapter.



Labelled and unlabelled data

Let us consider what is meant by **labelled** and **unlabelled data**:

Suppose a garage selling vehicles obtains them from three sources.

Vehicles from source 1 are always cars and always come fully serviced.

Vehicles from source 2 are vans and are usually unserviced.

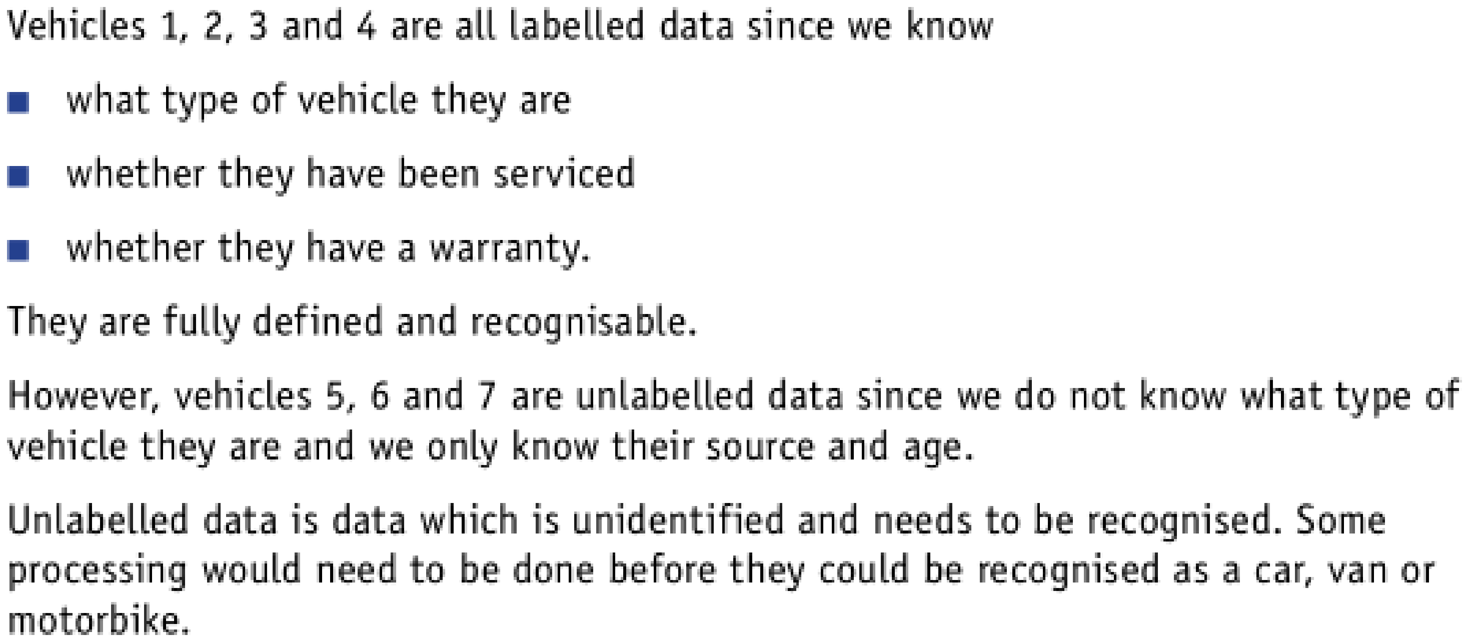
Vehicles from source 3 are motorbikes and are usually serviced.

Vehicles less than three years old also come with a warranty. Thus, the garage has in stock

- vehicle 1 – car, serviced, warranty
- vehicle 2 – van, no service, no warranty
- vehicle 3 – car, no service, warranty
- vehicle 4 – motorbike, serviced, warranty.

Coming into stock in the next few days are

- vehicle 5 – from source 1, two years old
- vehicle 6 – from source 3, four years old
- vehicle 7 – from source 2, one year old.



Vehicles 1, 2, 3 and 4 are all labelled data since we know

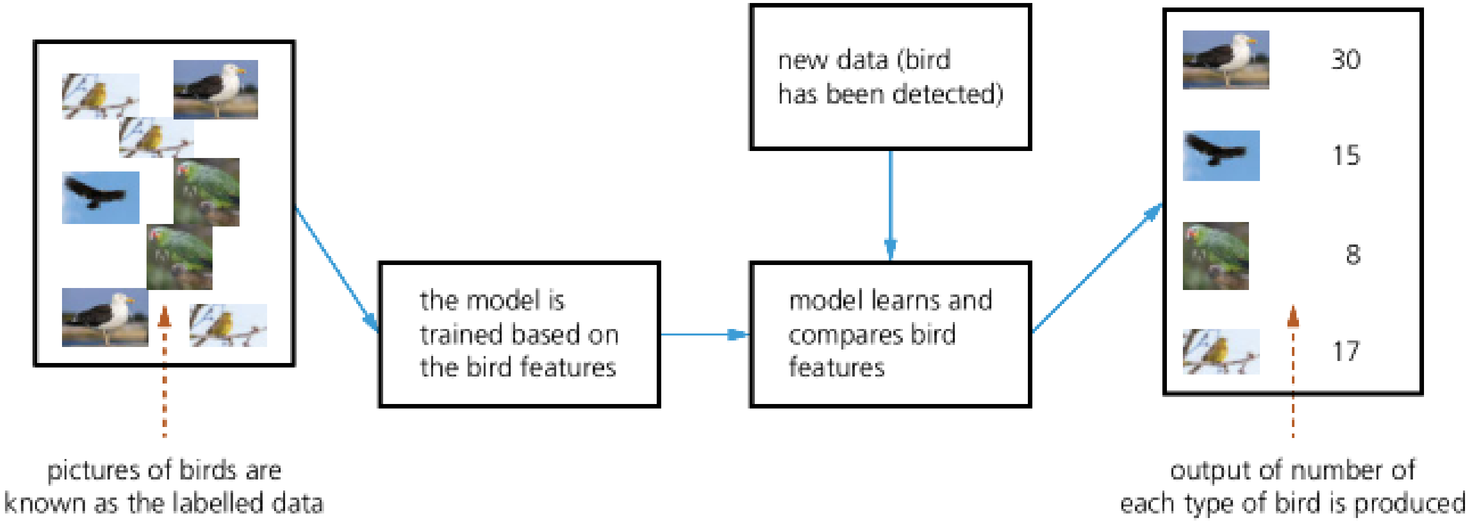
- what type of vehicle they are
- whether they have been serviced
- whether they have a warranty.

They are fully defined and recognisable.

However, vehicles 5, 6 and 7 are unlabelled data since we do not know what type of vehicle they are and we only know their source and age.

Unlabelled data is data which is unidentified and needs to be recognised. Some processing would need to be done before they could be recognised as a car, van or motorbike.

Now, suppose you want to automatically count the types of birds seen in a bird sanctuary. The proposed system will consider bird features such as shape of beak, colour of feathers, body size, and so on. This requires the use of labelled data to allow the birds to be recognised by the system



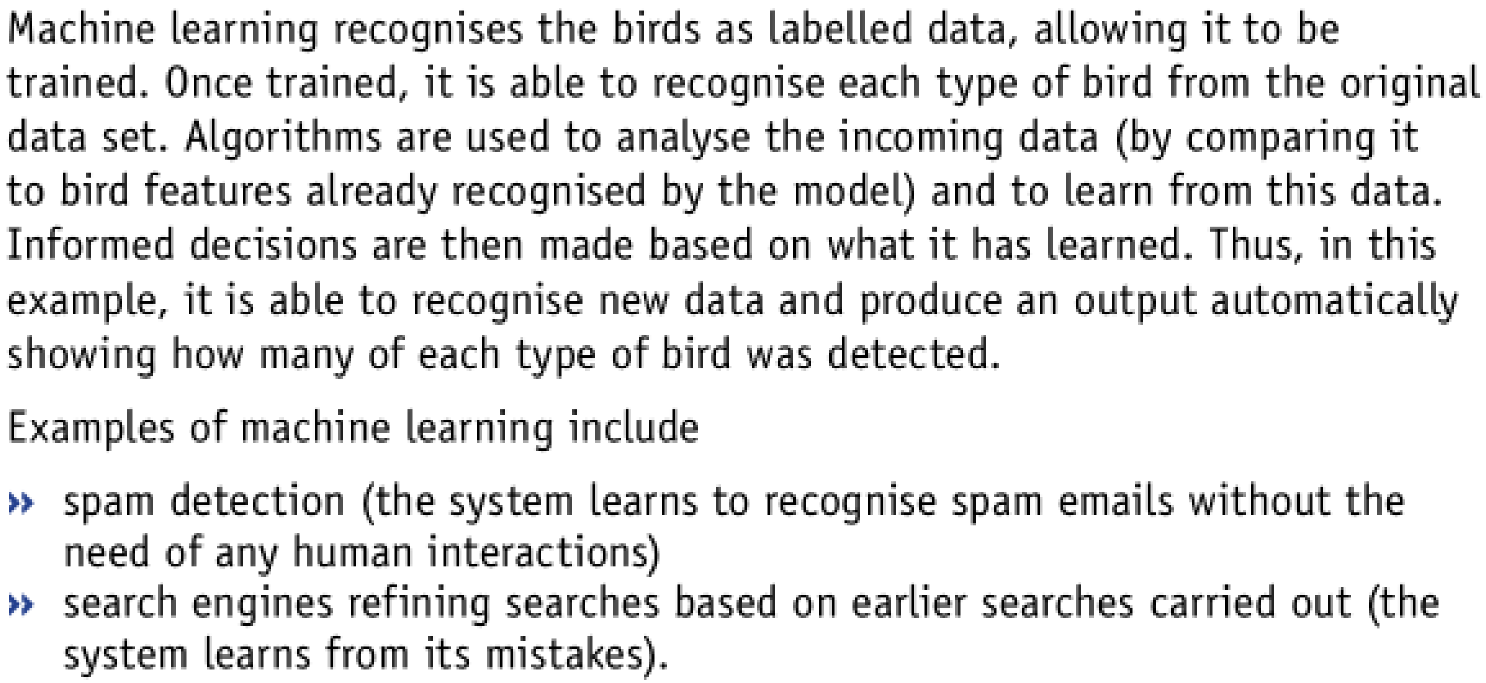
new data (bird has been detected)

the model is trained based on the bird features

model learns and compares bird features

pictures of birds are known as the labelled data

output of number of each type of bird is produced



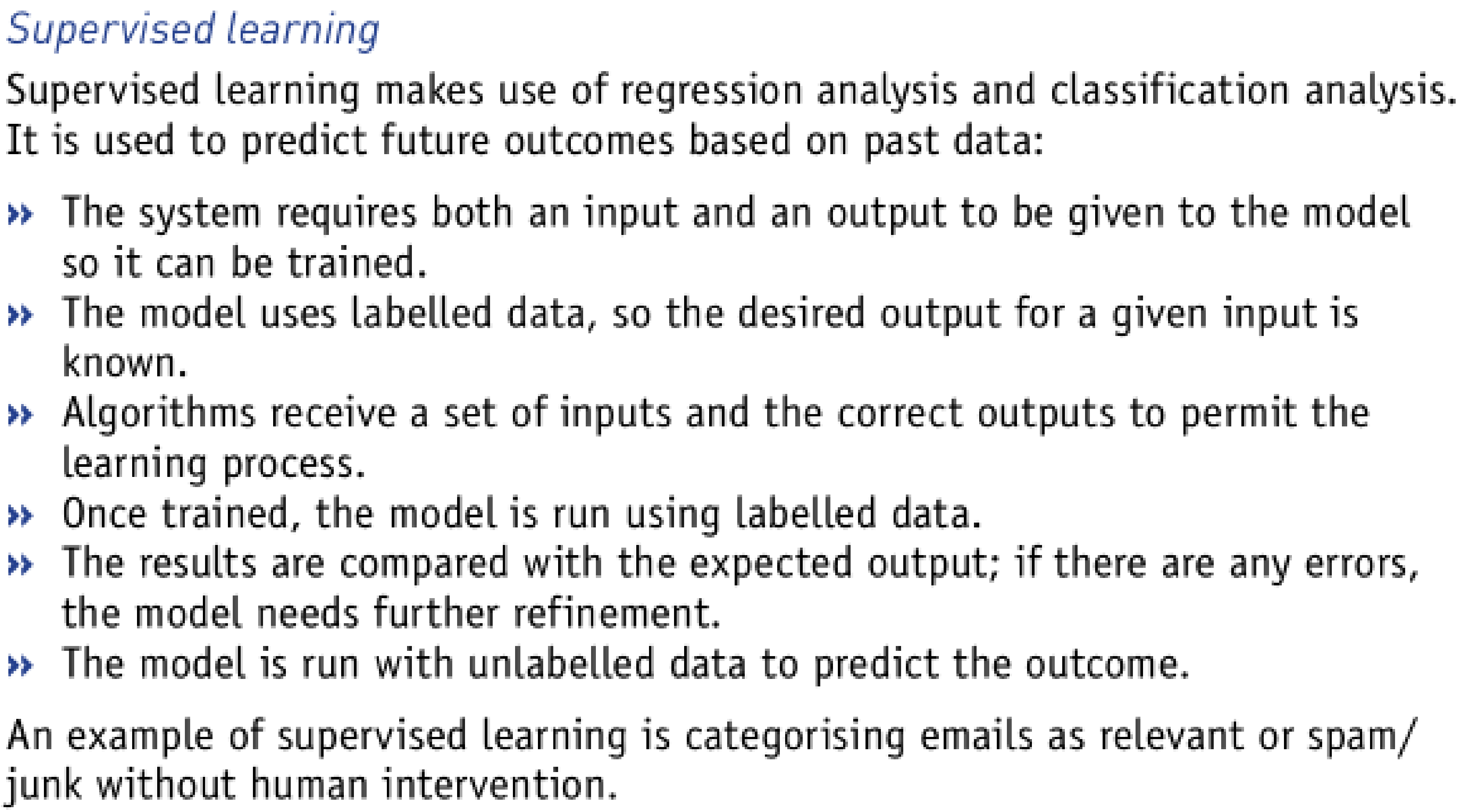
Machine learning recognises the birds as labelled data, allowing it to be trained. Once trained, it is able to recognise each type of bird from the original data set. Algorithms are used to analyse the incoming data (by comparing it to bird features already recognised by the model) and to learn from this data. Informed decisions are then made based on what it has learned. Thus, in this example, it is able to recognise new data and produce an output automatically showing how many of each type of bird was detected.

Examples of machine learning include

- spam detection (the system learns to recognise spam emails without the need of any human interactions)
- search engines refining searches based on earlier searches carried out (the system learns from its mistakes).

Types of machine learning

There are a number of different types of machine learning, including supervised, unsupervised learning, and reinforcement.

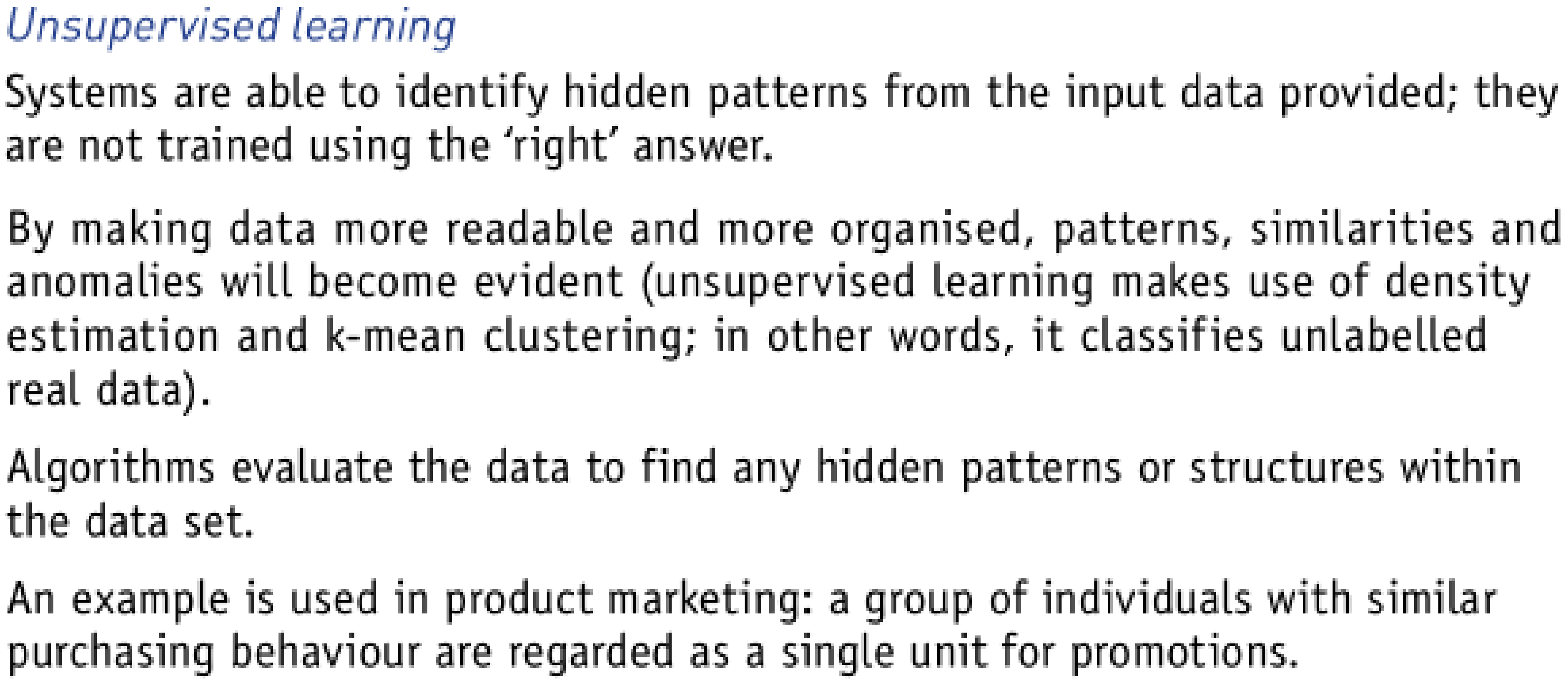


Supervised learning

Supervised learning makes use of regression analysis and classification analysis. It is used to predict future outcomes based on past data:

- The system requires both an input and an output to be given to the model so it can be trained.
- The model uses labelled data, so the desired output for a given input is known.
- Algorithms receive a set of inputs and the correct outputs to permit the learning process.
- Once trained, the model is run using labelled data.
- The results are compared with the expected output; if there are any errors, the model needs further refinement.
- The model is run with unlabelled data to predict the outcome.

An example of supervised learning is categorising emails as relevant or spam/ junk without human intervention.



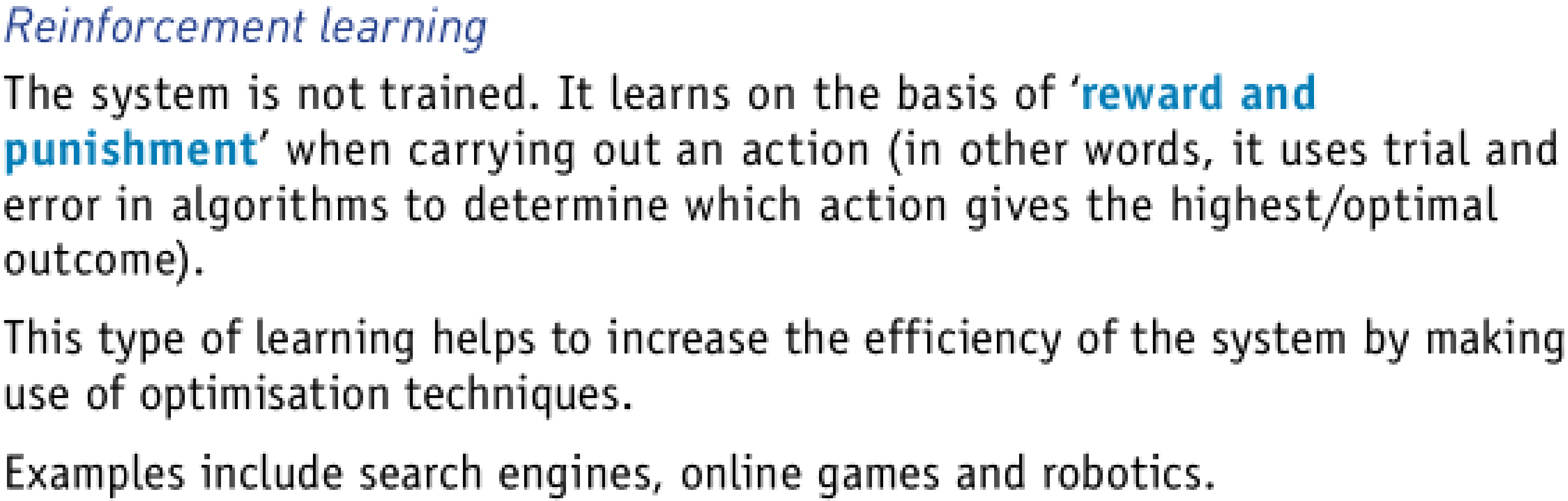
Unsupervised learning

Systems are able to identify hidden patterns from the input data provided; they are not trained using the 'right' answer.

By making data more readable and more organised, patterns, similarities and anomalies will become evident (unsupervised learning makes use of density estimation and k-mean clustering; in other words, it classifies unlabelled real data).

Algorithms evaluate the data to find any hidden patterns or structures within the data set.

An example is used in product marketing: a group of individuals with similar purchasing behaviour are regarded as a single unit for promotions.

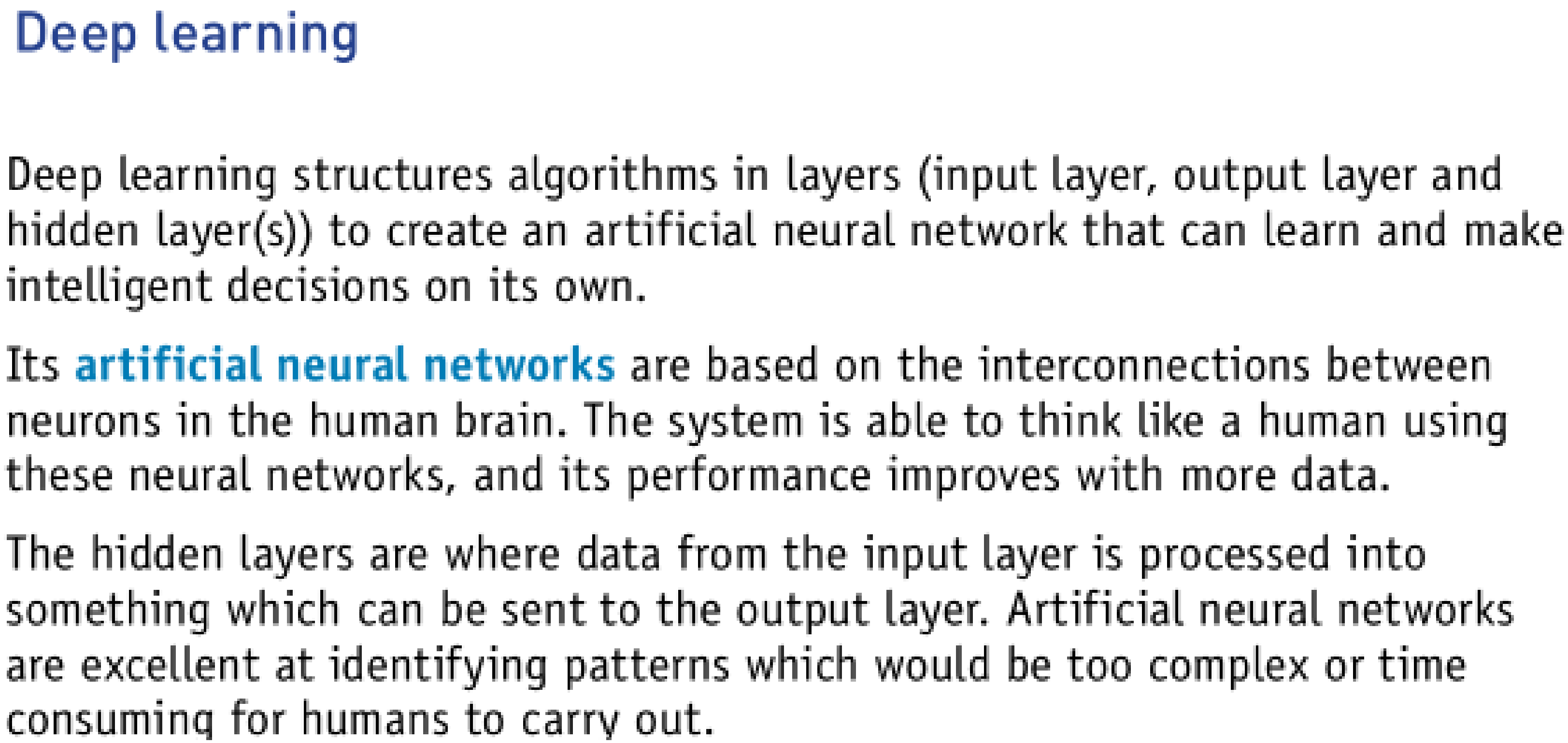


Reinforcement learning

The system is not trained. It learns on the basis of '**reward and punishment**' when carrying out an action (in other words, it uses trial and error in algorithms to determine which action gives the highest/optimal outcome).

This type of learning helps to increase the efficiency of the system by making use of optimisation techniques.

Examples include search engines, online games and robotics.



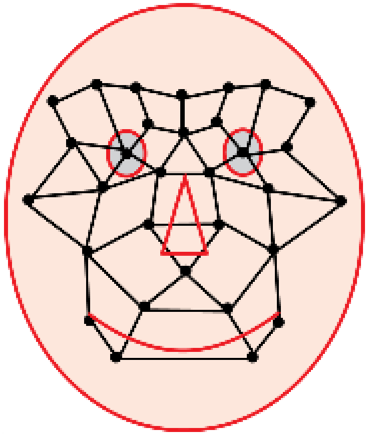
Deep learning

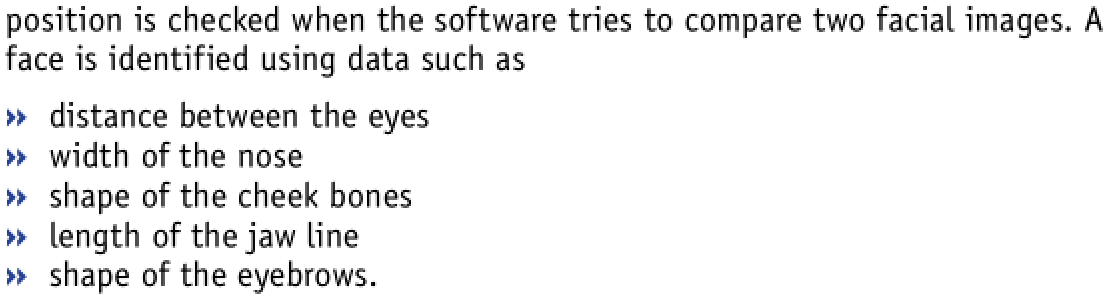
Deep learning structures algorithms in layers (input layer, output layer and hidden layer(s)) to create an artificial neural network that can learn and make intelligent decisions on its own.

Its **artificial neural networks** are based on the interconnections between neurons in the human brain. The system is able to think like a human using these neural networks, and its performance improves with more data.

The hidden layers are where data from the input layer is processed into something which can be sent to the output layer. Artificial neural networks are excellent at identifying patterns which would be too complex or time consuming for humans to carry out.

For example, they can be used in face recognition. The face in Figure shows several of the positions used by the face recognition software.

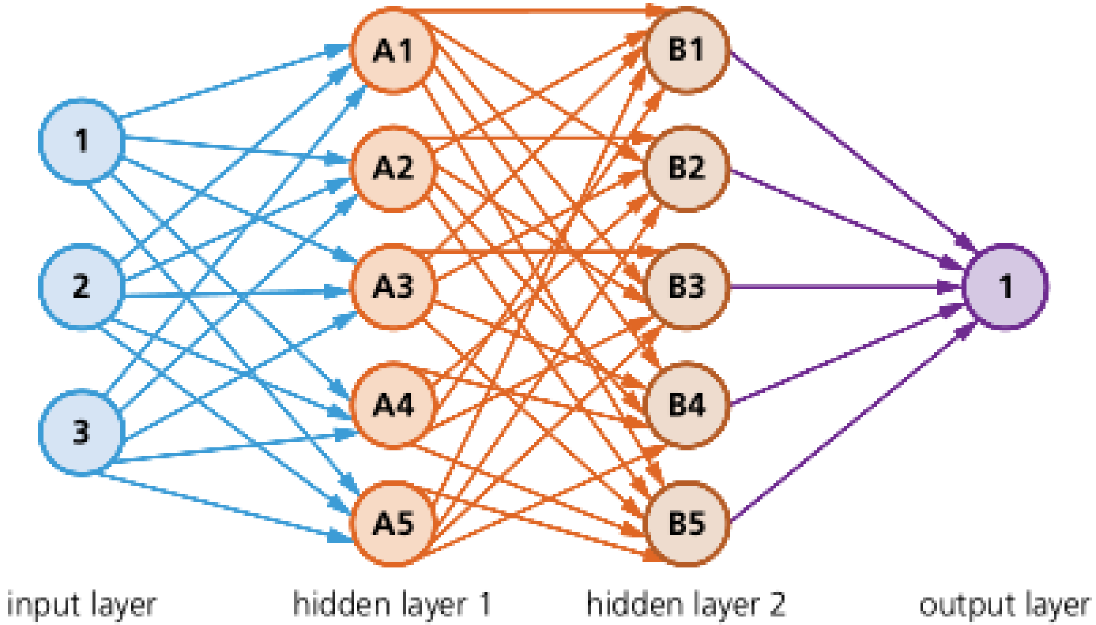




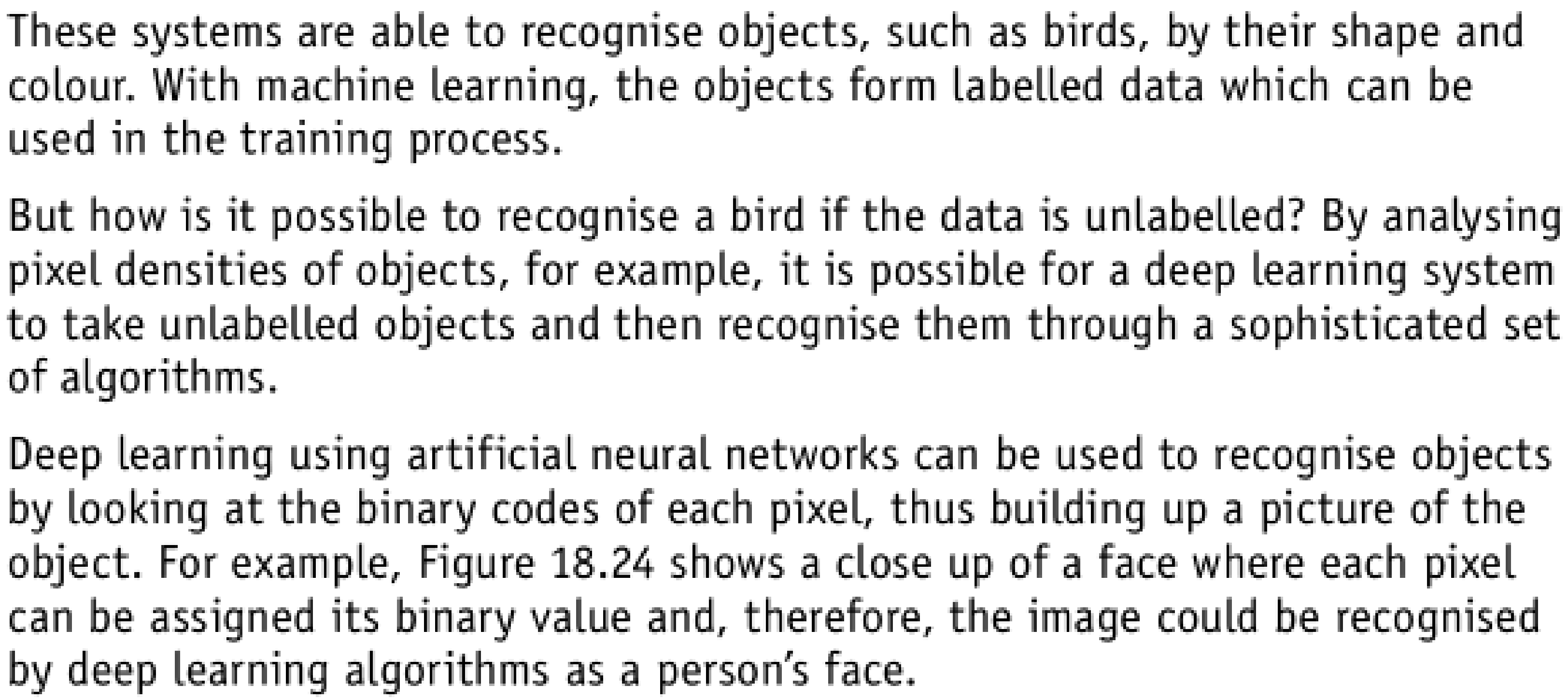
position is checked when the software tries to compare two facial images. A face is identified using data such as

- distance between the eyes
- width of the nose
- shape of the cheek bones
- length of the jaw line
- shape of the eyebrows.

Figure 18.23 shows an artificial neural network (with two hidden layers).



▲ **Figure 18.23 An artificial neural network**

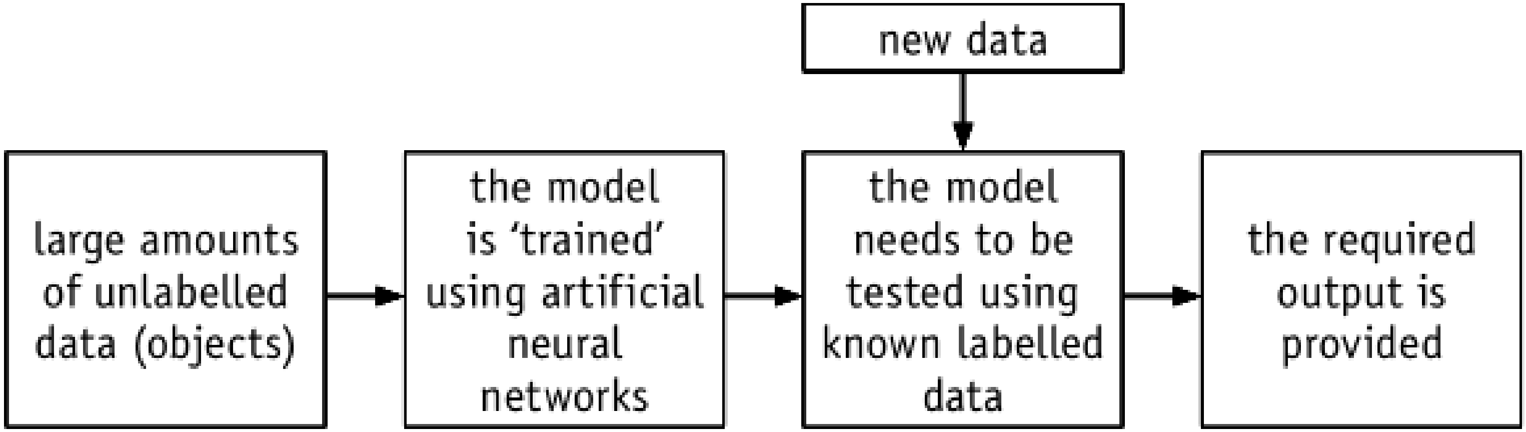


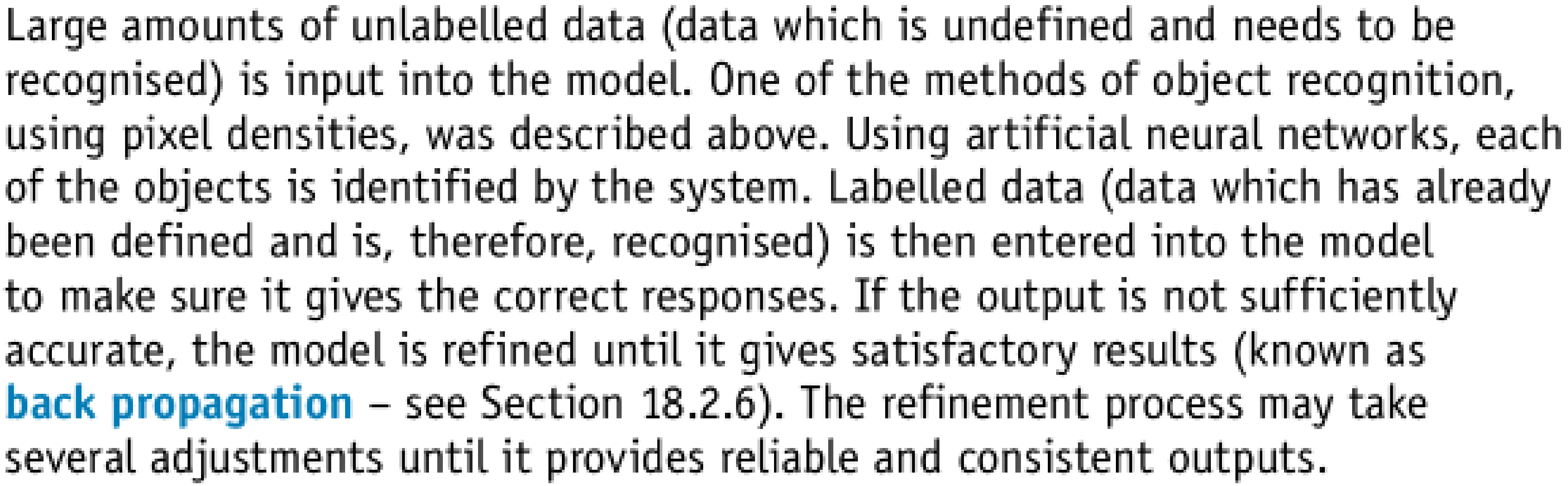
These systems are able to recognise objects, such as birds, by their shape and colour. With machine learning, the objects form labelled data which can be used in the training process.

But how is it possible to recognise a bird if the data is unlabelled? By analysing pixel densities of objects, for example, it is possible for a deep learning system to take unlabelled objects and then recognise them through a sophisticated set of algorithms.

Deep learning using artificial neural networks can be used to recognise objects by looking at the binary codes of each pixel, thus building up a picture of the object. For example, Figure 18.24 shows a close up of a face where each pixel can be assigned its binary value and, therefore, the image could be recognised by deep learning algorithms as a person's face.

This summarises how deep learning works:

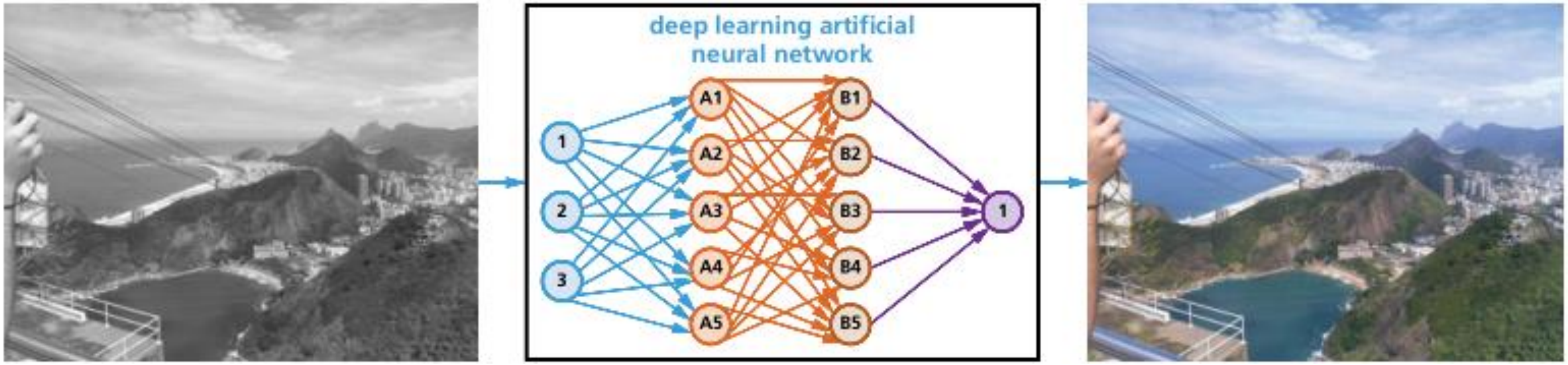

new data
large amounts of unlabelled data (objects)
the model is 'trained' using artificial neural networks
the model needs to be tested using known labelled data
the required output is provided



Large amounts of unlabelled data (data which is undefined and needs to be recognised) is input into the model. One of the methods of object recognition, using pixel densities, was described above. Using artificial neural networks, each of the objects is identified by the system. Labelled data (data which has already been defined and is, therefore, recognised) is then entered into the model to make sure it gives the correct responses. If the output is not sufficiently accurate, the model is refined until it gives satisfactory results (known as **back propagation** – see Section 18.2.6). The refinement process may take several adjustments until it provides reliable and consistent outputs.

Turning monochrome photos into colour

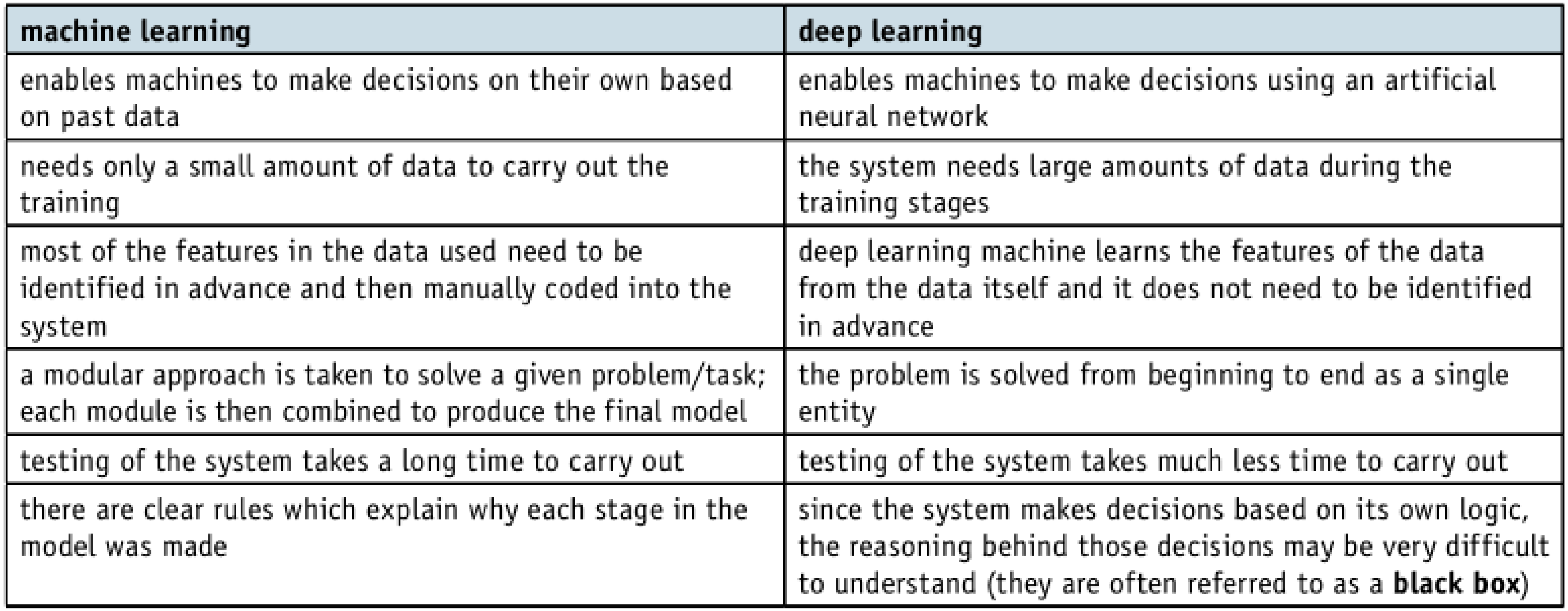
Deep learning can be used to turn monochrome (black and white) photographs into coloured photographs. This is a sophisticated system which produces a better photograph than simply turning grey-scale values into an approximated colour. In Figure 18.28, the original monochrome image has been processed to give a very accurate coloured image.



Deep learning can change black and white photographs to colour

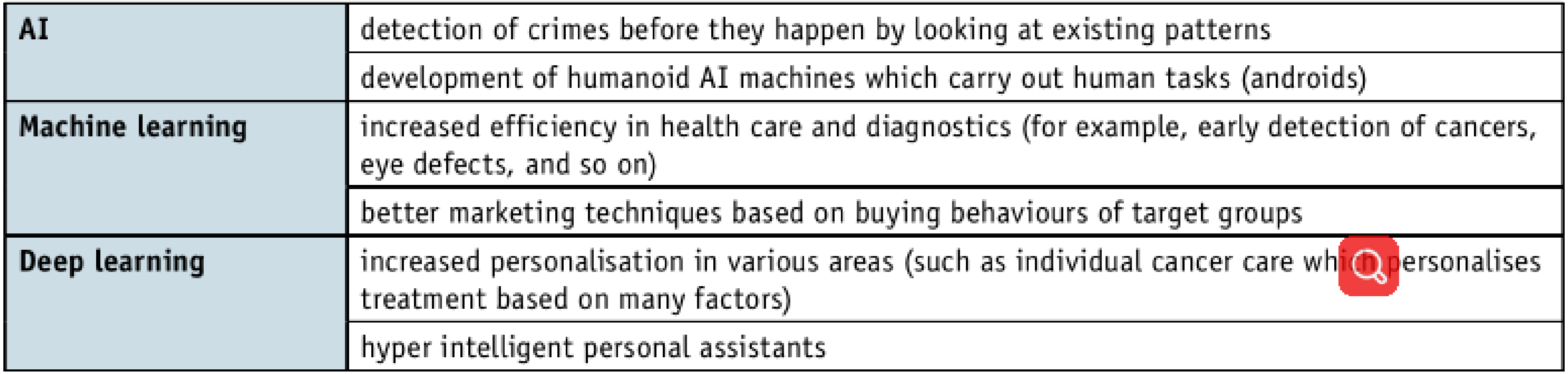
The deep learning system is trained by searching websites for data which allows it to recognise features and then map a particular colour to a photograph/ object thus producing an accurate coloured image.

Comparison between machine learning and deep learning



machine learning	**deep learning**
enables machines to make decisions on their own based on past data	enables machines to make decisions using an artificial neural network
needs only a small amount of data to carry out the training	the system needs large amounts of data during the training stages
most of the features in the data used need to be identified in advance and then manually coded into the system	deep learning machine learns the features of the data from the data itself and it does not need to be identified in advance
a modular approach is taken to solve a given problem/task; each module is then combined to produce the final model	the problem is solved from beginning to end as a single entity
testing of the system takes a long time to carry out	testing of the system takes much less time to carry out
there are clear rules which explain why each stage in the model was made	since the system makes decisions based on its own logic, the reasoning behind those decisions may be very difficult to understand (they are often referred to as a **black box**)

What will happen in the future?

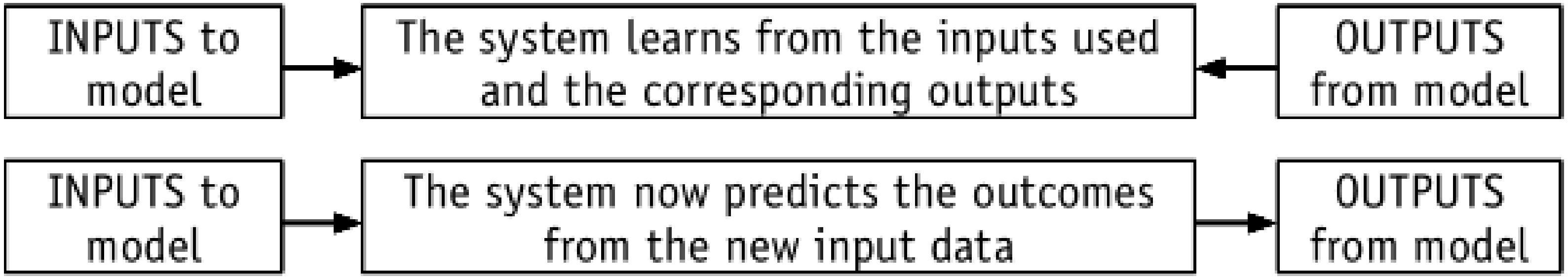


AI	detection of crimes before they happen by looking at existing patterns
	development of humanoid AI machines which carry out human tasks (androids)
Machine learning	increased efficiency in health care and diagnostics (for example, early detection of cancers, eye defects, and so on)
	better marketing techniques based on buying behaviours of target groups
Deep learning	increased personalisation in various areas (such as individual cancer care which personalises treatment based on many factors)
	hyper intelligent personal assistants

Back propagation and regression methods

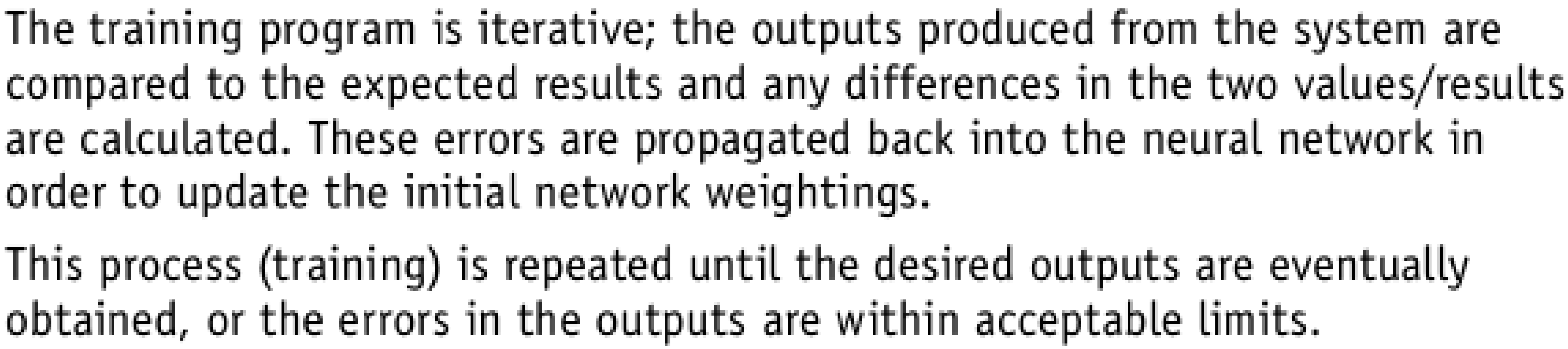
Back propagation

When designing neural networks, it is necessary to give random weightings to each of the neural connections. However, the system designer will not initially know which weight factors to use to produce the best results. It is necessary to train the neural networks during the development stage:



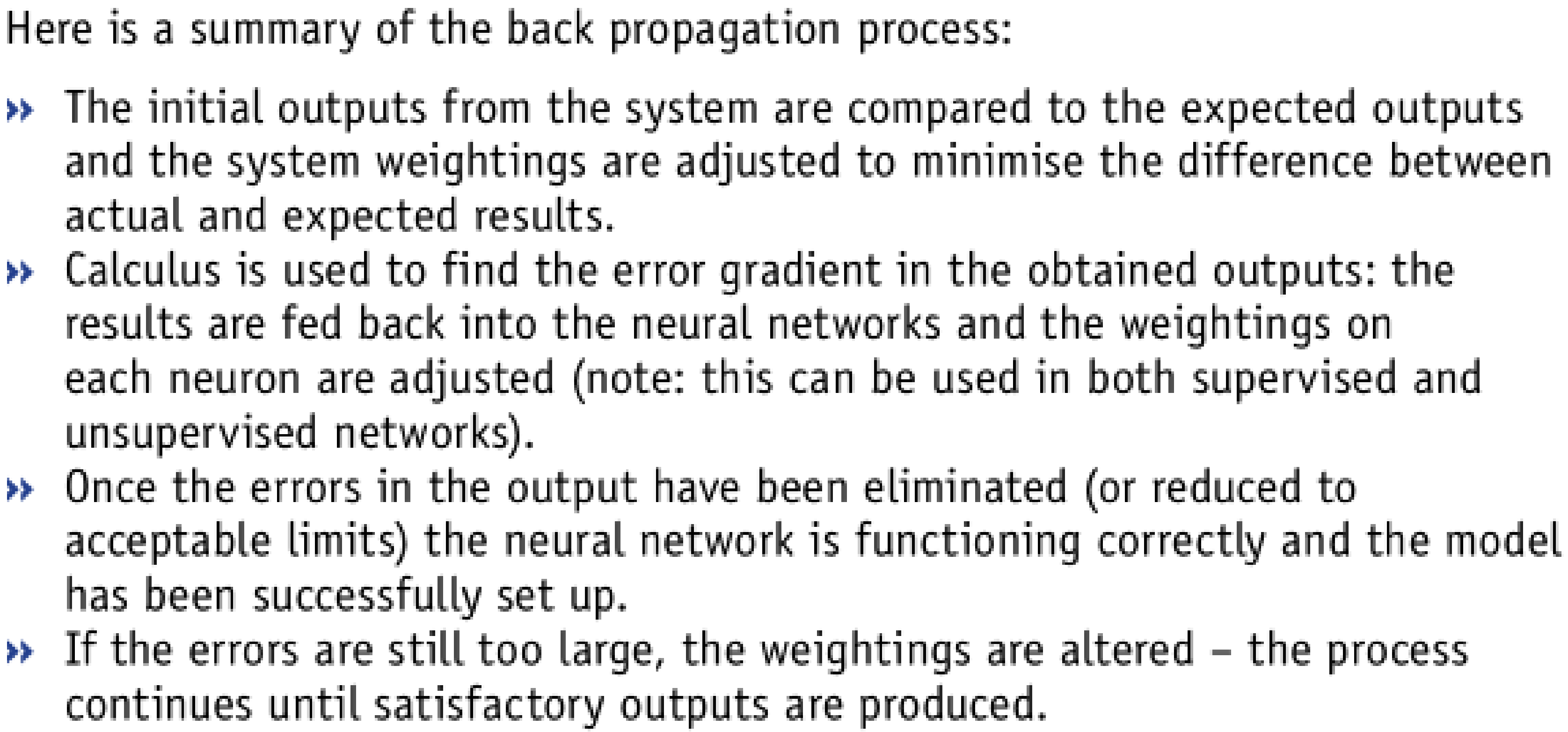
INPUTS to model → The system learns from the inputs used and the corresponding outputs ← OUTPUTS from model

INPUTS to model → The system now predicts the outcomes from the new input data → OUTPUTS from model



The training program is iterative; the outputs produced from the system are compared to the expected results and any differences in the two values/results are calculated. These errors are propagated back into the neural network in order to update the initial network weightings.

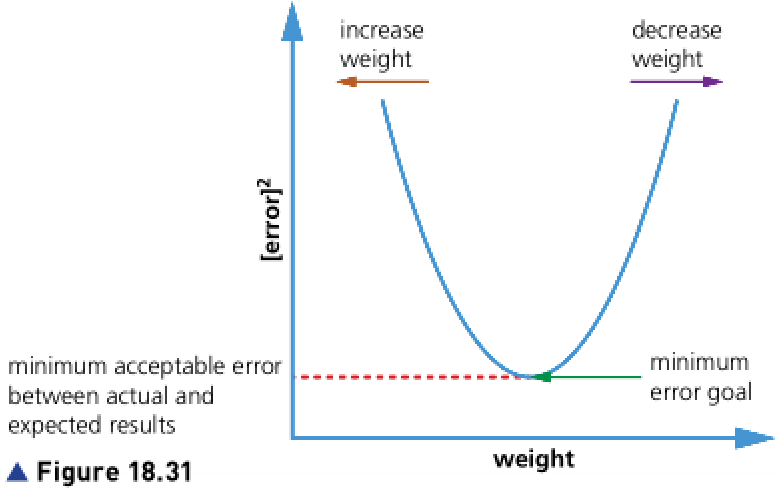
This process (training) is repeated until the desired outputs are eventually obtained, or the errors in the outputs are within acceptable limits.



Here is a summary of the back propagation process:

- The initial outputs from the system are compared to the expected outputs and the system weightings are adjusted to minimise the difference between actual and expected results.
- Calculus is used to find the error gradient in the obtained outputs: the results are fed back into the neural networks and the weightings on each neuron are adjusted (note: this can be used in both supervised and unsupervised networks).
- Once the errors in the output have been eliminated (or reduced to acceptable limits) the neural network is functioning correctly and the model has been successfully set up.
- If the errors are still too large, the weightings are altered – the process continues until satisfactory outputs are produced.

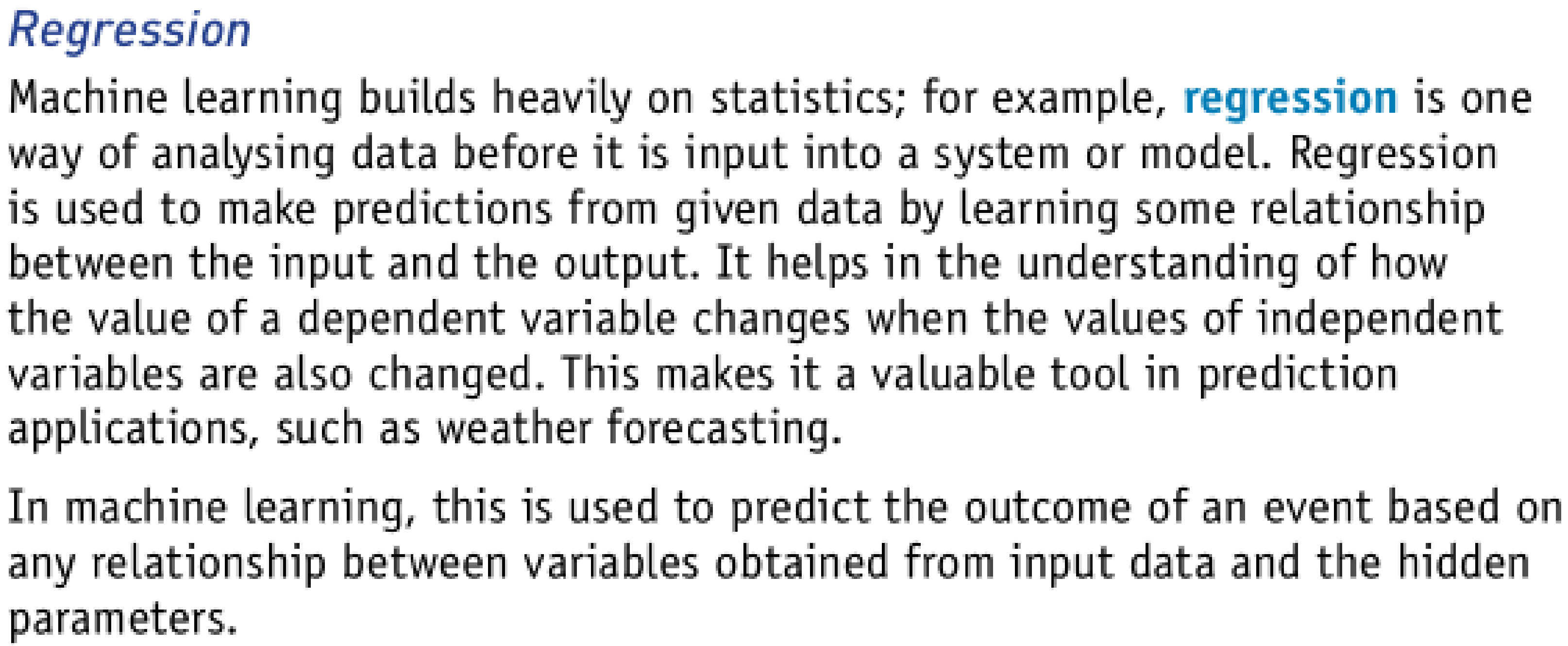
Figure 18.31 shows the ultimate goal of the back propagation process.



▲ **Figure 18.31**

There are two types of back propagation: static and recurrent:

- Static maps static inputs to a static output.
- Mapping is instantaneous in static, but this is not the case with recurrent.
- Training a network/model is more difficult with recurrent than with static.
- With recurrent, activation is fed forward until a fixed value is achieved.



Regression

Machine learning builds heavily on statistics; for example, **regression** is one way of analysing data before it is input into a system or model. Regression is used to make predictions from given data by learning some relationship between the input and the output. It helps in the understanding of how the value of a dependent variable changes when the values of independent variables are also changed. This makes it a valuable tool in prediction applications, such as weather forecasting.

In machine learning, this is used to predict the outcome of an event based on any relationship between variables obtained from input data and the hidden parameters.

Artificial neural networks

- Creating a system that functions along similar lines to the way that the human brain works
- Nodes represent artificial neurons
- Nodes are arranged in layers
- A node can adjust the weighting applied to its inputs
- Back propagation of errors is used
- A deep learning system requires many layers