Syllabus Content:

18.1 Artificial intelligence (OS)

- Show understanding of how graphs can be used to aid Artificial Intelligence (AI).

  Notes and guidance
  - Purpose and structure of a graph.
  - Use of **Dijkstra's algorithm** to perform searches on a graph.
  - Use **A* algorithm** to perform searches on a graph.
  - Candidates will not be required to write algorithms to set up, access, or perform searches on graphs
- Show understanding of how artificial neural networks have helped with machine learning
- Show understanding of Deep Learning, Machine Learning and Reinforcement Learning and the reasons for using these methods.

  Notes and guidance
  - Understand machine learning categories, including **supervised learning, unsupervised learning**
- Show understanding of back propagation of errors and regression methods in machine learning

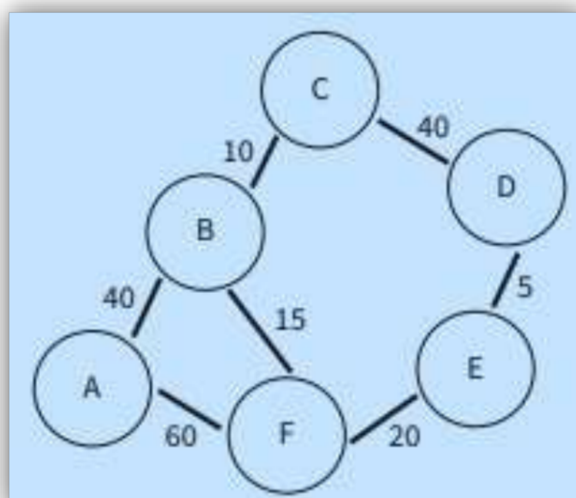
What is Artificial Intelligence?

Artificial Intelligence is concerned with "how to make computers do things at which, at the moment, people are better." (E. Rich. Artificial Intelligence. McGraw-Hill, 1983)

Artificial intelligence (AI), the ability of a **digital computer** or **computer-controlled robot** to perform tasks commonly associated with **intelligent beings**. The term is frequently applied to the project of developing systems endowed with the **intellectual processes characteristic of humans**, such as the ability to reason, discover meaning, generalize, or learn from past experience.

Purpose and structure of graphs:

A graph is a collection of **nodes** or **vertices** between which there can be edges. Each **node** has a name. An edge can have an associated label which is a numerical value. An example is presented



A graph can be used to represent a variety of scenarios. One common representation is that the nodes represent places and the edge labels represent the distances between those places.

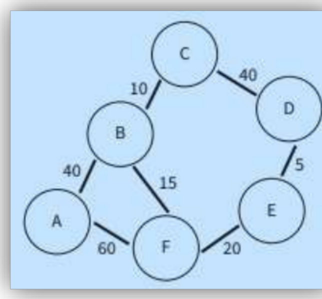
Edges are only included in the graph when there is a route available for direct travel between the pair of nodes. Such graphs can, for example, find the shortest route between two nodes that are not adjacent to each other.

We could use our intelligence to find the shortest route between node A and node D by considering all of the possible routes and calculating the overall distance for each route.

For A to B to C to D overall distance is 40 + 10 + 40 = 90

For A to B to F to E to D overall distance is 40 + 15 + 20 + 5 = 80, which is the shortest

For A to F to E to D overall distance is **60 + 20 + 5 = 85**
For A to F to B to C to D overall distance is **60 + 15 + 10 + 40 = 125**

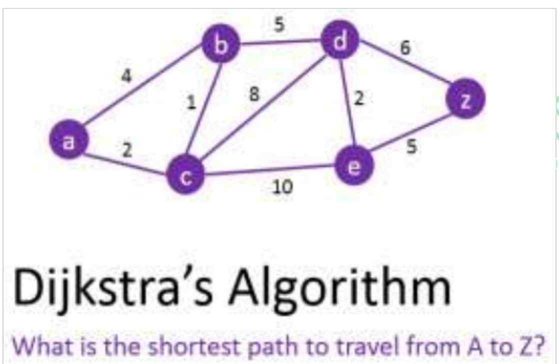


For a graph containing 100 nodes this could be quite time consuming.

Fortunately, a number of artificial intelligence algorithms have been developed to solve this type of problem.

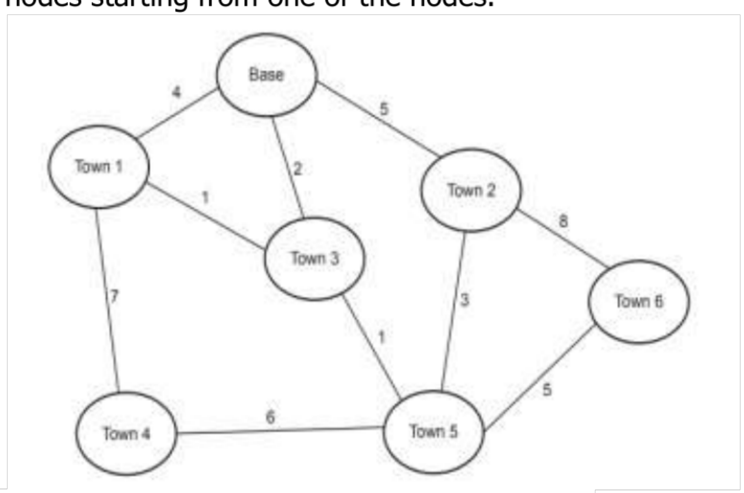
Dijkstra's Algorithm:

(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 algorithm finds the shortest path to each of the other nodes starting from one of the nodes.



9618/32/M/J/21

Q5 (a) Calculate the shortest distance between the base and each of the other towns in the diagram using **Dijkstra's algorithm.**



Show your working and write your answers in the table provided

Working.......................

Answers

Town 1	Town 2	Town 3	Town 4	Town 5	Town 6

[5]

Question	Answer	Marks
5(a)	Working **(Max 3)** May be seen on diagram • Initialisation: setting Base to 0 • … and the rest of the towns to ∞ • Evidence to show values at nodes being updated • Evidence to show 'visited node(s)' May be seen in working section of paper • Evidence to show calculation of at least one route • Evidence to show more than one route has been calculated for at least one town **Correct Answer (Max 2)** **One mark** for four correct values… … **One mark** for all values correct	**5**

Town 1	Town 2	Town 3	Town 4	Town 5	Town 6
3	5	2	9	3	8

What is A* Algorithm:

A * algorithm is a searching algorithm that searches for the shortest path between the ***initial and the final state.*** It is used in various applications, such as ***maps***.

In ***maps*** the A* algorithm is used to calculate the shortest distance between the **source (initial state)** and the **destination (final state)**

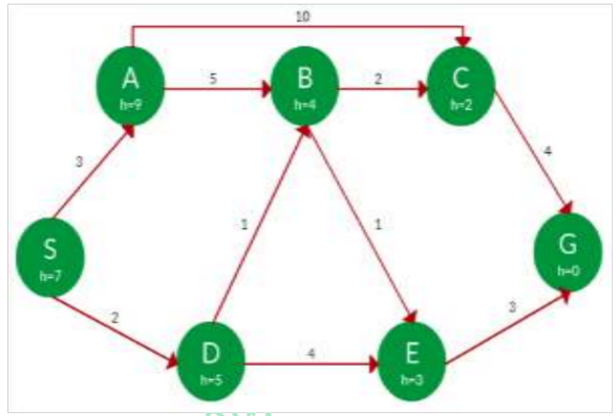
How it works:

A* algorithm has 3 parameters:

- **(g):** is the cost of moving the initial cell to the current cell. Basically it's the sum of all the cells that have been visited since leaving the first cell.
- **(h): heuristic value**: is the **estimated** cost of moving from the current cell to the final cell. The actual cost cannot be calculated until the final cell is reached. Hence **h** is the estimated cost. Many books have many different methods to find out heuristic value.
  We must make sure that there is never an over estimation of the cost.
    - There are different ways to find (h)
        - Manhattan method (values of differences in the **goal's x and y coordinates**)
        - Straight-line heuristic using (Longitude and Latitude)
        - Landmark heuristic etc

Heuristic value (h) is normally given in exam questions to calculate shortest path.

- (**f**): is the parameter used to find the least coast from one node to another.
- **{ f = g + h }**
  Responsible to find the optimal path between source and destination.
- The way algorithm makes its decisions is by taking the f value into account. The algorithm selects smallest f-values cells and moves to that cells. The process continues until algorithm reaches its goal cell.



Example:

Question: Find the path to reach from S to G using A* Algorithm:

- Let's see how calculation works.
- **S = source** and **G = destination**
- Initially the cost of **source & destination is always 0**, so **S = 0 & G = 0**.
- At every step **f** is re-calculated by adding together **g** and **h**
- Suppose we keep heuristic value **h = 9**
- For source **S** using formula **f = g + h is**

  f = 0 +9 = 9

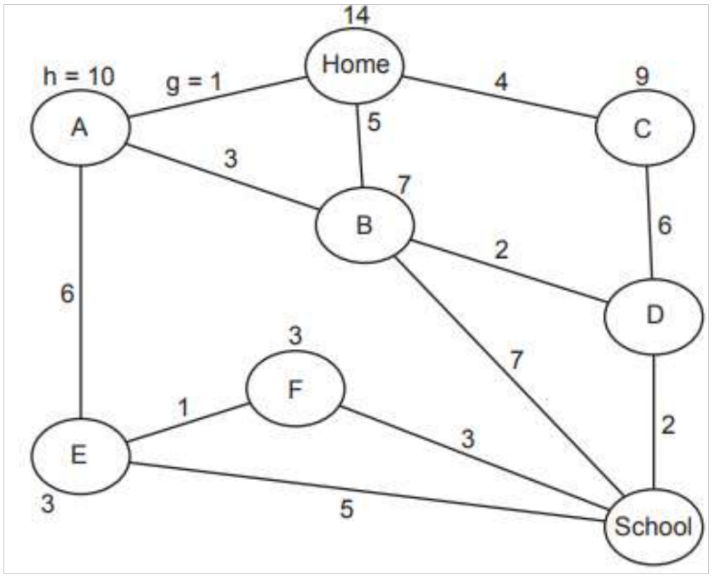
Path		h(x)	g(x)	f(x)
S		7	0	7
S -> A		9	3	12
S -> D	✓	5	2	7
S -> D -> B	✓	4	2 + 1 = 3	7
S -> D -> E		3	2 + 4 = 6	9
S -> D -> B -> C	✓	2	3 + 2 = 5	7
S -> D -> B -> E	✓	3	3 + 1 = 4	7
S -> D -> B -> C -> G		0	5 + 4 = 9	9
S -> D -> B -> E -> G	✓	0	4 + 3 = 7	7

- The path from S g(x) + h(x) = f(x) so it bcomes **f = g + h is f = 0 +7 = 7**
- S connects to **A and D**
- The cost from **S to A** is **f(x) = g(x) + h(x) is f = 3 +9 = 12**
- The coast from **S to D** is **f = g + h is f = 2 +5 = 7 we select S to D**
- Similarly we keep on finding path using the formula and select the value which is less than other options.
- So the final path becomes

Path: S -> D -> B -> E -> G

Let's take an example from Past paper (9618/32/O/N/21):

Q9 (b) Find the shortest path between the Home and School nodes using the A* algorithm. Show your working in the table provided. The first two rows in the table have been completed



Node	Cost from Home node (g)	Heuristic (h)	Total (f = g + h)
Home	0	14	14
A	1	10	11

Final path	

[5]

- Cost of Home node (g) = 0 and Heuristic value is given (h) = 14. Total f= g + h = 14
- From Home to A (**g = 1**) and (**h = 10**), so (**f = 1+10 = 11**)
- Now you see that Home is connected to nodes A, B and C
- So in B's case g=5 and h = 7 so **f = 5+7 = 12 (bigger than A so we hold it)**
- Now in C's case g=4 and h=9 so **f = 4+ 9 = 13 (bigger than A, so we hold it)**
- We take smallest route**: Home to A.**
- Now **A** connects to **B** and **E**
- In **B's case g(1+3 = 4) + h(7) = 11, so f = 4 + 7 = 11**
- In **E's case g(1+6 = 7) + h(3) = 11, so f = 7 + 3 = 10 (E < B so we take E)**
- Now **E** connects **F** and **School**
- In **F's case g(7+1=8) + h(3) = 11, so f = 8 + 3 = 11 (F has smaller value so we take F)**
- **E** to **Schools** case **g(7+5=12) + h(0) = 12, so f = 12 + 0 = 12 (School > F so we hold)**

So final path becomes:

Final Path	Home ⇨ A ⇨ E ⇨ F ⇨ School

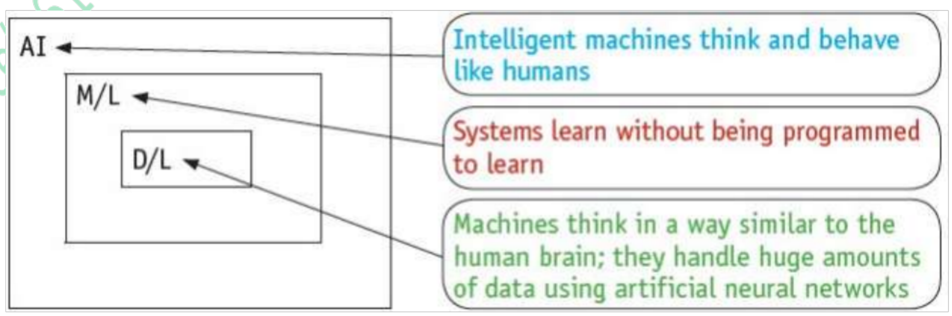
Below is answer from marking scheme:

	Node	Cost from Home Node (g)	Heuristic (h)	Total (f = g + h)
1	*Home*	*0*	*14*	*14*
2	*A*	*1*	*10*	*11*
3	B	5	7	12
4	C	4	9	13
5	B	1 + 3 = 4	7	11
6	E	1 + 6 = 7	3	10
7	F	7 + 1 = 8	3	11
8	School	7 + 5 = 12	0	12
9	School	8 + 3 = 11	0	11

Final Path	Home ⇨ A ⇨ E ⇨ F ⇨ School

Machine Learning:

Figure below 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.



AI can be thought of as a machine with **cognitive abilities** such as **problem solving** and **learning from examples**.

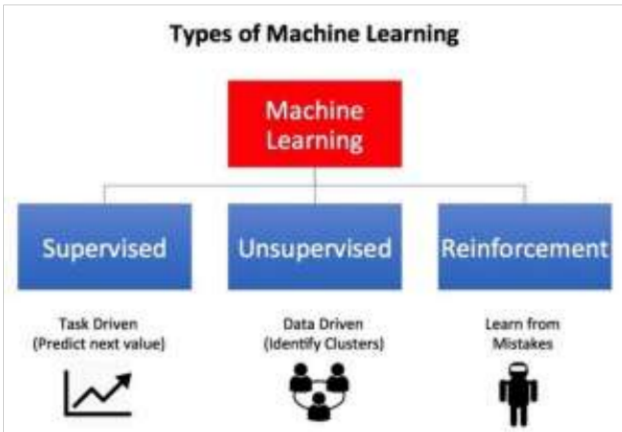
All of these can be measured against human benchmarks such as reasoning, speech and sight. AI is often split into three categories.

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.

The requirements for machine learning can be summarized as:

- a computer-based system has a defined task or tasks to perform
- knowledge is acquired through the experience of performing the tasks
- as a result of this experience and the knowledge gained the performance of future tasks is improved.

The ability to learn from experience is an indication of intelligence. Machine learning is therefore one of the many individual approaches defined under the umbrella of artificial intelligence. There are a number of ways to describe how the learning can take place.



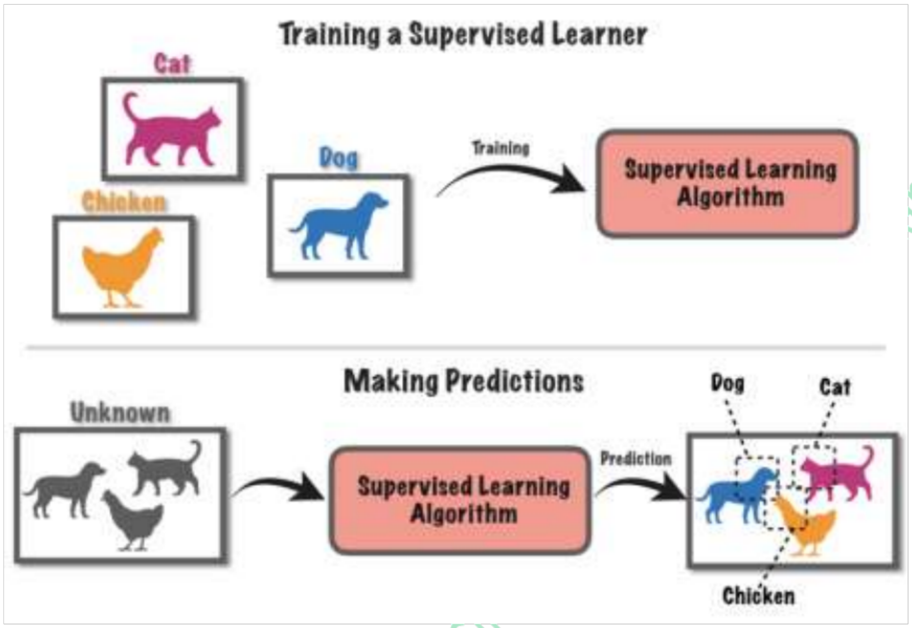
Three of these will be considered here:

1- Supervised learning
2- Unsupervised learning
3- Reinforcement learning

Supervised learning

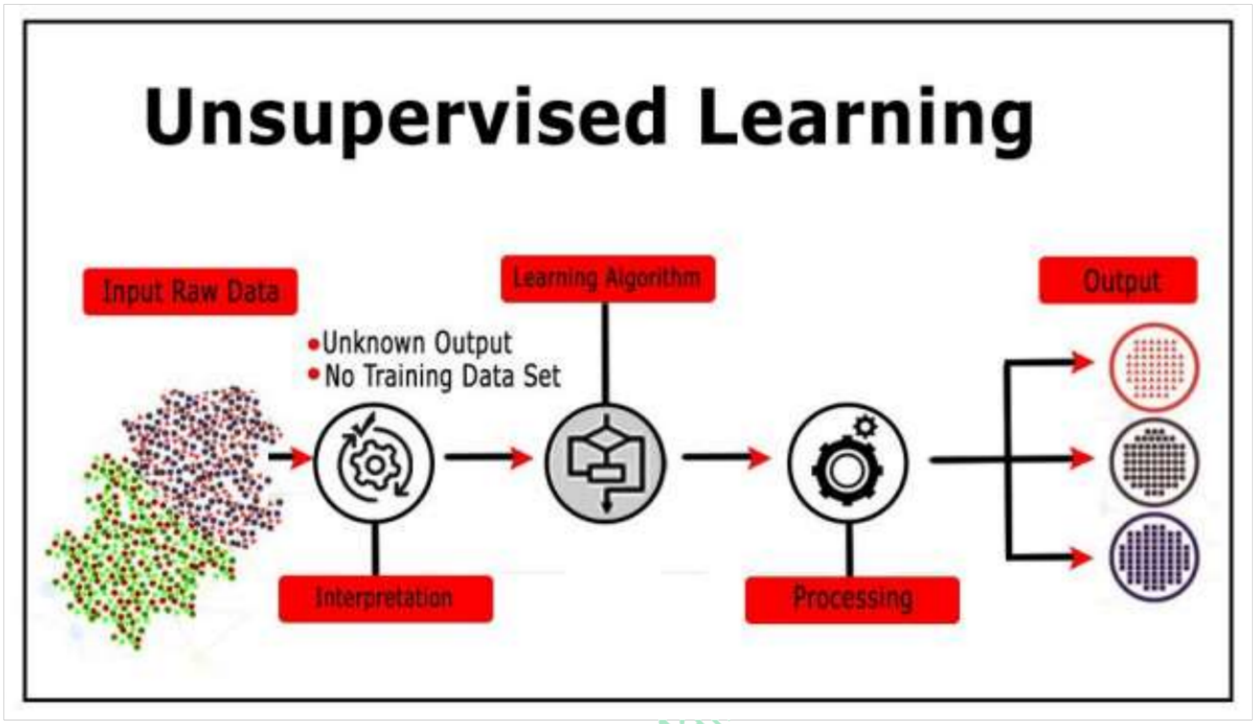
- The system is fed knowledge with associated classification.
- For example, an AI program might be under development for marking exam paper questions. In the supervised learning, answers to examination questions could be provided together with a grade for each one or with categorised comments.
- A special case of supervised learning is where an expert system is being developed.
- An expert system always has a focus on a narrowly defined domain of knowledge. In this case human experts are given samples of data requiring analysis.
- The experts provide the conclusions to be drawn from the data. The data and conclusions are input to the knowledge base.
- The effectiveness of the system can be tested by a human expert providing sample data and checking the accuracy of the conclusions provided by the expert system.

- If performance is poor, then further data and conclusions are input to the system. Although an expert system is an example of AI it is not an example of machine learning because there is no expectation that the system will improve its performance unaided.



Unsupervised learning

- The system has to draw its own conclusions from its experience of the results of the tasks it has performed. For this, algorithms are needed that can organise or categorise the knowledge acquired.
- An example is where 'conceptual clusters' are identified which are based on a hierarchical framework.
- In this approach the knowledge is initially all placed in the root of a tree structure.
- Then, depending on attributes of the knowledge, selected groups are moved into branches of the tree.
- Nowadays unsupervised learning is a dominant activity. Powerful computer systems having access to massive data banks are regularly used to make decisions based on previous actions recorded.
- We all have our activity on the **world wide web recorded and stored.**
- This stored data is then used to make decisions about what products or services should be recommended to us in future Internet use.
- There is no theoretical framework for this; it is rather as though the intelligence is built-in to the data.



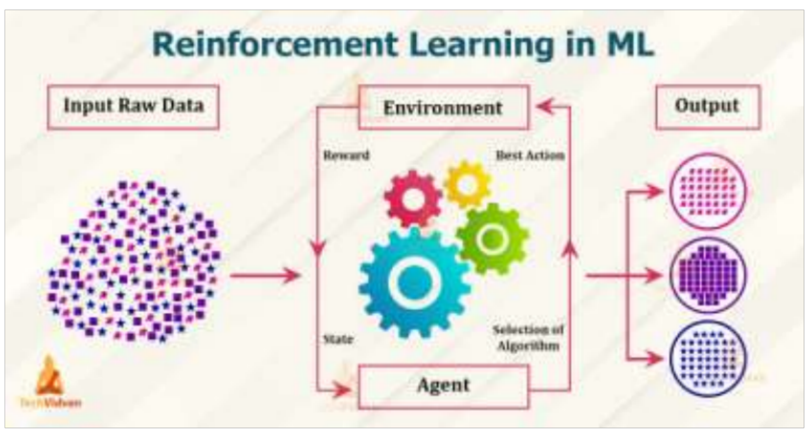
Systems are able to identify hidden patterns from the input data provided; they are not trained using the 'right' answer.

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:

Reinforcement learning has some features similar to **unsupervised learning** and other features similar to **supervised learning**.
The method has its own specific vocabulary.



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.**

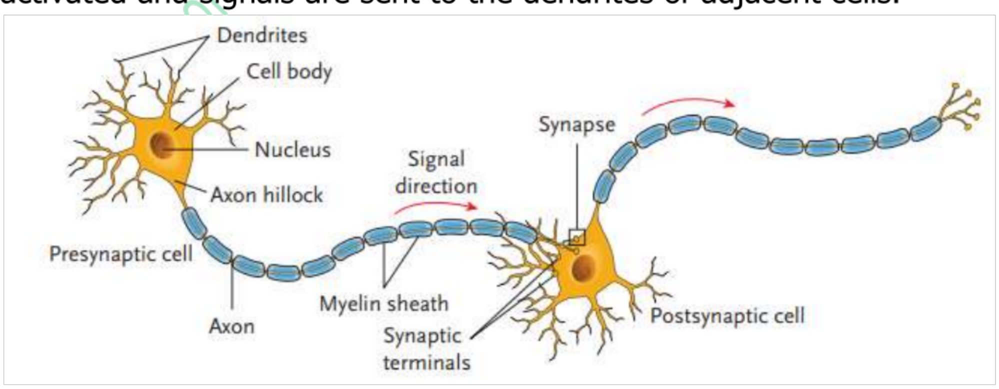
The following statements use this vocabulary in describing aspects of how a reinforcement learning algorithm works.

- An agent is learning how best to perform in an environment.
- The environment has many defined states.
- At each step the agent takes an action.
- An agent has a policy that guides its actions.
- The policy is influenced by the recorded history and the knowledge of the current state of the environment.
- An action changes the environment to a new state.
- The agent receives a reward following an action which is a measure of how effective the action was in relation to the achievement of the overall goal.
- The policy will guide the agent in deciding whether the next action should be exploiting knowledge already known or exploring a new avenue.

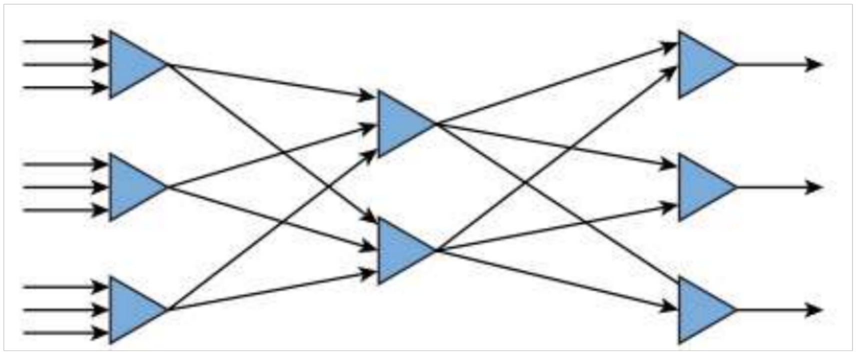
Artificial neural networks

The neural networks in our brains provide our intelligence. It therefore seems obvious that artificial neural networks should be considered as a foundation for artificial intelligence systems. **Figure below** shows a representation of two nerve cells.

At the one end of a nerve cell there are many dendrites which can receive signals. At the other end of the cell there are many axon terminal buttons that can transmit signals. The synapse is the region between an axon terminal button and a dendrite which contains neurotransmitters. When a nerve cell receives input signals the voltage in the axon increases. At some threshold value of this voltage neurotransmitters are activated and signals are sent to the dendrites of adjacent cells.



An artificial neural network could be created in software or hardware. The components of the network can be represented by a diagram as illustrated in **Figure below**.



The triangles are the nodes in the network which represent **artificial neurons.** (Sometimes these are represented as circles).

In general, a node can **receive one** or **more inputs** and can provide an output to one or more of the other nodes.

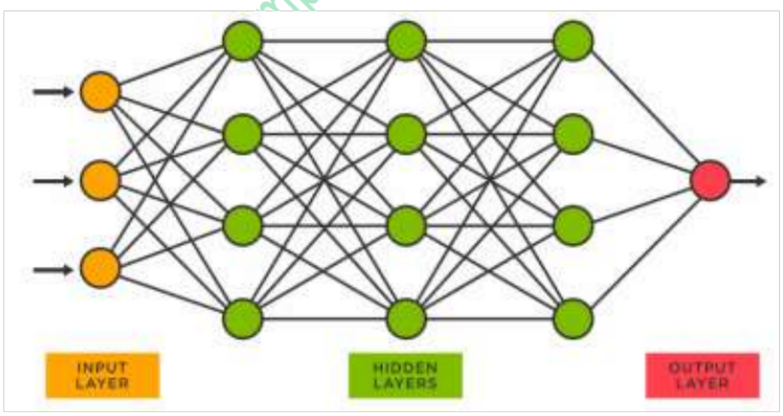
The modelling of the action of the node involves applying a weighting factor to each input. The weighted input values are summed and then an activation function is used to compute a value for the output of the node. If the input is not a numerical value it must be converted to one so that the weighted values can be summed.

Figure below shows a very simple network structure consisting of three layers.

The column of three nodes on the **left receive input.**

The column on the **right provides output.**

The two nodes in between form what is referred to as **a hidden layer**.



Some artificial neural networks will contain several hidden layers

How Neural Networks Function Similar to the Human Brain

An artificial neural network in its most basic form has three layers of neurons. Information flows from one to the next, just as it does in the human brain:

- **The input layer:** the data's entry point into the system
- **The hidden layer:** where the information gets processed
- **The output layer:** where the system decides how to proceed based on the data

More complex artificial neural networks will have multiple layers, some hidden.

The neural network functions via a collection of nodes or connected units, just like artificial neurons.

These nodes **loosely model the neuron network** in the animal brain. Just like its biological counterpart, **an artificial neuron** receives a signal in the form of a **stimulus**, processes it, and signals other **neurons connected to it**.

But the similarities end there.

The Neuronal Workings Of An Artificial Neural Network

In an artificial neural network, the artificial neuron receives a stimulus in the form of a signal that is a real number. Then:

- The output of **each neuron is computed** by a **nonlinear function** of the **sum of its inputs**.
- The **connections** among the **neurons** are called **edges**.
- Both **neurons** and **edges** have a **weight.**
- This **parameter** adjusts and **changes as the learning proceeds**.
- The **weight increases** or **decreases** the strength of the signal at a **connection**.
- Neurons may have a **threshold**. A signal is sent onward only if the aggregate signal crosses this threshold.

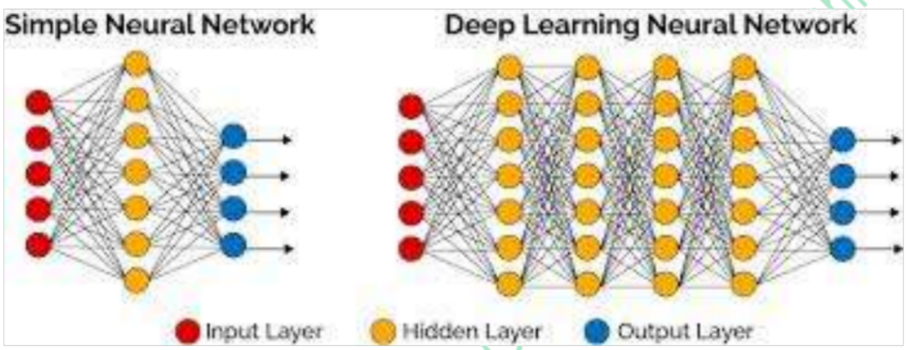
As mentioned earlier, neurons aggregate into layers. Different layers may perform different modifications on their inputs. Signals flit from the first layer (the input layer) to the last layer (the output layer) in the manner discussed above, sometimes after traversing the layers multiple times.

Neural networks inherently contain some **manner of a learning rule**, which modifies the **weights** of the **neural connections** in accordance with the input patterns they are presented with, just as a growing child learns to recognize animals from examples of animals.

Deep Learning

It is understood that in the brain there is a layer structure of neurons where **lower layers have readily understandable** functions but **where higher layers are concerned** with more **abstract processing** of information.
With the increasing computing power now available, artificial neural networks are being introduced with large numbers of **hidden layers** which are **attempting to achieve something similar.**

These are known as **Deep Learning systems.**



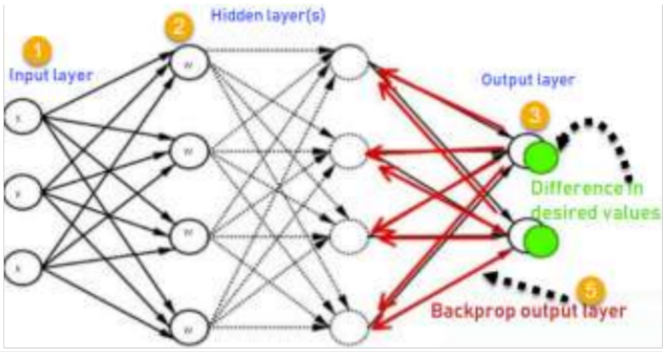
Backpropogation of Neural Network:

Backpropagation is the central mechanism by which **artificial neural networks** learn. It is the messenger telling the neural network whether or not it made a mistake when it made a prediction.

To propagate is to transmit something (light, sound, motion or information) in a particular direction or through a particular medium.

To backpropagate is to to transmit something in response, to send information back upstream – in this case, with the **purpose of correcting an error**.

When we discuss **backpropagation in deep learning,** we are talking about the transmission of information, and that information relates to the error produced by the neural network when it makes a **guess about data**. Backpropagation is synonymous with correction.



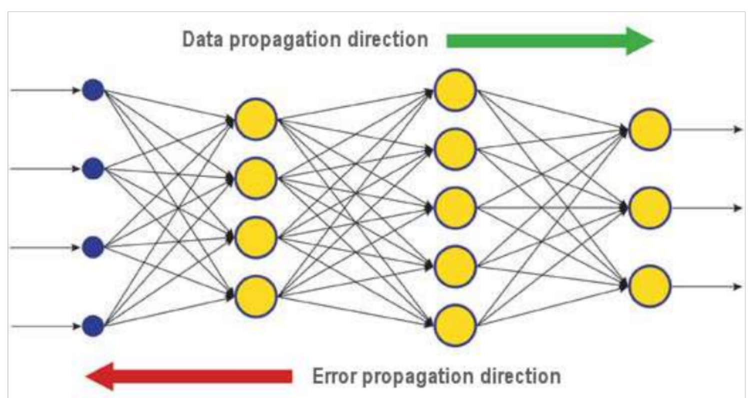
Untrained neural network models are like new-born babies: They are created **ignorant of the world**, and it is only through exposure to the world, **experiencing it**, that their ignorance is slowly relieved.

Algorithms experience the world **through data**. So by training a **neural network** on a relevant dataset, we seek to **decrease its ignorance**.

The way we measure progress is by monitoring the error produced by the network each time it makes a prediction. The way we achieve progress is by minimizing that error gradually in small steps. The errors are portals to discovery.

The knowledge of a neural network with regard to the world is captured by **its weights,** the **parameters that alter input data** as its **signal flows through** the neural network towards the **net's final layer**, which will make a decision about that input.

Those decisions are often wrong, because the parameters transforming the signal into a decision are **poorly calibrated;** they haven't **learned enough yet.**



Forward propagation is when a data instance sends its signal through a network's parameters toward the prediction at the end.

Once that prediction is made, **its distance from the ground truth (error) can be measured.**

So the parameters of the neural network **have a relationship with the error** the net produces, and when the **parameters change, the error does, too**.

We change the parameters using optimization algorithms. A very popular optimization method is called **gradient descent,** which is useful for finding the minimum of a function. We are seeking to minimize the error, which is also known as the ***loss function*** or the ***objective function***.

What are the Uses of Error Backpropagation?

Backpropagation is especially useful for deep neural networks working on error-prone projects, such as image or speech recognition. Taking advantage of the chain and power rules allows **backpropagation** to function with any number of outputs and better train all sorts of neural networks.

KEY TERM

Back propagation of errors: An algorithm for machine learning that optimises the values for parameters which are adjustable. It is applied first to the nodes in the output layer and then works backward through the nodes in hidden layers until finally the input nodes are considered.

Refrences:

- Cambridge Computer Science AS & A level by Sylvia Langfield and Dave Duddell
- Cambridge International AS & A level Computer Science by David Watson and Hellen Williams (Hodder Education)
- pastpapers
- https://www.educative.io/edpresso/what-is-the-a-star-algorithm
- https://www.geeksforgeeks.org/search-algorithms-in-ai/
- https://www.tibco.com/reference-center/what-is-a-neural-network
- https://wiki.pathmind.com/backpropagation