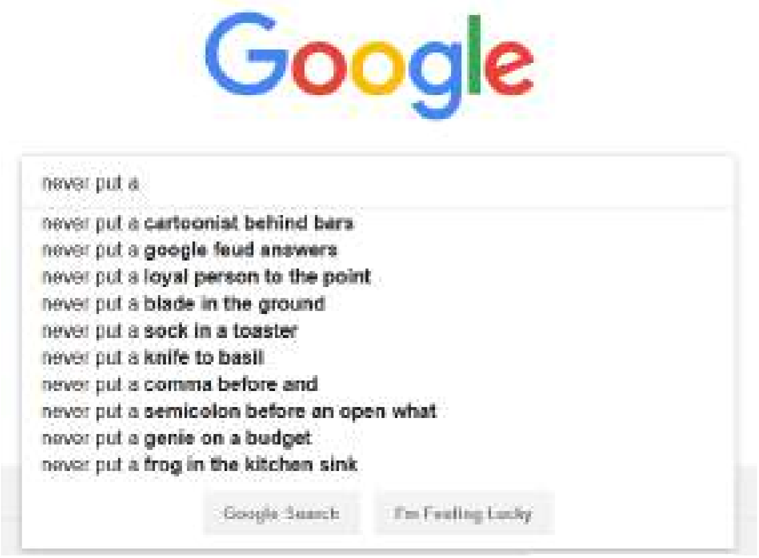
Natural Language Processing

Computers are great at working with standardized and structured data like database tables and financial records. They are able to process that data much faster than we humans can. But we humans don't communicate in "structured data" nor do we speak binary! We communicate using words, a form of *unstructured* data.

Unfortunately, computers suck at working with unstructured data because there are no standardized techniques to process it. When we program computers using something like C++, Java, or Python, we are essentially giving the computer a set of rules that it should operate by. With unstructured data, these rules are quite abstract and challenging to define concretely.



never put a

never put a **cartoonist behind bars**
never put a **google feud answers**
never put a **loyal person to the point**
never put a **blade in the ground**
never put a **sock in a toaster**
never put a **knife to basil**
never put a **comma before and**
never put a **semicolon before an open what**
never put a **genie on a budget**
never put a **frog in the kitchen sink**

There's a lot of unstructured natural language on the internet; sometimes even Google doesn't know what you're searching for!

Human vs Computer understanding of language

Humans have been writing things down for thousands of years. Over that time, our brain has gained a tremendous amount of experience in understanding natural language. When we read something written on a piece of paper or in a blog post on the internet, we *understand* what that thing really means in the real-world. We *feel* the emotions that reading that thing elicits and we often visualize how that thing would look in real life.

Natural Language Processing (*NLP)* is a sub-field of *Artificial Intelligence* that is focused on enabling computers to understand and process human languages, to get computers closer to a human-level understanding of language. Computers don't yet have the same intuitive understanding of natural language that humans do. They can't really understand what the language is *really* trying to say. In a nutshell, a computer *can't read between the lines*.

That being said, recent advances in Machine Learning (ML) have enabled computers to do quite a lot of useful things with natural language! Deep Learning has enabled us to write programs to perform things like language translation, semantic understanding, and text summarization. All of these things add real-world value, making it easy for you to understand and perform computations on large blocks of text without the manual effort.

Why Natural Language Processing?

- Classify text into categories
- Index and search large texts
- Automatic translation
- Speech understanding
    - Understand phone conversation
- Information extraction
    - Extract useful information from documents
- Automatic summarization
    - Question answering
- Text generations/dialogs

What is Natural Language Processing (NLP)?

Natural Language Processing (NLP) is the intersection of Computer Science, Linguistics and Machine Learning that is concerned with the communication between computers and humans in natural language. NLP is all about enabling computers to understand and generate human language. Applications of NLP techniques are Voice Assistants like Alexa and Siri but also things like Machine Translation and text-filtering. NLP is one of the fields that heavily benefited from the recent advances in Machine Learning, especially from Deep Learning techniques.

Components of NLP

There are three components of NLP:

1. **Speech Recognition**—The translation of spoken language into text.

2. **Natural Language Understanding (NLU)**
   Understanding involves the following tasks –

   - Mapping the given input in natural language into useful representations.
   - Analyzing different aspects of the language.

3. **Natural Language Generation (NLG)**

   It is the process of producing meaningful phrases and sentences in the form of natural language from some internal representation.

   It involves –

   - **Text planning** – It includes retrieving the relevant content from knowledge base.

- **Sentence planning** − It includes choosing required words, forming meaningful phrases, setting tone of the sentence.

- **Text Realization** − It is mapping sentence plan into sentence structure.

The NLU is harder than NLG.

Why NLP is difficult?

The process of reading and understanding language is far more complex than it seems at first glance. There are many things that go in to *truly* understanding what a piece of text means in the real-world. For example, what do you think the following piece of text means?

"Steph Curry was on fire last night. He totally destroyed the other team"

To a human it's probably quite obvious what this sentence means. We know Steph Curry is a basketball player; or even if you don't we know that he plays on some kind of team, probably a sports team. When we see "on fire" and "destroyed" we know that it means Steph Curry played really well last night and beat the other team.

Computers tend to take things a bit too literally. Viewing things literally like a computer, we would see "Steph Curry" and based on the capitalisation assume it's a person, place, or otherwise important thing which is great! But then we see that Steph Curry "was on fire"…. A computer might tell you that someone literally lit Steph Curry on fire yesterday! … Yikes. After that, the computer might say that Mr. Curry has physically destroyed the other team…. they no longer exist according to this computer… great…

Difficulties in NLU

NL has an extremely rich form and structure.

It is very ambiguous. There can be different levels of ambiguity −

- **Lexical ambiguity** − It is at very primitive level such as word-level.
  For example, treating the word "*board*" as noun or verb?

- **Syntax Level ambiguity** − A sentence can be parsed in different ways.
  For example, "*He lifted the beetle with red cap*." − Did he use cap to lift the beetle or he lifted a beetle that had red cap?

- **Referential ambiguity** − Referring to something using pronouns. For example, *Rima went to Gauri. She said, "I am tired."* − Exactly who is tired?
  One input can mean different meanings.
  Many inputs can mean the same thing.

NLP Terminology

- **Phonology** − It deals with the sounds/letters/pronunciations.

- **Morphology** – It is a study of construction of words from primitive meaningful units. .(eg. Child-children, book-books)

- **Morpheme** – It is primitive unit of meaning in a language

- **Syntax** – It refers to arranging words to make a sentence. It also involves determining the structural role of words in the sentence and in phrases.

- **Semantics** – It is concerned with the meaning of words and how to combine words into meaningful phrases and sentences.

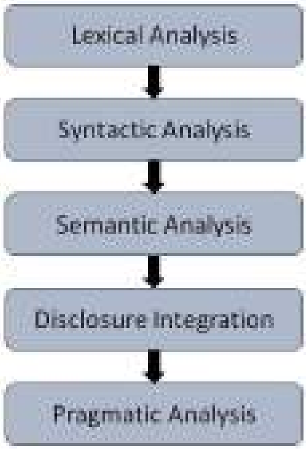
- **Pragmatics** – It deals with using and understanding sentences in different situations and how the interpretation of the sentence is affected.

- **Discourse** – It deals with how the immediately preceding sentence can affect the interpretation of the next sentence.

- **World Knowledge** – It includes the general knowledge about the world.

Steps in NLP

There are general five steps –



Lexical Analysis

Syntactic Analysis

Semantic Analysis

Disclosure Integration

Pragmatic Analysis

- **Lexical Analysis** – It involves identifying and analyzing the structure of words. Lexicon of a language means the collection of words and phrases in a language. Lexical analysis is dividing the whole chunk of txt into paragraphs, sentences, and words.
- **Syntactic Analysis (Parsing)** – It involves analysis of words in the sentence for grammar and arranging words in a manner that shows the relationship among the words. The sentence such as "The school goes to boy" is rejected by English syntactic analyzer.

- **Semantic Analysis** – It draws the exact meaning or the dictionary meaning from the text. The text is checked for meaningfulness. It is done by mapping syntactic structures and objects in the task domain. The semantic analyzer disregards sentence such as "hot ice-cream".

- **Discourse Integration** − The meaning of any sentence depends upon the meaning of the sentence just before it. In addition, it also brings about the meaning of immediately succeeding sentence.

- **Pragmatic Analysis** − During this, what was said is re-interpreted on what it actually meant. It involves deriving those aspects of language which require real world knowledge.

Implementation Aspects of Syntactic Analysis

There are a number of algorithms researchers have developed for syntactic analysis, but we consider only the following simple methods −

- Context-Free Grammar
- Top-Down Parser

Let us see them in detail –

Generative grammar

A formally specified grammar that can generate all and only the acceptable sentences of a natural language

Internal structure:

"The big dog slept"

can be bracketed

((The (big dog)) slept)

Constituent a phrase whose components form a coherent unit The internal structures are typically given labels, e.g. the big dog is a *noun phrase* (NP) and slept is a *verb phrase* (VP)

Context-Free Grammar

It is the grammar that consists of rules with a single symbol on the left-hand side of the rewrite rules.

1. a set of non-terminal symbols (e.g., S, VP);

2. a set of terminal symbols (i.e., the words);
3. a set of rules (productions), where the LHS (mother) is a single non-terminal and the RHS is a sequence of one or more non-terminal or terminal symbols (daughters);
   S -> NP VP
   V -> fish

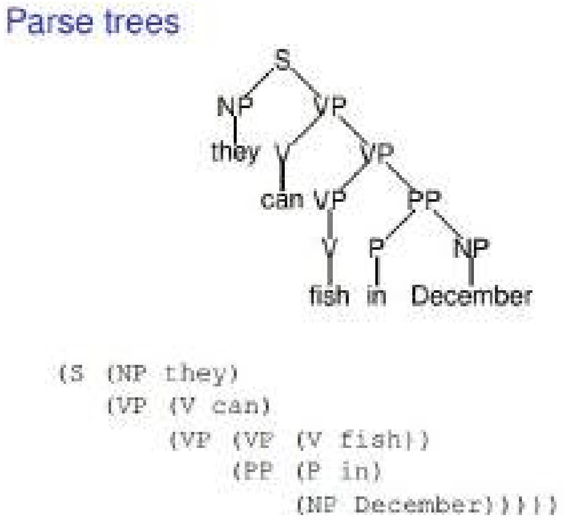
A simple CFG for a fragment of English

rules
S -> NP VP
VP -> VP PP
VP -> V
VP -> V NP
VP -> V VP
NP -> NP PP

lexicon
V -> can
V -> fish
NP -> fish
NP -> rivers

NP -> pools
NP -> December
NP -> Scotland
NP -> it

For example, *"they fish in rivers in December"*



Parse trees

```
(S (NP they)
   (VP (V can)
       (VP (VP (V fish))
           (PP (P in)
               (NP December)))))
```

Let us create grammar to parse a sentence –

"The bird pecks the grains"

Articles (DET)(Determiner) – a | an | the

Nouns – bird | birds | grain | grains

Noun Phrase (NP) – Article + Noun | Article + Adjective + Noun

= DET N | DET ADJ N

Verbs – pecks | pecking | pecked

Verb Phrase (VP) – NP V | V NP

Adjectives (ADJ) – beautiful | small | chirping

Preposition Phase(PP) – in|on|at|to|from|by|with|near|against

The parse tree breaks down the sentence into structured parts so that the computer can easily understand and process it. In order for the parsing algorithm to construct this parse tree, a set of rewrite rules, which describe what tree structures are legal, need to be constructed.

These rules say that a certain symbol may be expanded in the tree by a sequence of other symbols. According to first order logic rule, if there are two strings Noun Phrase (NP) and Verb Phrase (VP), then the string combined by NP followed by VP is a sentence. The rewrite rules for the sentence are as follows –

S → NP VP

NP → DET N | DET ADJ N

VP → V NP

Lexocon –

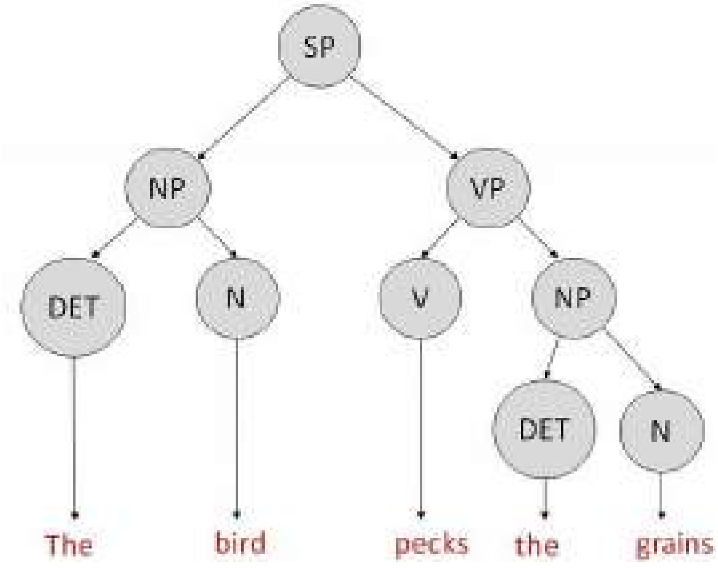
DET → a | the

ADJ → beautiful | perching

N → bird | birds | grain | grains

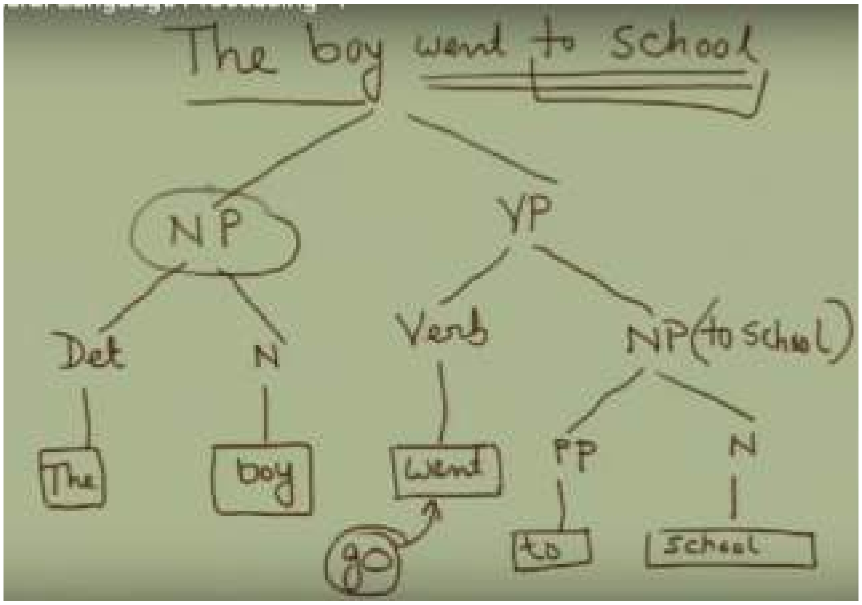
V → peck | pecks | pecking

The parse tree can be created as shown –



→ Terminal symbols

Now consider the above rewrite rules. Since V can be replaced by both, "peck" or "pecks", sentences such as "The bird peck the grains" can be wrongly permitted. i. e. the subject-verb agreement error is approved as correct.



Merit – The simplest style of grammar, therefore widely used one.

Demerits –

- For example, “The <u>dog</u> entered my room. <u>It</u> scared me”. If the question asked to the computer is “who scared you?” the computer should answer <u>dog</u> not I<u>t</u>.
- To bring out high precision, multiple sets of grammar need to be prepared. It may require a completely different sets of rules for parsing singular and plural variations, passive sentences, etc., which can lead to creation of huge set of rules that are unmanageable.

Top-Down Parser

Here, the parser starts with the S symbol and attempts to rewrite it into a sequence of *terminal symbols* that matches the classes of the words in the input sentence until it consists entirely of terminal symbols.

These are then checked with the input sentence to see if it matched. If not, the process is started over again with a different set of rules. This is repeated until a specific rule is found which describes the structure of the sentence.

Merit – It is simple to implement.

Demerits –

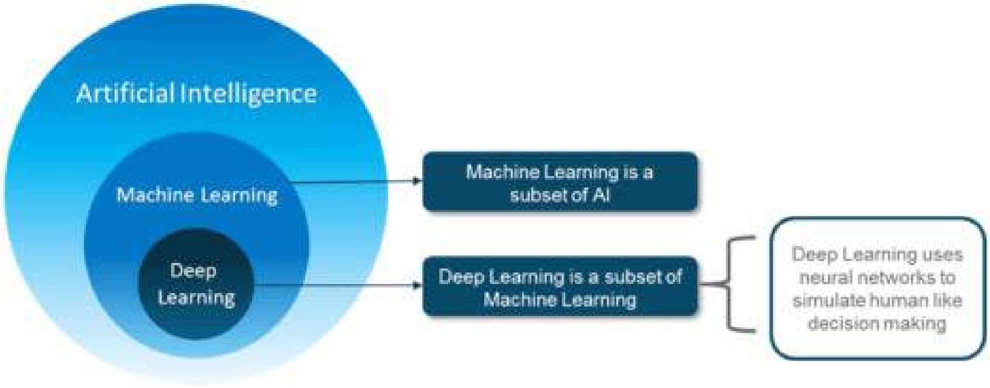
- It is inefficient, as the search process has to be repeated if an error occurs.
- Slow speed of working.

Neural Networks

Neural networks are parallel computing devices, which are basically an attempt to make a computer model of the brain. The main objective is to develop a system to perform various computational tasks faster than the traditional systems. These tasks include pattern recognition and classification, approximation, optimization, and data clustering.

Difference between AI, ML, and DL (Artificial Intelligence vs Machine Learning vs Deep Learning)

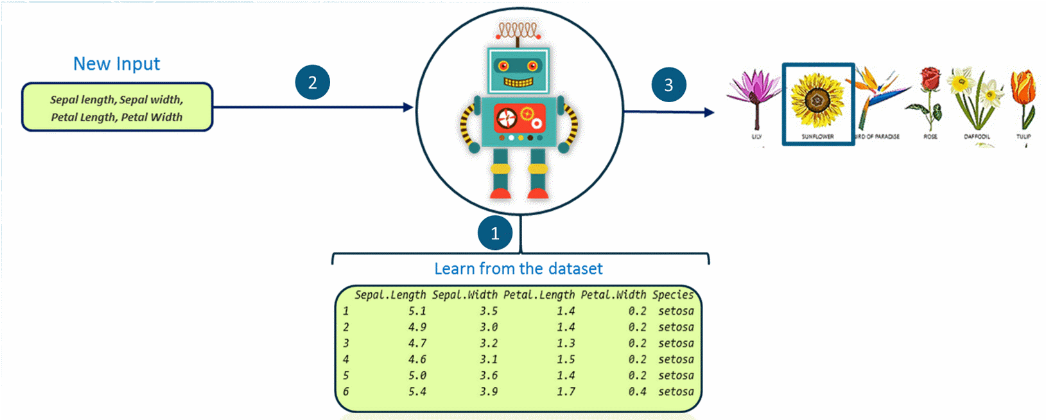
People often tend to think that Artificial Intelligence, Machine Learning, and Deep Learning are the same since they have common applications. For example, Siri is an application of AI, Machine learning and Deep learning.


Artificial Intelligence
Machine Learning
Deep Learning
Machine Learning is a subset of AI
Deep Learning is a subset of Machine Learning
Deep Learning uses neural networks to simulate human like decision making

Machine Learning

Machine Learning is a subset of Artificial Intelligence which provides computers with the ability to learn without being explicitly programmed. In machine learning, we do not have to define explicitly all the steps or conditions like any other programming application. On the contrary, the machine gets trained on a training dataset, large enough to create a model, which helps machine to take decisions based on its learning.

For example: We want to determine the species of a flower based on its petal and sepal length (leaves of a flower) using machine learning. Then, how will we do it?

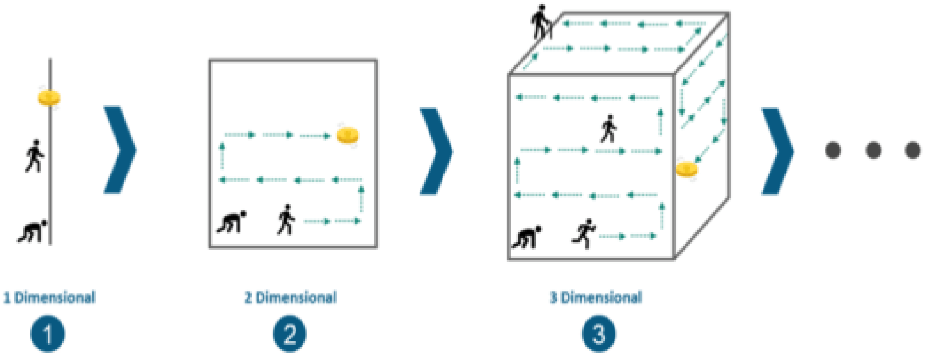


```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

1. We will feed the flower data set which contains various characteristics of different flowers along with their respective species into our machine as you can see in the above image. Using this input data set, the machine will create and train a model which can be used to classify flowers into different categories.
2. Once our model has been trained, we will pass on a set of characteristics as input to the model.
3. Finally, our model will output the species of the flower present in the new input data set. This process of training a machine to create a model and use it for decision making is called **Machine Learning.**

Limitations of Machine Learning

Machine Learning is not capable of handling high dimensional data that is where input & output is quite large. Handling and processing such type of data becomes very complex and resource exhaustive. This is termed as **Curse of Dimensionality**. To understand this in simpler terms, let's consider the following image:



1. Consider a line of 100 yards and you have dropped a coin somewhere on the line. Now, it's quite convenient for you to find the coin by simply walking on the line. This very line is a single dimensional entity.
2. Next, consider you have a square of side 100 yards each as shown in the above image and yet again, you dropped a coin somewhere in between. Now, it is quite evident that you are going to take more time to find the coin within that square as compared to the previous scenario. This square is a 2 dimensional entity.

3. Let's take it a step ahead by considering a cube of side 100 yards each and you have dropped a coin somewhere in between. Now, it is even more difficult to find the coin this time. This cube is a 3 dimensional entity.

Hence, you can observe the complexity is increasing as the dimensions are increasing. And in real-life, the high dimensional data that we were talking about has thousands of dimensions that make it very complex to handle and process. The high dimensional data can easily be found in use-cases like Image processing, NLP, Image Translation etc.

Machine learning was not capable of solving these use-cases and hence, Deep learning came to the rescue. Deep learning is capable of handling the high dimensional data and is also efficient in focusing on the right features on its own. This process is called feature extraction.

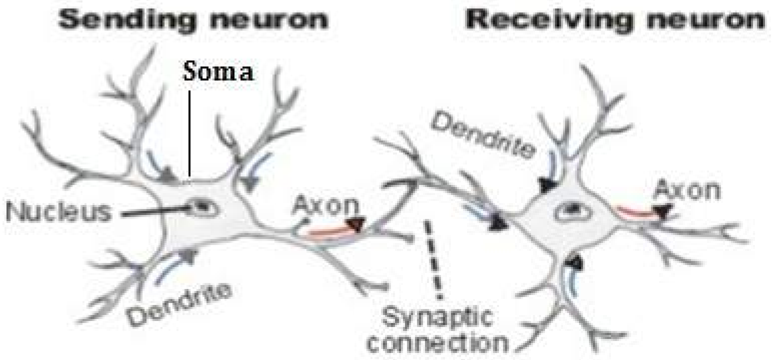
How Deep Learning Works?

In an attempt to re-engineer a human brain, Deep Learning studies the basic unit of a brain called a brain cell or a neuron. Inspired from a neuron an **artificial neuron** or a **perceptron** was developed. Now, let us understand the functionality of biological neurons and how we mimic this functionality in the **perceptron or an artificial neuron**.

Biological Neuron

A nerve cell (neuron) is a special biological cell that processes information. According to estimation, there are huge numbers of neurons, approximately 10^{11} with numerous interconnections, approximately 10^{15}.

Schematic Diagram



Working of a Biological Neuron

As shown in the above diagram, a typical neuron consists of the following four parts with the help of which we can explain its working –

- **Dendrites –** They are tree-like branches, responsible for receiving the information from other neurons it is connected to. In other sense, we can say that they are like the ears of neuron.
- **Soma –** It is the cell body of the neuron and is responsible for processing of information, they have received from dendrites.
- **Axon –** It is just like a cable through which neurons send the information as output, connected to dendrites of other neurons via synapses.
- **Synapses –** It is the connection between the axon and other neuron dendrites. It transfers the information between neurons (electrical-chemical-electrical).

The brain is principally composed of about 10 billion neurons, each connected to about 10,000 other neurons. Each of the neuronal cell bodies (soma), and the lines are the input and output channels (dendrites and axons) which connect them.

Each neuron receives electrochemical inputs from other neurons at the dendrites. If the sum of these electrical inputs is sufficiently powerful to activate the neuron, it transmits an electrochemical signal along the axon, and passes this signal to the other neurons whose dendrites are attached at any of the axon terminals. These attached neurons may then fire.

It is important to note that a neuron fires only if the total signal received at the cell body exceeds a certain level. The neuron either fires or it doesn't, there aren't different grades of firing.

So, our entire brain is composed of these interconnected electro-chemical transmitting neurons. From a very large number of extremely simple processing units (each performing a weighted sum of its inputs, and then firing a binary signal if the total input exceeds a certain level) the brain manages to perform extremely complex tasks.

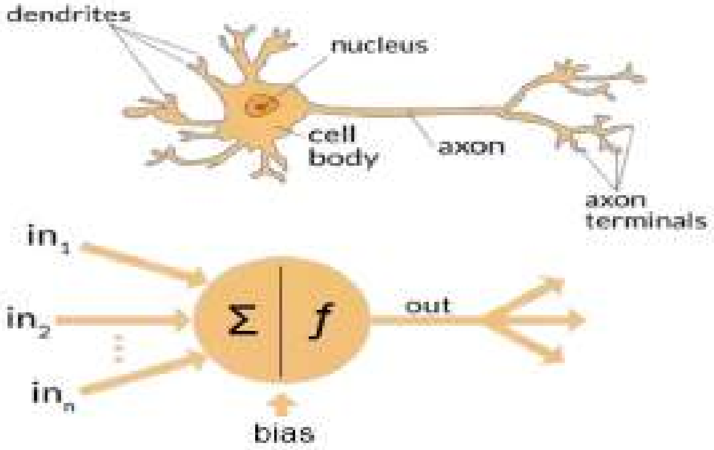
Artificial Neural Network (ANN)

Artificial Neural Network (ANN) is an efficient computing system whose central theme is borrowed from the analogy of biological neural networks. ANNs are also named as “artificial neural systems,” or “parallel distributed processing systems,” or “connectionist systems.” ANN acquires a large collection of units that are interconnected in some pattern to allow communication between the units. These units, also referred to as nodes or neurons, are simple processors which operate in parallel.

Basic Structure of ANNs

The idea of ANNs is based on the belief that working of human brain by making the right connections can be imitated using silicon and wires as living neurons and dendrites.

The human brain is composed of 86 billion nerve cells called neurons. They are connected to other thousand cells by Axons. Stimuli from external environment or inputs from sensory organs are accepted by dendrites. These inputs create electric impulses, which quickly travel through the neural network. A neuron can then send the message to other neuron to handle the issue or does not send it forward.

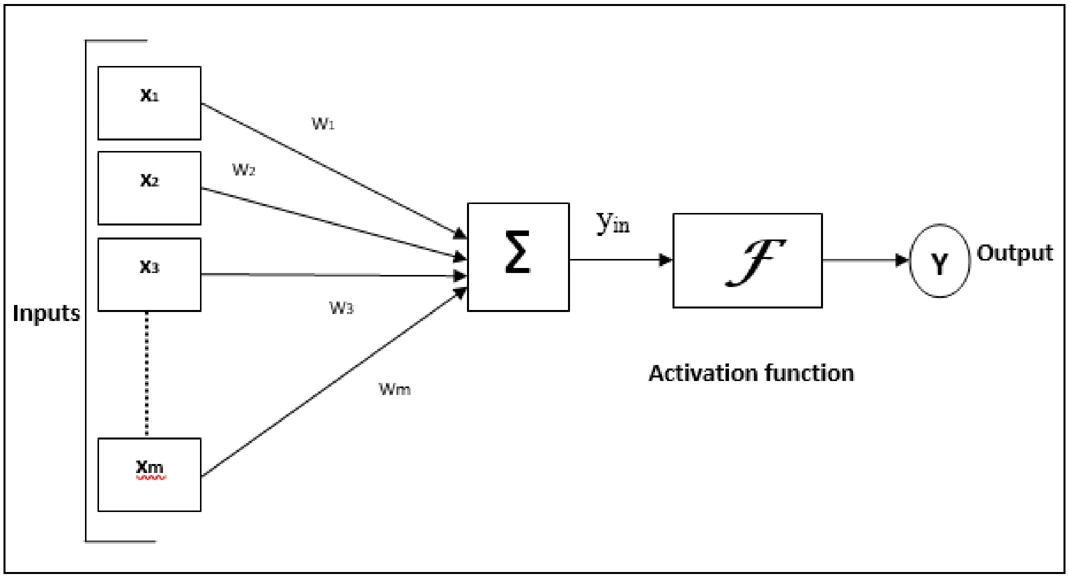


ANNs are composed of multiple **nodes**, which imitate biological **neurons** of human brain. Every neuron is connected with other neuron through a connection ***link*** and they interact with each other. Each connection link is associated with a ***weight*** that has information about the input signal. This is the most useful information for neurons to solve a particular problem because the weight usually excites (stimulates) or inhibits (prevents) the signal that is being communicated. Each neuron has an internal state, which is called an ***activation signal***. The nodes can take input data and perform simple operations on the data. Output signals called ***activation or node value***, which are produced after combining the input signals and activation rule, may be sent to other neurons.

The following illustration shows a simple ANN –

Model of Artificial Neural Network

The following diagram represents the general model of ANN followed by its processing.



The artificial neuron given in this figure has N input, denoted as X1, X2, ...Xm. Each line connecting these inputs to the neuron is assigned a weight, which are denoted as W1, W2, .., Wm respectively. Weights in the artificial model correspond to the synaptic connections in biological neurons.

The inputs (x) received from the input layer are multiplied with their assigned weights w. The multiplied values are then added to form the Weighted Sum. The weighted sum of the inputs and their respective weights are then applied to a relevant Activation Function. The activation function maps the input to the respective output.

For the above general model of artificial neural network, the net input can be calculated as follows –

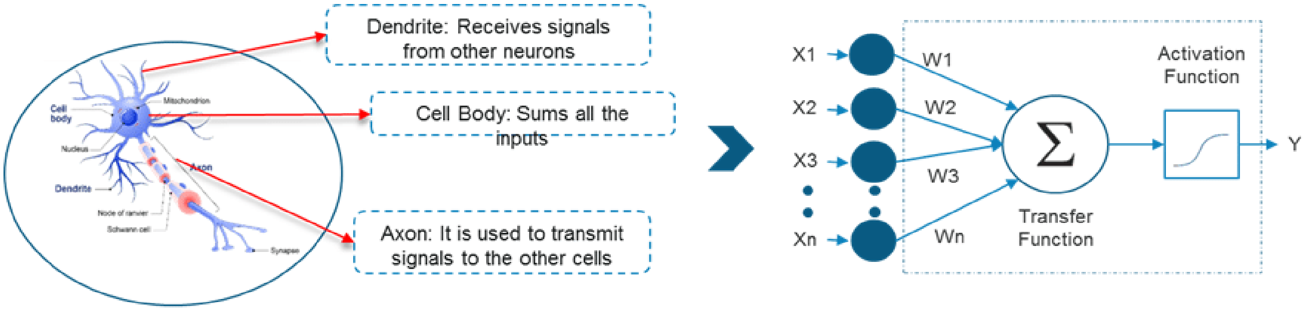
$$y_{in} = x_1.w_1 + x_2.w_2 + x_3.w_3 \dots x_m.w_m$$

$$\text{i.e., Net input } y_{in} = \sum_i^m x_i.w_i$$

The output can be calculated by applying the activation function over the net input.

$$Y = F(y_{in})$$

Output = function (net input calculated)


Dendrite: Receives signals from other neurons
Cell Body: Sums all the inputs
Axon: It is used to transmit signals to the other cells
Cell body
Mitochondrion
Nucleus
Axon
Dendrite
Node of ranvier
Schwann cell
Synapse
X1
X2
X3
Xn
W1
W2
W3
Wn
Σ
Transfer Function
Activation Function
Y

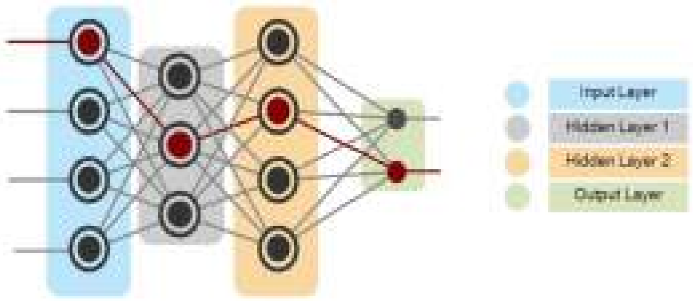
Neuron

Schematic for a neuron in a neural net

- If we focus on the structure of a biological neuron, it has dendrite which is used to receive inputs. These inputs are summed in the cell body and using the Axon it is passed on to the next biological neuron as shown in the above image.
- Similarly, a perceptron receives multiple inputs, applies various transformations and functions and provides an output.
- As we know that our brain consists of multiple connected neurons called neural network, we can also have a network of artificial neurons called perceptrons to form a Deep neural network.

So, let's move ahead in this Deep Learning Tutorial to understand how a Deep neural network looks like.

What is Deep Learning?



Any Deep neural network will consist of three types of layers:

- The Input Layer
- The Hidden Layer
- The Output Layer

- **Input Nodes** – The Input nodes provide information from the outside world to the network and are together referred to as the "Input Layer". No computation is performed in any of the Input nodes – they just pass on the information to the hidden nodes.
- **Hidden Nodes** – The Hidden nodes have no direct connection with the outside world (hence the name "hidden"). They perform computations and transfer information from the input nodes to the output nodes. A collection of hidden nodes forms a "Hidden Layer". While a network will only have a single input layer and a single output layer, it can have zero or multiple Hidden Layers. A Multi-Layer Perceptron has one or more hidden layers.
- **Output Nodes** – The Output nodes are collectively referred to as the "Output Layer" and are responsible for computations and transferring information from the network to the outside world.

We want to perform Image recognition using Deep Networks:

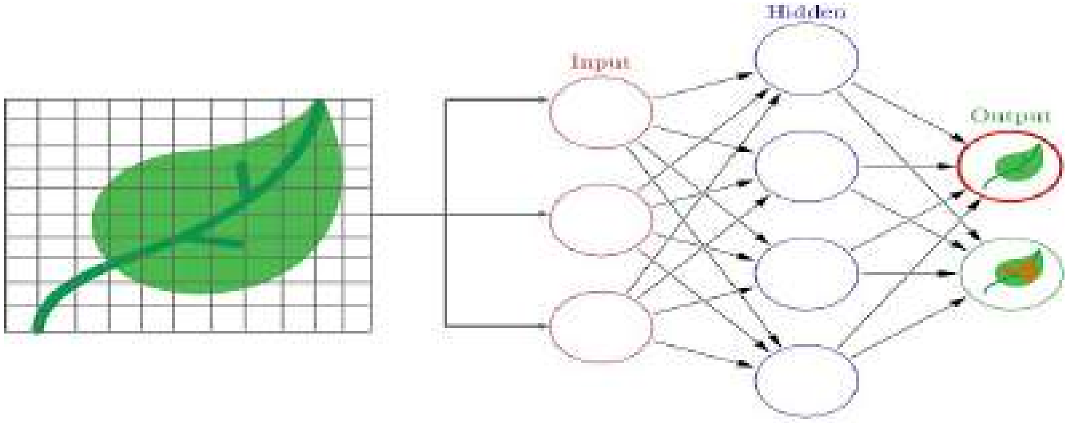
Example: Consider a scenario where you are to build an Artificial Neural Network (ANN) that classifies images into two classes:

- Class A: Containing images of non-diseased leaves
- Class B: Containing images of diseased leaves

So how do you create a Neural network that classifies the leaves into diseased and non-diseased crops?

The process always begins with processing and transforming the input in such a way that it can be easily processed. In our case, each leaf image will be broken down into pixels depending on the dimension of the image.

For example, if the image is composed of 30 by 30 pixels, then the total number of pixels will be 900. These pixels are represented as matrices, which are then fed into the input layer of the Neural Network.



Just like how our brains have neurons that help in building and connecting thoughts, an ANN has perceptrons that accept inputs and process them by passing them on from the input layer to the hidden and finally the output layer.

As the input is passed from the input layer to the hidden layer, an initial random weight is assigned to each input. The inputs are then multiplied with their corresponding weights and their sum is sent as input to the next hidden layer.

Here, a numerical value called bias is assigned to each perceptron, which is associated with the weightage of each input. Further, each perceptron is passed through activation or a transformation function that determines whether a particular perceptron gets activated or not.

An activated perceptron is used to transmit data to the next layer. In this manner, the data is propagated (Forward propagation) through the neural network until the perceptrons reach the output layer.

At the output layer, a probability is derived which decides whether the data belongs to class A or class B.

ANN versus BNN

Before taking a look at the differences between Artificial Neural Network (ANN) and Biological Neural Network (BNN), let us take a look at the similarities based on the terminology between these two.

Biological Neural Network (BNN)	**Artificial Neural Network (ANN)**

Soma	Node
Dendrites	Input
Synapse	Weights or Interconnections
Axon	Output

Processing of ANN depends upon the following three building blocks –

- Network Topology
- Adjustments of Weights or Learning

➢ Network Topology

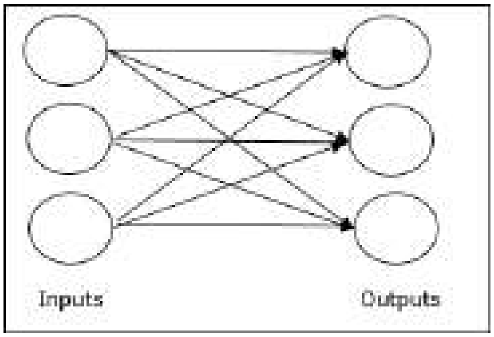
A network topology is the arrangement of a network along with its nodes and connecting lines.

According to the topology, ANN can be classified as the following kinds –

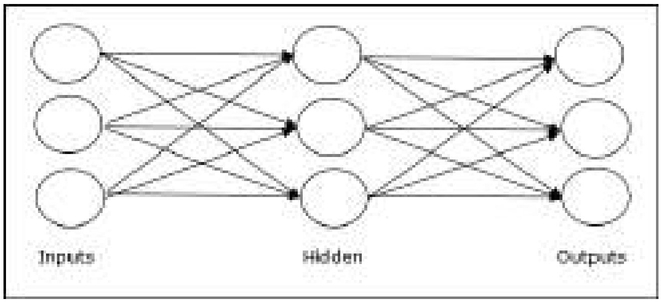
Feedforward Network

It is a non-recurrent network having processing units/nodes in layers and all the nodes in a layer are connected with the nodes of the previous layers. The connection has different weights upon them. There is no feedback loop means the signal can only flow in one direction, from input to output. They are used in pattern generation/recognition/classification. They have fixed inputs and outputs. It may be divided into the following two types –

- **Single layer feedforward network** – The concept is of feedforward ANN having only one weighted layer. In other words, we can say the input layer is fully connected to the output layer.



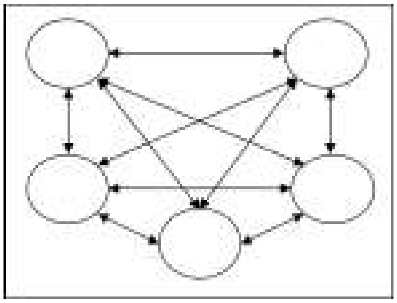
- **Multilayer feedforward network** – The concept is of feedforward ANN having more than one weighted layer. As this network has one or more layers between the input and the output layer, it is called hidden layers.



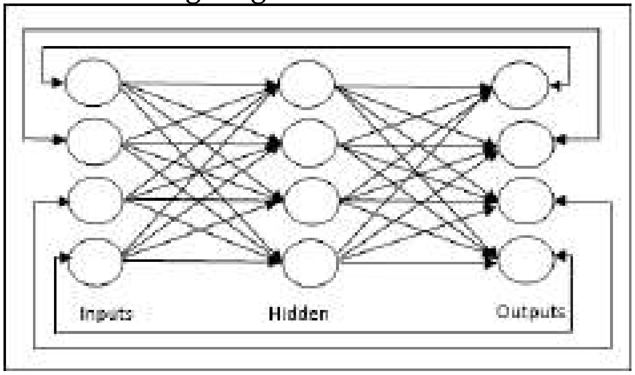
Feedback Network

As the name suggests, a feedback network has feedback paths, which means the signal can flow in both directions using loops. This makes it a non-linear dynamic system, which changes continuously until it reaches a state of equilibrium. It may be divided into the following types −

- **Recurrent networks** − They are feedback networks with closed loops. Following are the two types of recurrent networks.
- **Fully recurrent network** − It is the simplest neural network architecture because all nodes are connected to all other nodes and each node works as both input and output.



- **Jordan network** − It is a closed loop network in which the output will go to the input again as feedback as shown in the following diagram.



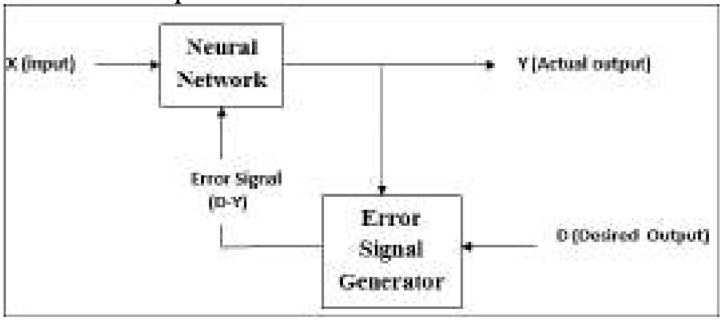
➢ Adjustments of Weights or Learning Techniques

Learning, in artificial neural network, is the method of modifying the weights of connections between the neurons of a specified network. Learning in ANN can be classified into three categories namely supervised learning, unsupervised learning, and reinforcement learning.

- **Supervised Learning**

As the name suggests, this type of learning is done under the supervision of a teacher. This learning process is dependent.

During the training of ANN under supervised learning, the input vector is presented to the network, which will give an output vector. This output vector is compared with the desired output vector. An error signal is generated, if there is a difference between the actual output and the desired output vector. On the basis of this error signal, the weights are adjusted until the actual output is matched with the desired output.

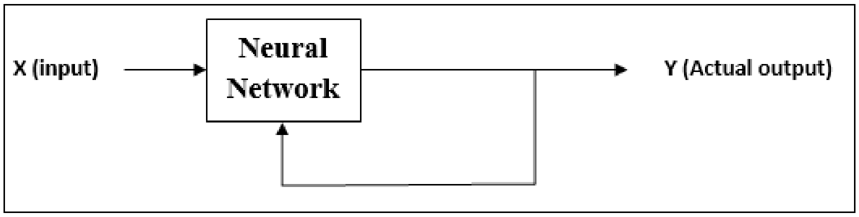


- **Unsupervised Learning**

As the name suggests, this type of learning is done without the supervision of a teacher. This learning process is independent.

During the training of ANN under unsupervised learning, the input vectors of similar type are combined to form clusters. When a new input pattern is applied, then the neural network gives an output response indicating the class to which the input pattern belongs.

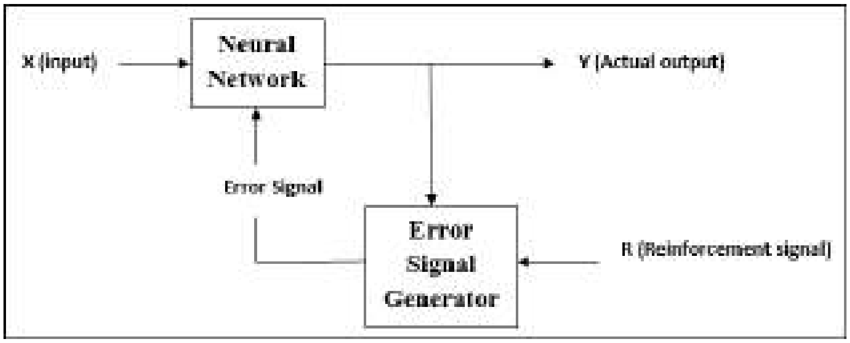
There is no feedback from the environment as to what should be the desired output and if it is correct or incorrect. Hence, in this type of learning, the network itself must discover the patterns and features from the input data, and the relation for the input data over the output.



- **Reinforcement Learning**

As the name suggests, this type of learning is used to reinforce or strengthen the network over some critic information. This learning process is similar to supervised learning; however we might have very less information.

During the training of network under reinforcement learning, the network receives some feedback from the environment. This makes it somewhat similar to supervised learning. However, the feedback obtained here is evaluative not instructive, which means there is no teacher as in supervised learning. After receiving the feedback, the network performs adjustments of the weights to get better critic information in future.



Naïve Bayes Classifier

What is a classifier?

A classifier is a machine learning model that is used to discriminate different objects based on certain features.

Principle of Naive Bayes Classifier:

A **Naive Bayes** classifier is a probabilistic machine learning model that's used for classification task. This classifier assigns class labels to the samples from the available set of labels. This method assumes each feature's value as independent and will not consider any correlation or relationship between the features.

Bayes Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Using Bayes theorem, we can find the probability of **A** happening, given that **B** has occurred. Here, **B** is the evidence and **A** is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called naive.

A and **B** are Boolean variables that represent the occurrence of an event.

$$0 \leq P(A) \leq 1$$

If an event is certain to occur then the probability is 1. i.e, **P(True)=1.**

If an event is certain to not occur then the probability is 0. i.e, **P(False)=0.**

If the probability of the event is uncertain then the probability is **between 0 and 1.**

For example

Suppose you are one of the 1/10 people that have a headache (H). i.e, **P (H) =1/10.**

Suppose 1/40 of people have the flu (F) i.e**, P (F) =1/40**

Given the fact that you have a headache what are the chances that you have the flu? **P (F/H) =?**

$$P(F|H) = \frac{(P(H|F)P(F))}{(P(H))} = ?$$

P(F|H) = Posterior probability of Flu given Headache

P(H|F) =conditional probability of x given Flu (often called likelihood of headache given Flu.

P(F) = prior probability of hypothesis Flu

P(H) = prior probability that example is observed.

$$P(F|H) = \frac{(0.5*0.025)}{0.1}$$
$$P(F|H) = 0.125$$

Conditional probability and the chain rule.

Probability that **A** and **B** occur.

$$P(A|B) = \frac{(P(A \wedge B))}{(P(B))}$$
$$P(A \wedge B) = P(A|B)P(B)$$

What is the probability that a person has a **head ache** and the **Flu**?

$$P(H \wedge F) = P(H|F)P(F)$$
$$P(H \wedge F) = 0.5*0.025$$
$$P(H \wedge F) = 0.0125$$

A problem with Naïve Bayes Classification

- The assumption that all class attributes are independent results in a loss of accuracy.
- Recall the example about headache and flu shown before. Clearly there is a dependency between attributes which a naïve classifier would not be able to model.

The solution for this problem is Bayesian Belief Networks

Bayesian networks

A Bayesian network is a data structure used to represent knowledge in an uncertain domain (i.e) to represent the dependence between variables and to give a whole specification of the joint probability distribution.

A Bayesian network is a probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph (DAG).

A belief network is a graph in which the following holds.

I. A set of random variables makes up the nodes of the network.

II. A set of directed links or arrows connects pairs of nodes x→y,x has a direct influence on y.

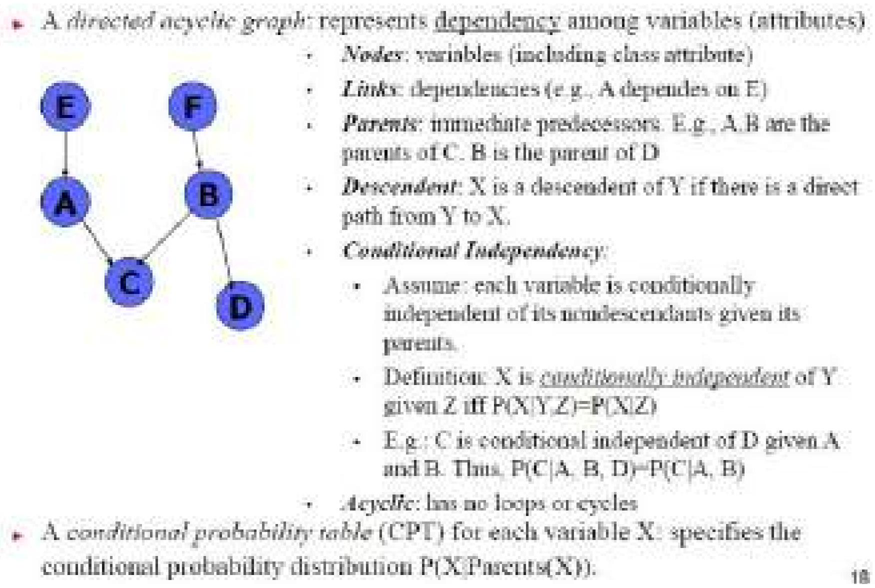
III. Each node has a conditional probability tale that quantifies the effects that the parents have on the node. The parents of a node are all nodes that have arrows pointing to it.

IV. Graph has no directed cycles(DAG)

The other names of Belief network are Bayesian network ,probabilistic network, casual network and knowledge map.

Bayesian networks are ideal for taking an event that occurred and predicting the likelihood that any one of several possible known causes was the contributing factor.

For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Disease and symptoms are connected using a network diagram. All symptoms connected to a disease are used to calculate the probability of the existence of the disease.



- A *directed acyclic graph*: represents dependency among variables (attributes)
  - *Nodes*: variables (including class attribute)
  - *Links*: dependencies (e.g., A dependes on E)
  - *Parents*: immediate predecessors. E.g., A,B are the parents of C. B is the parent of D
  - *Descendent*: X is a descendent of Y if there is a direct path from Y to X.
  - *Conditional Independency*:
    - Assume: each variable is conditionally independent of its nondescendants given its parents.
    - Definition: X is *conditionally independent* of Y given Z iff P(X|Y,Z)=P(X|Z)
    - E.g.: C is conditional independent of D given A and B. Thus, P(C|A, B, D)=P(C|A, B)
  - *Acyclic*: has no loops or cycles
- A *conditional probability table* (CPT) for each variable X: specifies the conditional probability distribution P(X|Parents(X)).

18

Example 1:

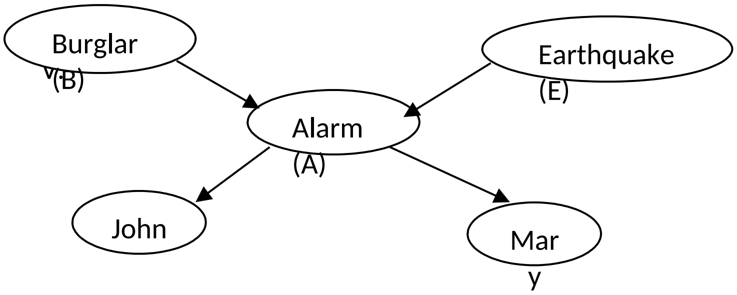
A new burglar alarm has been installed at home.

- It is fairly reliable at detecting a burglary but also responds on occasion to minor earthquakes.
- You also have two neighbors, John and Mary, who have promised to call you at work when they hear the alarm.
- John always calls when he hears the alarm but sometimes confuses the telephone ringing with the alarm and calls then too.
- Mary on the other hand likes rather loud music and sometimes misses the alarm together.
- Given the evidence of who has or has not called estimate the probability of a burglary

Uncertainty:

I. Mary currently listening to loud music
II. John confuses telephone ring with alarm → laziness and ignorance in the operation
III. Alarm may fail off → power failure, dead battery, cut wires etc

DAG (Directed Acyclic Graph) for the example



Conditional Probability Tables

B	P(Burglar)
T	0.001
F	0.999

E	P(Earthquake)
T	0.002
F	0.998

Burglary	Earthquake	P(Alarm/Burglary, Earthquake)	
		True	False
T	T	0.95	0.05
T	F	0.94	0.06
F	T	0.29	0.71
F	F	0.001	0.999

A	P(John=T)	P(John=F)
T	0.90	0.10
F	0.05	0.95

A	P(Mary=T)	P(Mary=F)
T	0.70	0.30
F	0.01	0.99

Let's infer the probability that the burglar is not in the house given that John and Mary heard the alarm

P (John, Mary, A, ~B, ~E)
= P(P1|A). P(P2|A).P(A|~B~E).P(~B).P(~E)
= .90 X 0.70 X 0.001 X 0.999 X 0.998
= 0.00062

Building a Bayesian Network

A knowledge engineer can build a Bayesian network. There are a number of steps the knowledge engineer needs to take while building it.

Example 2:

Lung cancer. A patient has been suffering from breathlessness. He visits the doctor, suspecting he has lung cancer. The doctor knows that barring lung cancer, there are various other possible diseases the patient might have such as tuberculosis and bronchitis.

Step 1: Gather Relevant Information of Problem

- Is the patient a smoker? If yes, then high chances of cancer and bronchitis.
- Is the patient exposed to air pollution? If yes, what sort of air pollution?
- Take an X-Ray positive X-ray would indicate either TB or lung cancer.

Step 2: Identify Interesting Variables

The knowledge engineer tries to answer the questions –

- Which nodes to represent?
- What values can they take? In which state can they be?

For now let us consider nodes, with only discrete values. The variable must take on exactly one of these values at a time.

Common types of discrete nodes are –

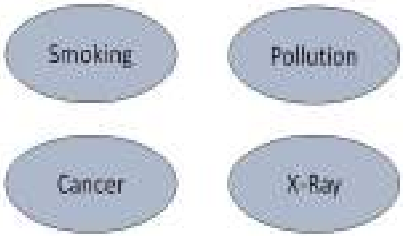
- **Boolean nodes** – they represent propositions, taking binary values TRUE (T) and FALSE (F).
- **Ordered values** – A node *Pollution* might represent and take values from {low, medium, high} describing degree of a patient's exposure to pollution.
- **Integral values** – A node called *Age* might represent patient's age with possible values from 1 to 120. Even at this early stage, modeling choices are being made.

Step 3: Create nodes

Possible nodes and values for the lung cancer example –

Node Name	Type	Value
Pollution	Binary	{LOW, HIGH, MEDIUM}
Smoker	Boolean	{TRUE, FASLE}
Lung-Cancer	Boolean	{TRUE, FASLE}
X-Ray	Binary	{Positive, Negative}

Nodes Creation

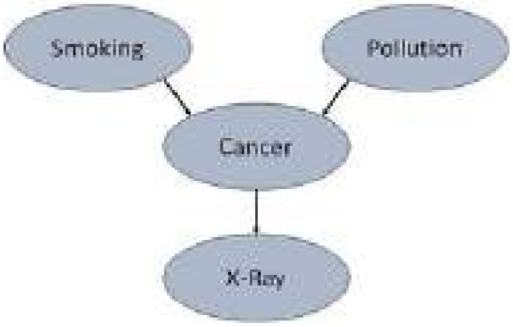


Step 4: Create Arcs between Nodes

Topology of the network should capture qualitative relationships between variables.

For example, what causes a patient to have lung cancer? - Pollution and smoking. Then add arcs from node *Pollution* and node *Smoker* to node *Lung-Cancer.*

Similarly if patient has lung cancer, then X-ray result will be positive. Then add arcs from node *Lung-Cancer* to node *X-Ray.*



Step 5: Specify Topoıogy

Conventionally, BNs are laid out so that the arcs point from top to bottom. The set of parent nodes of a node X is given by Parents(X).

The *Lung-Cancer* node has two parents (reasons or causes): *Pollution* and *Smoker*, while node *Smoker* is an **ancestor** of node *X-Ray*. Similarly, *X-Ray* is a child (consequence or effects) of node *Lung-Cancer* and **successor** of nodes *Smoker* and *Pollution.*

Step 6: Conditional Probabilities

Now quantify the relationships between connected nodes: this is done by specifying a conditional probability distribution for each node. As only discrete variables are considered here, this takes the form of a **Conditional Probability Table (CPT).**

First, for each node we need to look at all the possible combinations of values of those parent nodes. Each such combination is called an **instantiation** of the parent set. For each distinct instantiation of parent node values, we need to specify the probability that the child will take.

For example, the *Lung-Cancer* node's parents are *Pollution* and *Smoking.* They take the possible values = { (H,T), (H,F), (L,T), (L,F)}. The CPT specifies the probability of cancer for each of these cases as <0.05, 0.02, 0.03, 0.001> respectively.

Applications of Neural Networks

1. Social Media

- **Facebook**

  As soon as you upload any photo to Facebook, the service automatically highlights faces and prompts friends to tag.

- **Instagram**

  uses deep learning by making use of a connection of recurrent neural networks to identify the contextual meaning of an emoji – which has been steadily replacing slangs (for instance, a laughing emoji could replace "rofl").
- **Pinterest**

  Pinterest uses computer vision – another application of neural networks, where we teach computers to "see" like a human, in order to automatically identify objects in images (or "pins", as they call it) and then recommend visually similar pins. Other applications of neural networks at Pinterest include spam prevention, search and discovery, ad performance and monetization, and email marketing.

2. Online Shopping

- **Search**

  Your Amazon searches ("earphones", "pizza stone", "laptop charger", etc) return a list of the most relevant products related to your search, without wasting much time. In a description of its product search technology, Amazon states that its algorithms learn automatically to combine multiple relevant features. It uses past patterns and adapts to what is important for the customer in question.
- **Recommendations**

  Amazon shows you recommendations using its "customers who viewed this item also viewed", "customers who bought this item also bought", and also via curated recommendations on your homepage, on the bottom of the item pages, and through emails. Amazon makes use of Artificial Neural Networks to train its algorithms to learn the pattern and behaviour of its users. This, in turn, helps Amazon provide even better and customized recommendations.

3. Banking/Personal Finance

- **Cheque Deposits through Mobile**

  Most large banks are eliminating the need for customers to physically deliver a cheque to the bank by offering the ability to deposit cheques through a smartphone application. The technologies that power these applications use Neural Networks to decipher and convert handwriting on checks into text. Essentially, Neural Networks find themselves at the core of any application that requires handwriting/speech/image recognition.
- **Fraud Prevention**

  Artificial Intelligence is used to create systems that learn through training what types of transactions are fraudulent (speak learning, speak Neural Networks!).

4. Image Processing and Character recognition

Character recognition like handwriting has lot of applications in fraud detection (e.g. bank fraud) and even national security assessments.

Image recognition is an ever-growing field with widespread applications from facial recognition in social media, cancer detection in medicine, satellite imagery processing for agricultural and defense usage.

5. Forecasting

Forecasting is required extensively in everyday business decisions (e.g. sales, financial allocation between products, capacity utilization), in economic and monetary policy, in finance and stock market.

Advantages of Neural Networks

- A neural network can perform tasks that a linear program can not.
- When an element of the neural network fails, it can continue without any problem by their parallel nature.
- A neural network learns and does not need to be reprogrammed.
- It can be implemented in any application.
- It can be performed without any problem.

Limitations of Neural Networks

- The neural network needs the training to operate.
- The architecture of a neural network is different from the architecture of microprocessors, therefore, needs to be emulated.
- Requires high processing time for large neural networks.