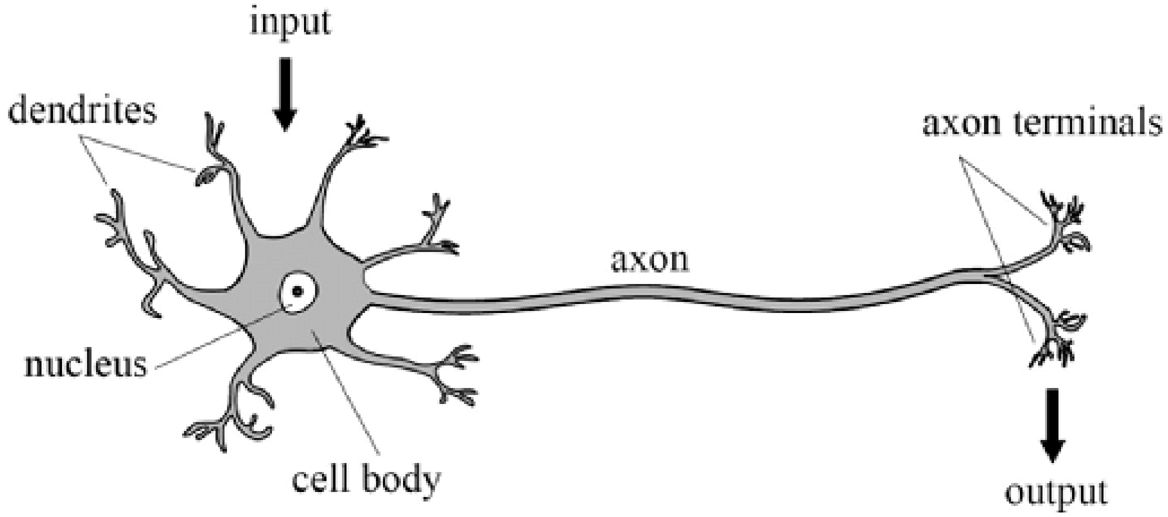
In This Chapter

- *Artificial Neural Networks (ANN)*
  - *Neural Networks (NN) and ANN*
  - *Activation functions: unit (unary and binary), ramp, piecewise linear, & sigmoid*
  - *Training and testing: Basic concept*
  - *Mc-Colloch-Pits neuron model*
  - *Realization of AND, OR, NOT, and XOR gates*
  - *Neural network architectures*
  - *Single layer feed-forward architecture: ADALINE, Perceptron NN*
  - *Applications of ANN*
- *Natural Language Processing (NLP)*
- *Fundamentals of language processing*

Biological Neuron

- 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.
- **Synapses:** It is the connection between the axon and other neuron dendrites.



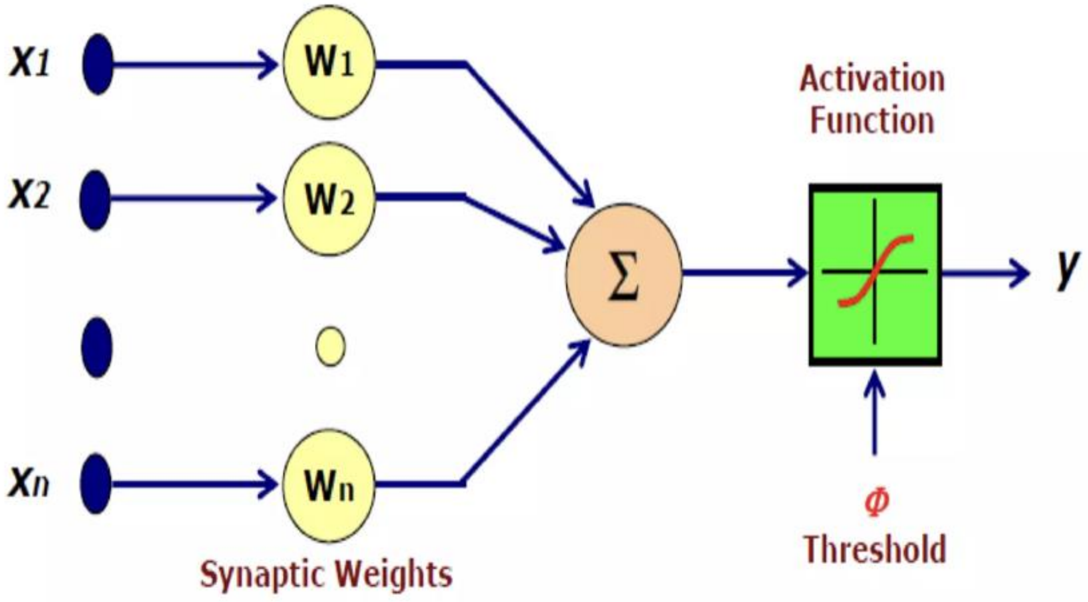
Contd..

- **Neuron** is a cell in brain whose principle function is the collection, Processing, and dissemination of signals.
- Networks of such neurons is called neural network .
- Neural network is capable of information processing.
- It is also known as:
  - Connectionism
  - Parallel and distributed processing and
  - neural computing

What is Artificial Neural network?

- An Artificial Neural Network (ANN) is an information processing paradigm(model) that is inspired by the way biological nervous systems process information.
- The key elements of this paradigm are:
- **Nodes(units):** Nodes represent a cell of neural network.
- **Links:** Links are directed arrows that show propagation of information from one node to another node.
- **Activation:** Activations are inputs to or outputs from a unit.
- **Weight:** Each link has weight associated with it which determines strength of the connection.
- **Activation function:** A function which is used to derive output activation from the input activations to a given node is called activation function

Contd..



X1
X2
Xn
W1
W2
Wn
Σ
Activation Function
y
Φ
Threshold
Synaptic Weights

Why use neural networks?

- Ability to derive meaning from complicated or imprecise data.
- Other reasons for using the ANNs are as follows:
  - **Adaptive learning:** An ability to learn how to do tasks based on the data given for training or initial experience.
  - **Self-Organization:** An ANN can create its own organization or representation of the information it receives during learning time.
  - **Real Time Operation:** ANN computations may be carried out in parallel

Neural networks versus conventional computers

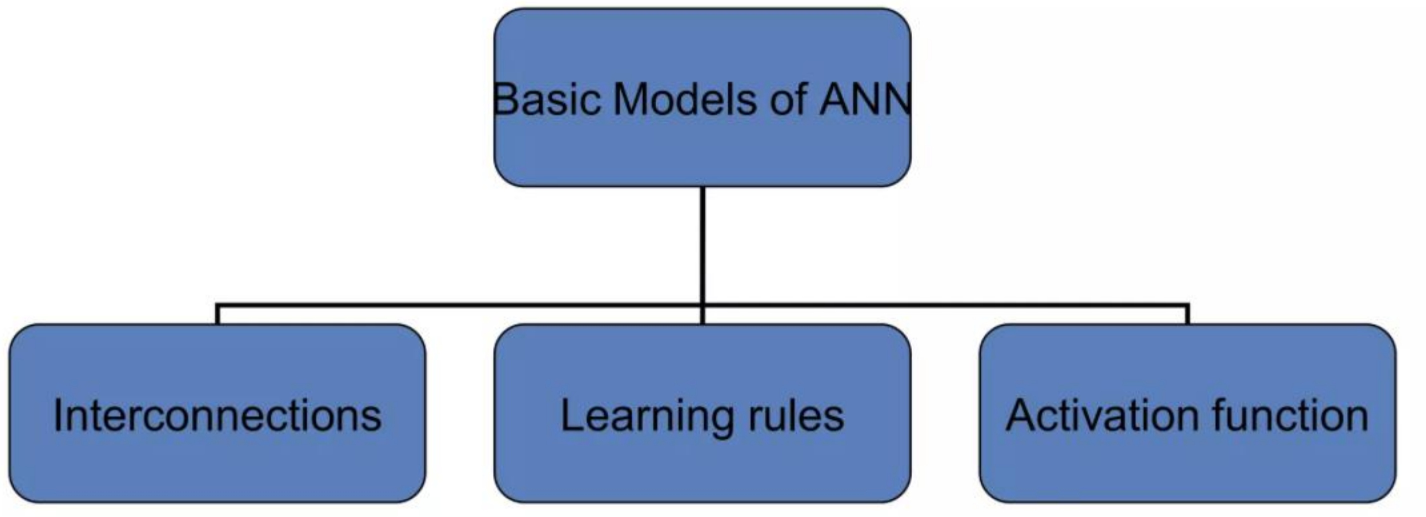
- Parallelism is NNs characteristics, because the computations of its components are largely independent of each other. Neural networks and conventional algorithmic computers are not in competition but complement each other.
- Different approach used to problem solving is the main difference between neural networks and conventional computers.
- Conventional computers use an algorithmic approach, for instance, the conventional computer follows a set of instructions in order to solve a problem.
- Thus, the specific steps must be well-defined so that the computer can follow to solve the problem.
- This rule restricts the problem-solving ability of conventional computers. That means the computer must be given exactly how to solve a problem.

Contd..

- As for neural networks, information is processed in much a similar way the human brain does.
- A NN consists of a large amounts of highly interconnected processing elements--neurons which works in parallel to solve a specific problem.
- Neural networks can learn by examples, but they cannot be programmed to perform a specific task like conventional computers.
- Moreover, the examples must be selected carefully otherwise it might be functioning incorrectly or waste time.
- This problem arises because of no method found currently to testify if the system is faulty or not

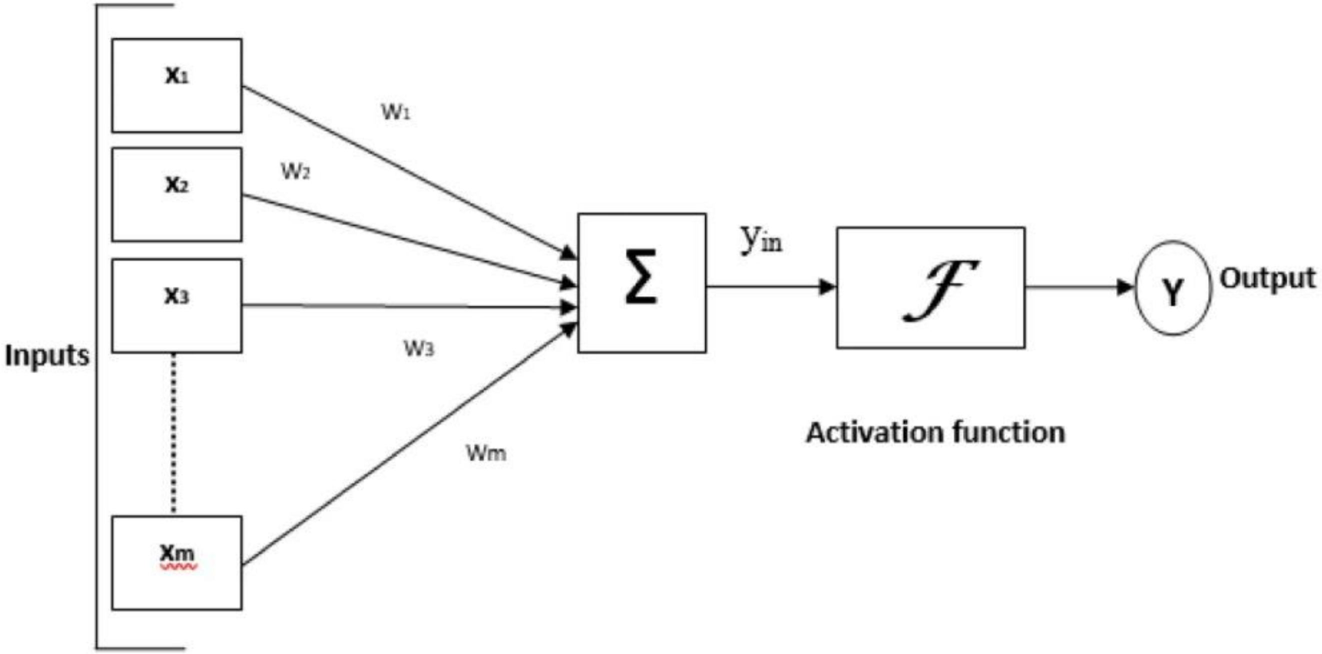
Basic model of ANN

- Processing of ANN depends upon the following three building blocks:
  - Network Architecture(interconnections)
  - Adjustments of Weights or Learning rules
  - Activation Functions



First artificial neurons: The McCulloch-Pitts model

- The McCulloch-Pitts model is an extremely simple artificial neuron as shown in figure below:



Contd..

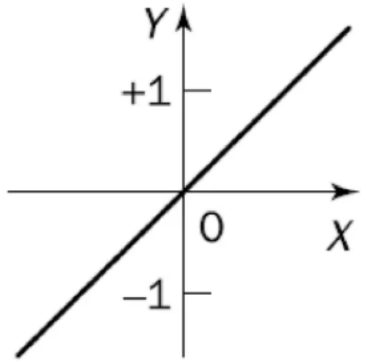
- In the previous figure, the variables W,, W2, W3.... Wm are called "weights". Xy, X2, X3,..Xm represent the inputs. Net input can be calculated as a weighted sum of its inputs (actual input): Net Input = x, W, + X2W2+ X3W3+..+XmWm
- Then check if (Net input < Threshold) or not (i.e., apply the activation function to the Net Input to derive the output).
- If it is, then the output is made zero.
- Otherwise, it is made a one.
- Note: The inputs can be either a zero or a one. And the output is a zero or a one

Activation functions

- It is also known as the transfer function.
- Activation function typically falls into one of three main categories:
- Linear activation function
  - Piecewise, and
  - Ramp
- Threshold activation function
  - Step, and
  - sign
- Sigmoid activation function
  - Binary, and
  - Bipolar

Linear Activation

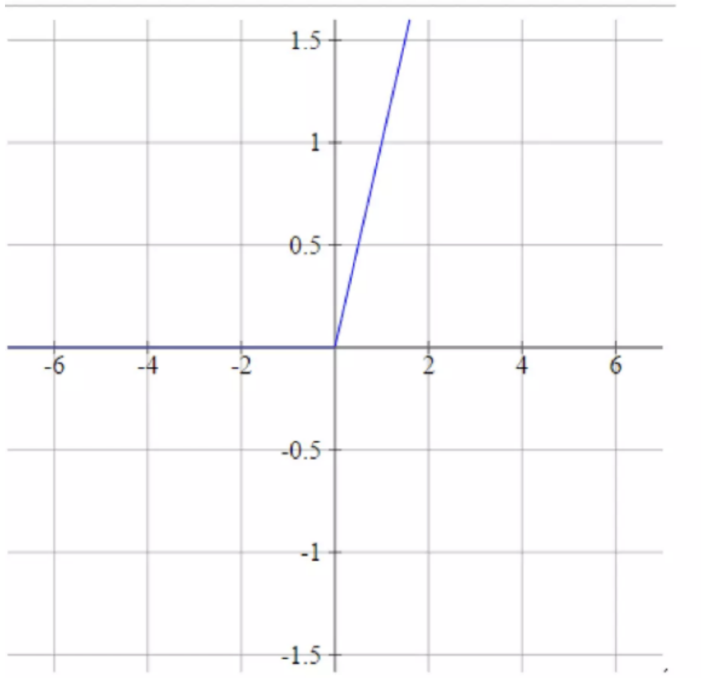
- Also known as identity activation function. For linear activation functions, the output activity is proportional to the total weighted sum(Net Input).
- Output (Y)= mx + c, where m and c are constant



- Neurons with the linear function are often used for linear approximation

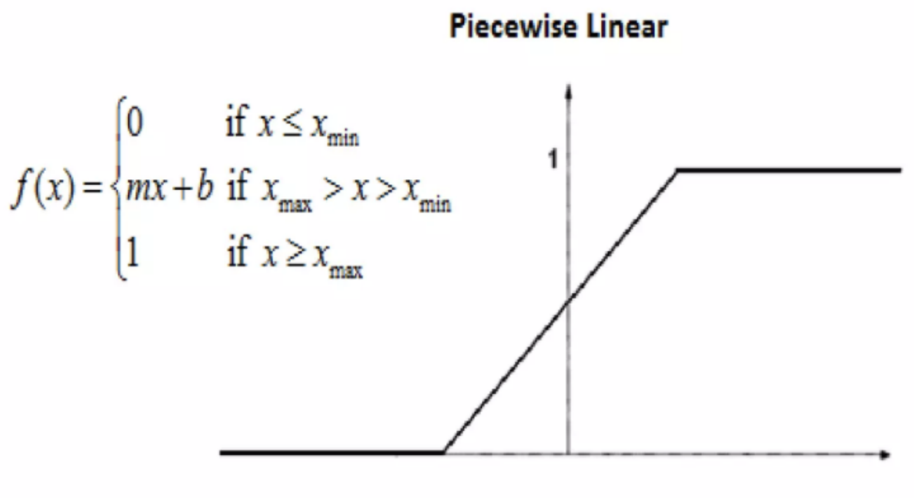
Contd..

- Linear activation functions are of two types:
  - Piecewise, And
  - Ramp
- Ramp Linear activation function:
  - Also known as Rectifier, or Max.
  - Output(Y) = Max(0, Net input)



Contd..

- **Piecewise Linear Activation**
- Choose some Xmin and *max which is our "range".
- Everything less than this range will be 0, and everything greater than this range will be 1.
- Anything else is linearly-interpolated between.
- Here x is net input, f(x) is output

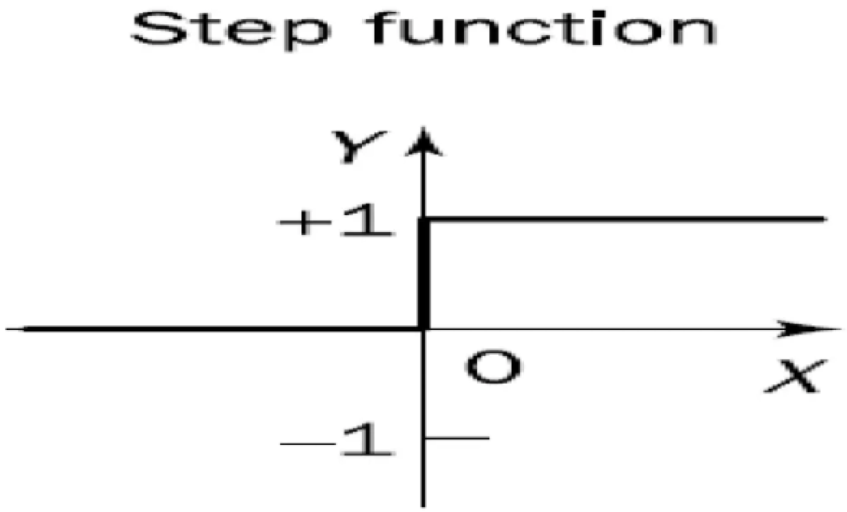


Threshold(Unit) activation Function

- Also known as hard limit functions. For threshold activation functions, the output are set at one of two levels, depending on whether the Net _input is greater than or less than some threshold value.
  Threshold activation functions are of two types:
  - Step activation function ,and
  - sign activation function.
- hard limit functions, are often used in decision-making neurons for classification and pattern recognition tasks

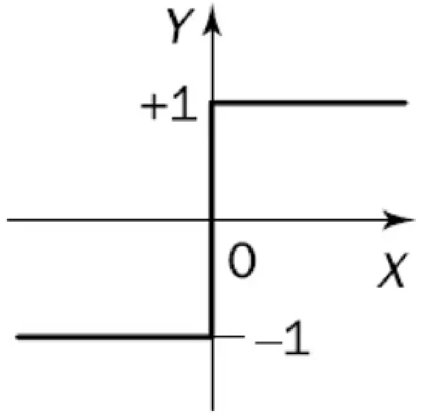
Step(Binary) activation

$$\text{output}(y) = \begin{cases} 1, \text{ if } X \geq 0 \\ 0, \text{ if } X < 0 \end{cases}$$



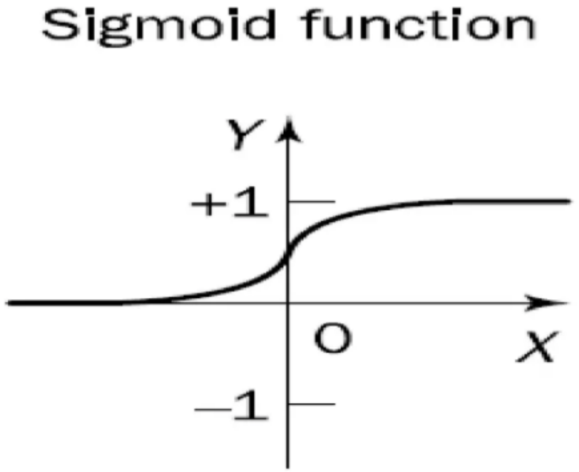
Sign(Bipolar) activation

- Output(Y) $= \begin{cases} +1, \text{ if } X \geq 0 \\ -1, \text{ if } X < 0 \end{cases}$



Sigmoid Activation Function

- For sigmoid activation functions, the output varies continuously but not linearly as the input changes.



Contd..

- It is of two types as follows:

Binary Sigmoid Function

This activation function perform input editing between 0 and 1. It is positive in nature. It is always bounded, which means its output cannot be less than 0 and more than 1. It is also strictly increasing in nature, which means more the input higher would be the output. It can be defined as:

$$\text{output (Y)} = \frac{1}{1+\exp(-x)}$$

Contd..

- **Bipolar sigmoid function:**
  - This activation function performs input editing between - 1 and 1. It can be positive or negative in nature. It is always bounded, which means its output cannot be less than -1 and more than 1. It is also strictly increasing in nature like sigmoid function. It can be defined a

$$\text{Output(Y)} = \frac{2}{1+\exp(-x)} - 1 = \frac{1-\exp(x)}{1+\exp(x)}$$

Logic Gates With MP Neurons

- We can use McCulloch-Pitts neurons to implement the basic logic gates.
- All we need to do is find the appropriate connection weights and neuron thresholds to produce the right outputs for each set of inputs

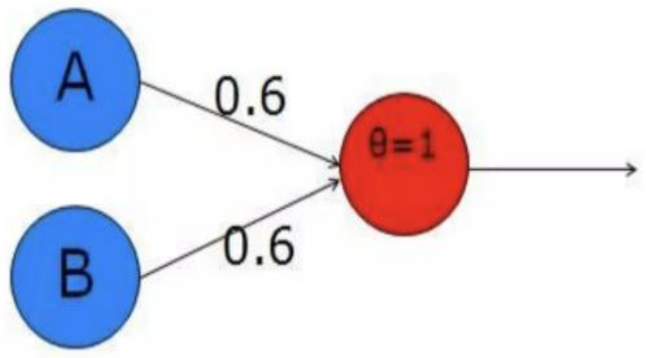
Realization of AND gate

- Looking at the logic table for the A^B we can see that we only want the neuron to output a 1 when both inputs are activated.

A	B	A^B
0	0	0
0	1	0
1	0	0
1	1	1

Contd..

- To do this, we want the sum of both inputs to be greater than the threshold, but each input alone must be lower than the threshold.
- Let's use a threshold of 1 (simple and convenient!). So, now we need to choose the weights according to the constraints :
- E.g., 0.6 and 0.6
- With these weights, individual activation of either input A or B will not exceed the threshold, while the sum of the two will be 1.2, which exceeds the threshold and causes the neuron to fire. Here is a diagram



Contd..

- Here, letter theta denote the threshold.
- A and B are the two inputs. They will each be set to 1 or 0 depending upon the truth of their proposition.
- The red neuron is our decision neuron. If the sum of the weighted inputs is greater than the threshold, this will output a 1, otherwise it will output 0 .
- So, to test it by hand, we can try setting A and B to the different values in the truth table and seeing if the decision neuron's output matches the A^B column

Contd..

- If A=0 & B=0 -> 0*0.6 + 0*0.6 = 0. This is not greater than the threshold of 1, so the output = 0.
- If A=0 & B=1 -> 0*0.6 + 1*0.6 = 0.6. This is not greater than the threshold, so the output = 0.
- If A=1 & B=0 - exactly the same as above.
- If A=1 & B=1 -> 1*0.6 + 1*0.6 = 1.2. This exceeds the threshold, so the output = 1.

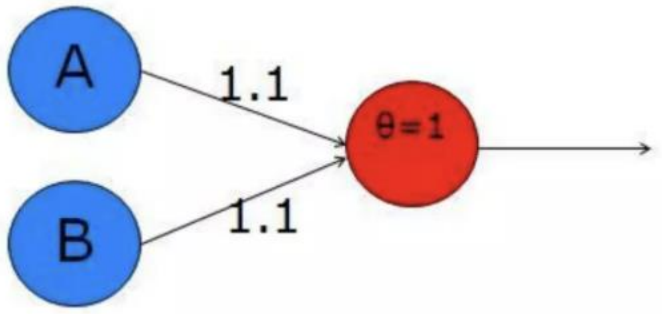
Realization of OR Gate

- Logic Table for the OR GATE is as shown in below:
- In this case, we want the output to be 1 when either or both of the inputs, A and B are active, but 0 when both of the inputs are 0

A	B	AvB
0	0	0
0	1	1
1	0	1
1	1	1

Contd..

- This is simple enough. If we make each synapse greater than the threshold, then it'll fire whenever there is any activity in either or both of A and B. This is shown in figure below. Synaptic values of 1.1 are sufficient to surpass the threshold of 1 whenever their respective input is active.



Realization of NOT Gate

- Logic table for the NOT gate is as shown in table below:
- The NOT operator simply negates the input, When A is true (Value 1), ~A is false(value 0) and vice-versa.

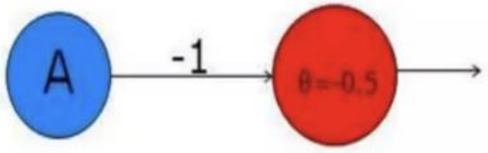
A	¬ A
0	1
1	0

Contd..

- This is a little tricky. We have to change a 1 to a 0 - this is easy, just make sure that the input doesn't exceed the threshold.
- However, we also have to change a 0 to a 1 - how can we do this?
- The answer is to think of our decision neuron as a tonic neuron - one whose natural state is active.
- To make this, all we do is set the threshold lower than 0, so even when it receives no input, it still exceeds the threshold.
- In fact, we have to set the synapse to a negative number (here use 1) and the threshold to some number between that and 0 (use - 0.5)

Contd..

- The decision neuron shown below, with threshold -0.5 and synapse weight - 1, will reverse the input:
- A =1 --> output = 0
- A =0 --> output = 1



Realization of XOR Gate

- XOR (exclusive OR) operator actually put a spanner in the works of neural network research for a long time because it is not possible to create an XOR gate with a single neuron, or even a single layer of neurons - we need to have two layers.
- The truth table for the XOR gate is as shown below:

A	B	AxorB
0	0	0
0	1	1
1	0	1
1	1	0

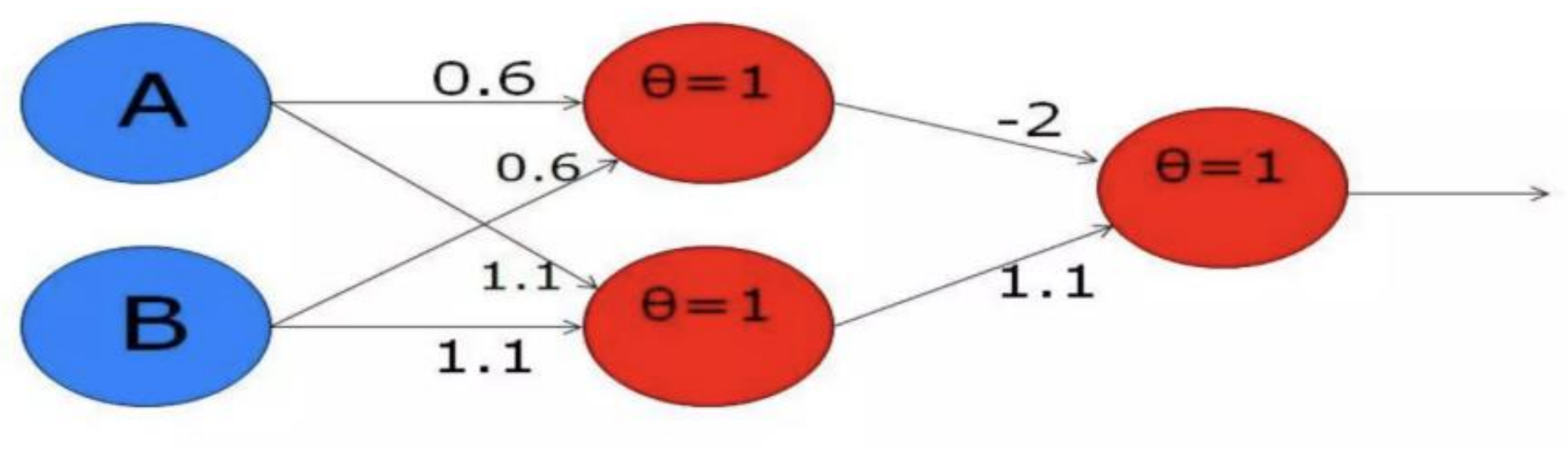
Contd..

- Exclusive OR means that we want a truth value of 1 either when A is 1, or when B is 1(i.e. A and B have different truth value), but not when both A and B are 1 or 0(i.e., both have same truth value).
- The truth table on the above shows this.
- To use a real world example, in the question "would you like tea or coffee?" , the "or" is actually and exclusive or, because the person is offering you one or the other, but not both. Contrast this with "would you like milk or sugar?". In this case you can have milk, or sugar, or both

Contd..

- The only way to solve this problem is to have a bunch of neurons working together. start by breaking down the XOR operation into a number of simpler logical functions:
- A xor B = (AvB) ^ (A^B)
- This line of logic contains three important operations: an OR operator in the brackets on the left, an AND operator in the brackets on the right, and another AND operator in the middle.
- We can create a neuron for each of these operations, and stick them together like this

Contd..



Contd..

- The upper of the two red neurons in the first layer has two inputs with synaptic weights of 0.6 each and a threshold of 1, exactly the same as the AND gate we made earlier. This is the AND function in the brackets on the right of the formula wrote earlier.
- Notice that it is connected to the output neuron with a negative synaptic weight (- 2). This accounts for the NOT operator that precedes the brackets on the right hand side.
- The lower of the two red neurons in the first layer has two synaptic weights of 1.1 and a threshold of 1, just like the OR gate we made earlier. This neuron is doing the job of the OR operator in the brackets on the left of the formula.
- The output neuron is performing another AND operation - the one in the middle of the formula. Practically, this output neuron is active whenever one of the inputs, A or B is on, but it is overpowered by the inhibition of the upper neuron in cases when both A and B are on

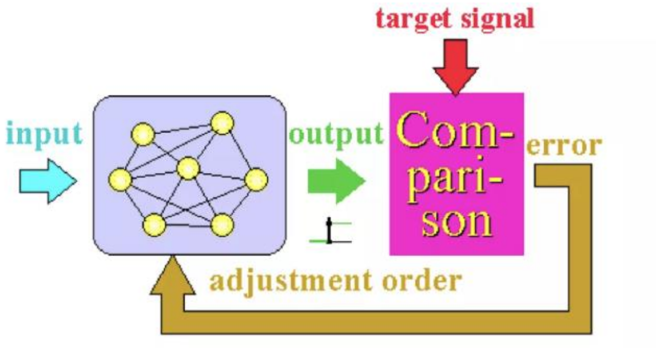
Learning in Neural Networks

- **Learning:**
- Learning in neural networks is carried out by adjusting the connection weights among neurons.
- There is no algorithm that determines how the weights should be assigned in order to solve specific problems. Hence, the weights are determined by a learning process
- **Learning may be classified into two categories:**
  - Supervised Learning
  - Unsupervised Learning

Contd..

Supervised Learning:

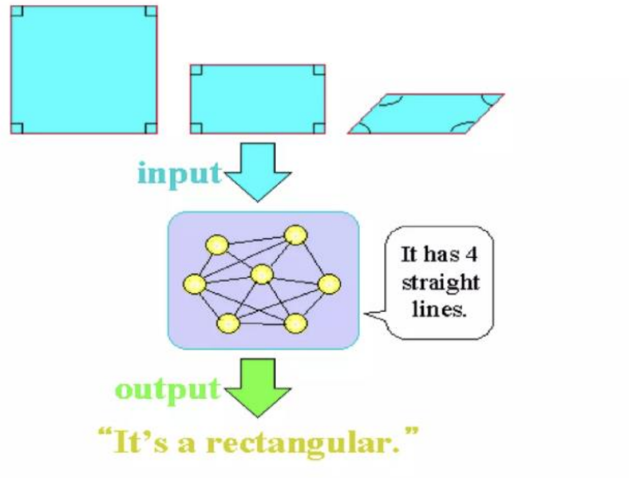
- In supervised learning, the network is presented with inputs together with the target (teacher signal) outputs.
- Then, the neural network tries to produce an output as close as possible to the target output by adjusting the values of internal weights.
- The most common supervised learning method is the "error correction method".
  - Neural networks are trained with this method in order to reduce the error (difference between the network's output and the desired output) to zero.



Contd..

Unsupervised Learning:

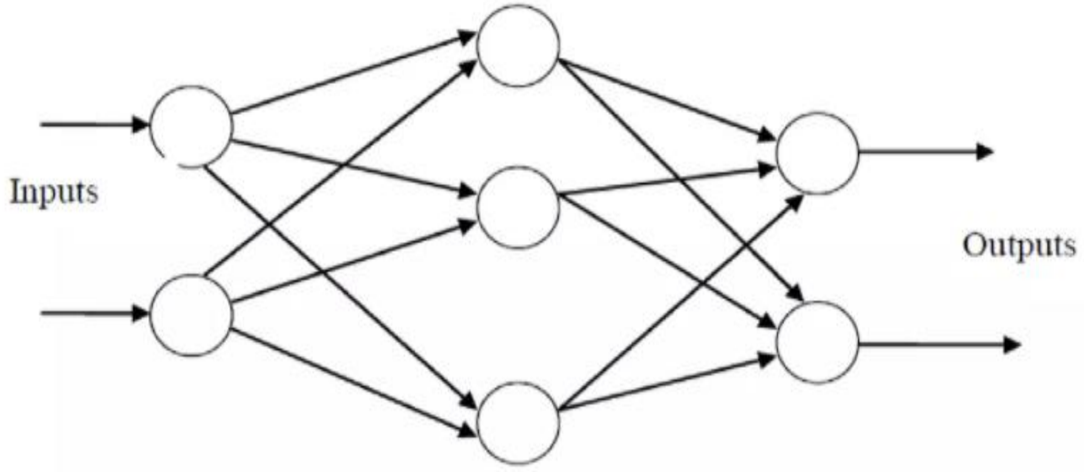
- In unsupervised learning, there is no teacher (target signal/output) from outside and the network adjusts its weights in response to only the input patterns
- A typical example of unsupervised learning is Hebbian learning



Network Architecture

Feed-forward networks:

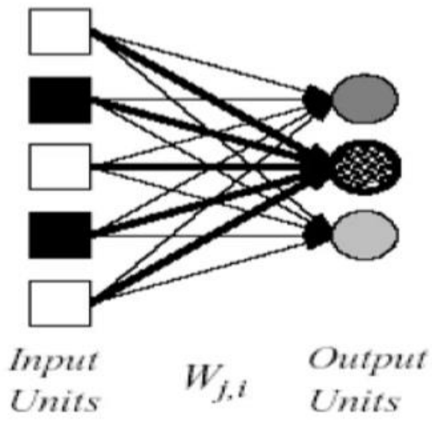
- Feed-forward ANNs allow signals to travel one way only; from input to output.
- There is no feedback (loops) i.e. the output of any layer does not affect that same layer.
- Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs.



Types of Feed Forward Neural Network:

a. **Single Layer Neural Networks**

A neural network in which all the input connected directly to the outputs is called a single layer neural network.



Contd..

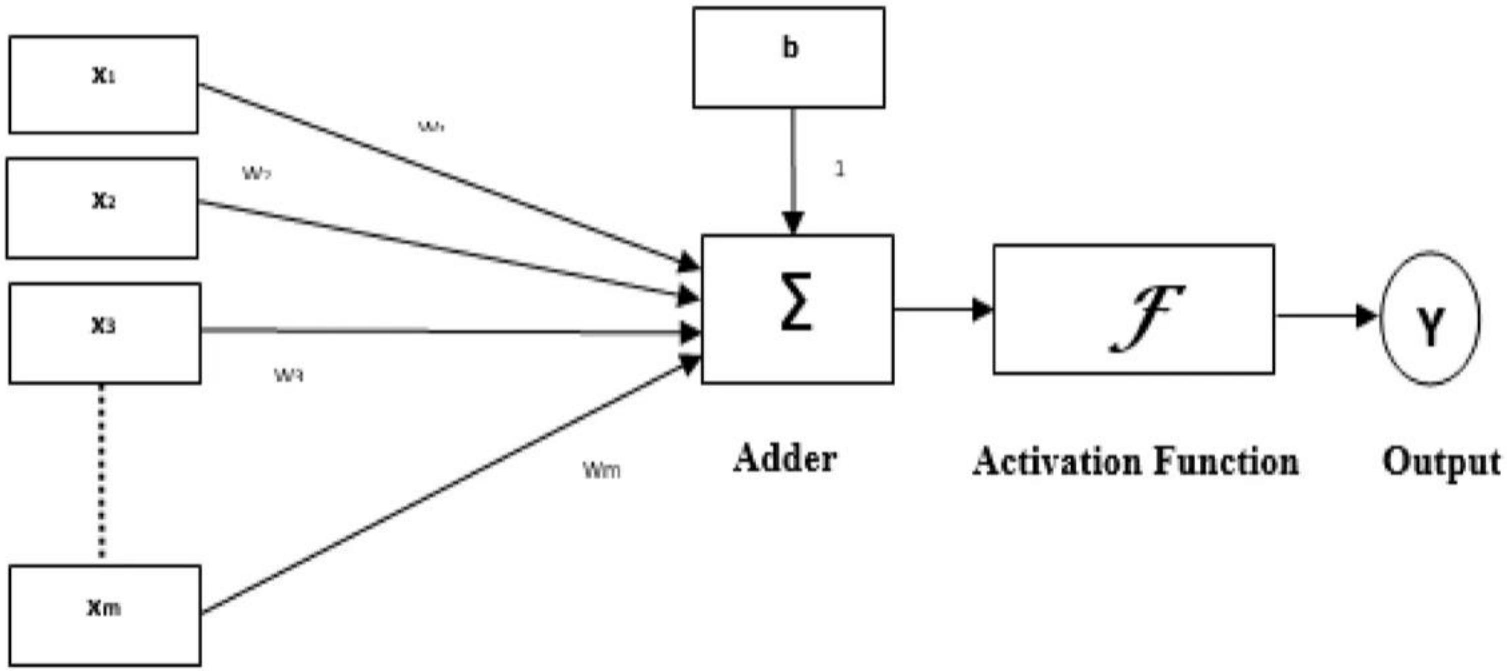
- A. Single-layer feed forward Neural Networks
- Two types
  - Perception
  - ADLINE

Perceptron

- Developed by Frank Rosenblatt by using McCulloch and Pitts model, perceptron is the basic operational unit of artificial neural networks.
- It employs supervised learning rule and is able to classify the data into two classes.
- Operational characteristics of the perceptron:
  - It consists of a single neuron with an arbitrary number of inputs along with adjustable weights, but the output of the neuron is 1 or -1 depending upon the input. It also consists of a bias whose weight is always 1.
- Following figure gives a schematic representation of the perceptron

Contd..

- Following figure gives a schematic representation of the perception:



Contd..

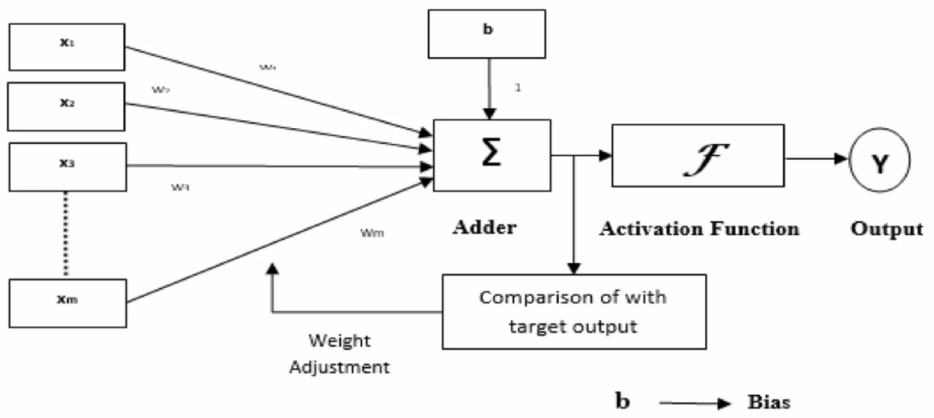
- Perceptron thus has the following three basic elements:
- **Links:** It would have a set of connection links, which carries a weight including a bias always having weight 1.
- **Adder:** It adds the input after they are multiplied with their respective weights. It also add up the bias.
- **Activation function:** It limits the output of neuron. The most basic activation function is a sign function that has two possible outputs. This function returns 1, if the input is positive, and -1 for any negative input

Adaptive Linear Neuron (ADALINE)

- ADALINE which stands for Adaptive Linear Neuron, is a network having a single linear unit.
- It was developed by Widrow and Hoff in 1960. Some important points about ADALINE are as follows:
  - It uses bipolar activation function.
  - It uses delta rule for training to minimize the Mean-Squared Error (MSE) between the actual output and the desired/target output.
  - The weights and the bias are adjustable

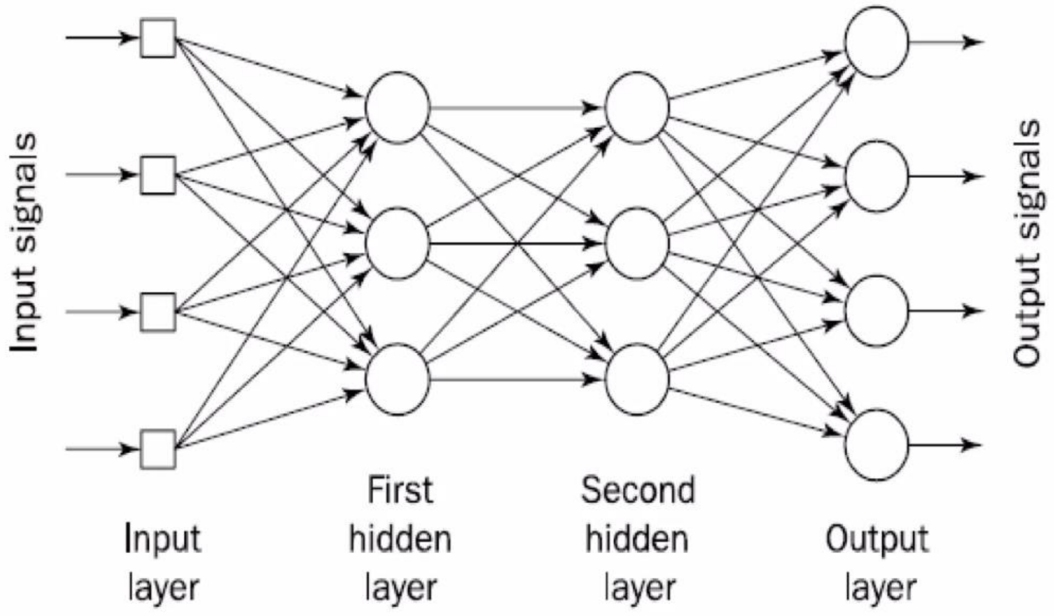
Contd..

- Architecture :
  - The basic structure of ADALINE is similar to perceptron having an extra feedback loop with the help of which the calculated output is compared with the desired/target output.
  - After comparison on the basis of training algorithm, the weights and bias will be updated



Types of Feed Forward Neural Network

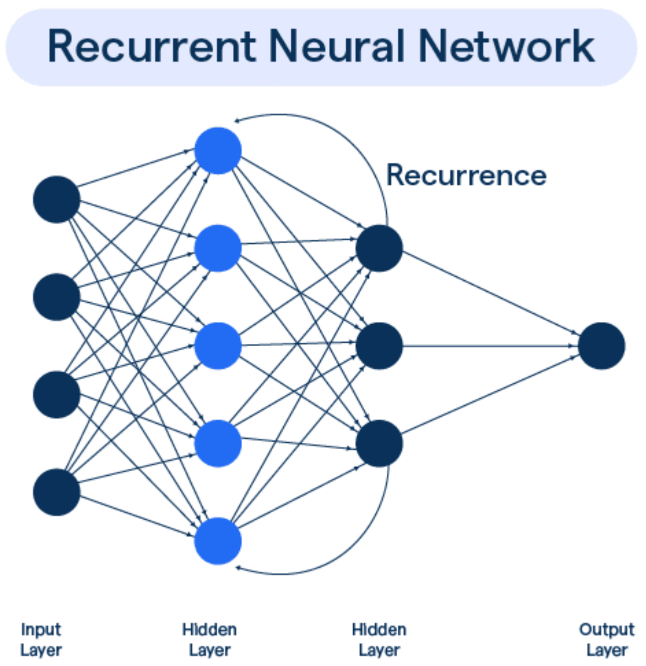
- Multilayer neural networks The neural network which contains input layers, output layers and some hidden layers also is called multilayer neural network.



Network Architectures

Feedback networks (Recurrent networks:)

- Feedback networks can have signals traveling in both directions by introducing loops in the network.
    - very powerful
    - extremely complicated.
    - dynamic: Their 'state' is changing continuously until they reach an equilibrium point.
- also known as interactive or recurrent.



Applications of Neural Network

- Speech recognition
- Optical character recognition
- Face Recognition
- Pronunciation (NETtalk)
- Stock-market prediction
- Navigation of a car
- Signal processing/Communication

Natural Language Processing

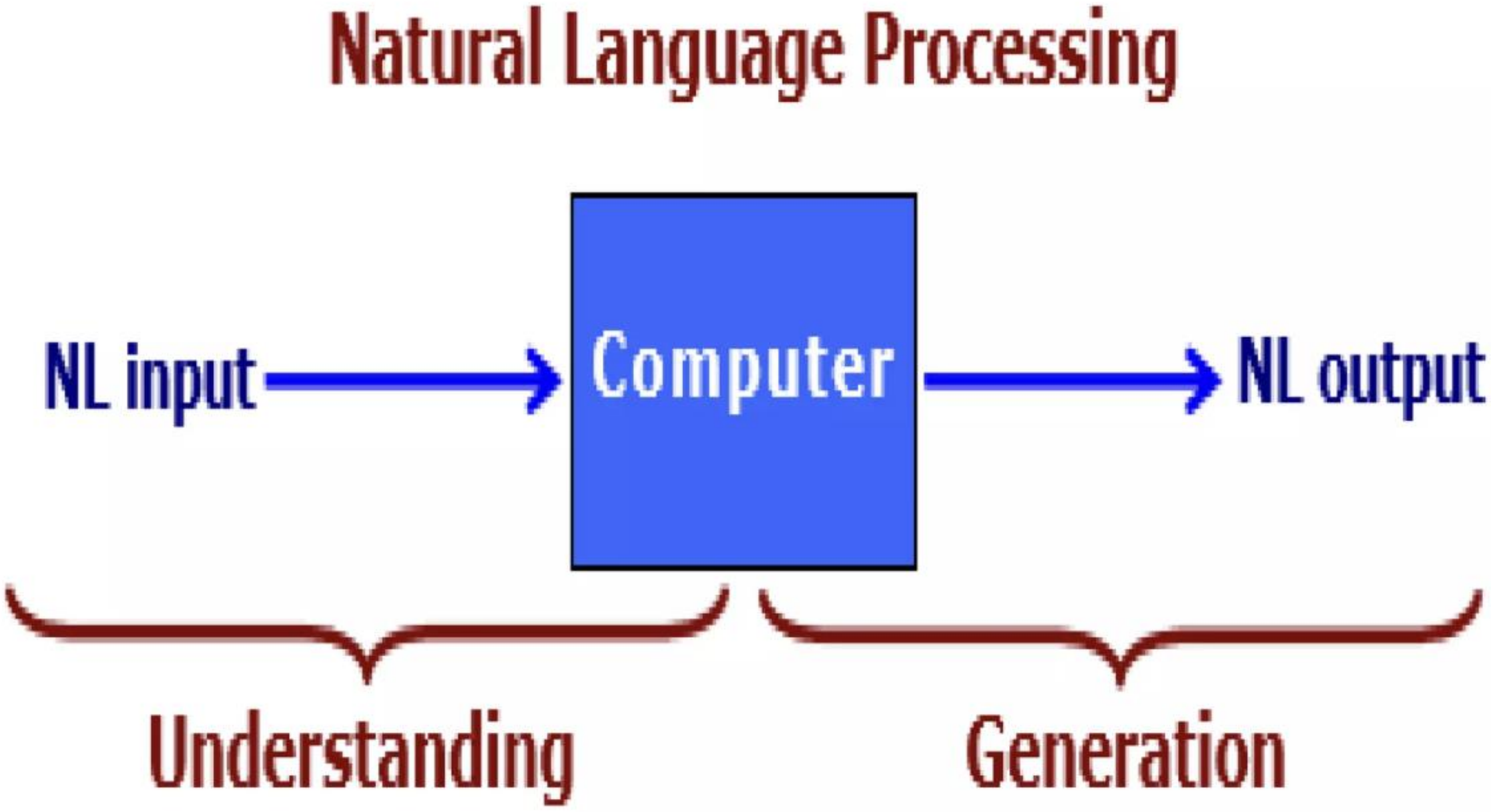
- **What is Natural language:**
- Natural language refers to the way we, humans, communicate with each other.
- English, Spanish, French, and Nepali are all examples of a natural language.
- Natural communication takes place in the form of speech and text

What is Natural Language Processing

- Natural language processing is a technology which involves converting spoken or written human language into a form which can be processed by computers, and vice versa.
- The goal of NLP is to make interactions between computers and humans feel exactly like interactions between humans and humans

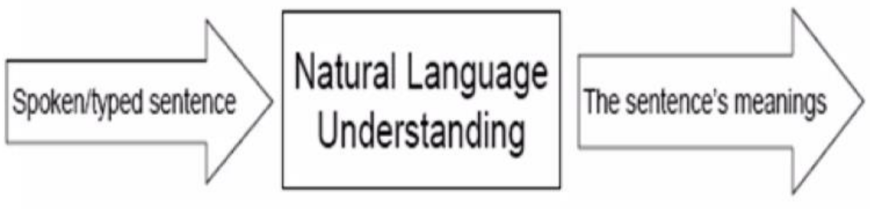
Components of NLP

- Because computers operate on artificial languages, they are unable to understand natural language. This is the problem that NLP solves.
- With NLP, a computer is able to listen to a natural language being spoken by a person, understand the meaning of it, and then if needed, respond to it by generating natural language to communicate back to the person.
- There are two components of NLP as given
  - Natural Language Understanding (NLU)
  - Natural Language Generation (NLG)


Natural Language Processing
NL input
Computer
NL output
Understanding
Generation

Natural Language Understanding(NLU)

- The most difficult part of NLP is understanding, or providing meaning to the natural language that the computer received.
- To understand a language a system should have understanding about:
  - structures of the language including what the words are and how they combine into phrases and sentences.
  - the meanings of the words and how they contribute to the meanings of the sentence and
  - context within which they are being use



Natural Language Understanding

- But, Developing programs that understand a natural language is a difficult and challenging task in the AI due to the following reasons:
  - Natural languages are large: They contain infinity of different sentences. No matter how many sentences a person has heard or seen, new ones can always be produced.
  - much ambiguity in a natural language:
    - Many words have several meanings such as can, bear, bank etc, and
    - sentences have different meanings in different contexts.
      - Example:- a can of juice. I can do it

Natural Language Generation(NLG)

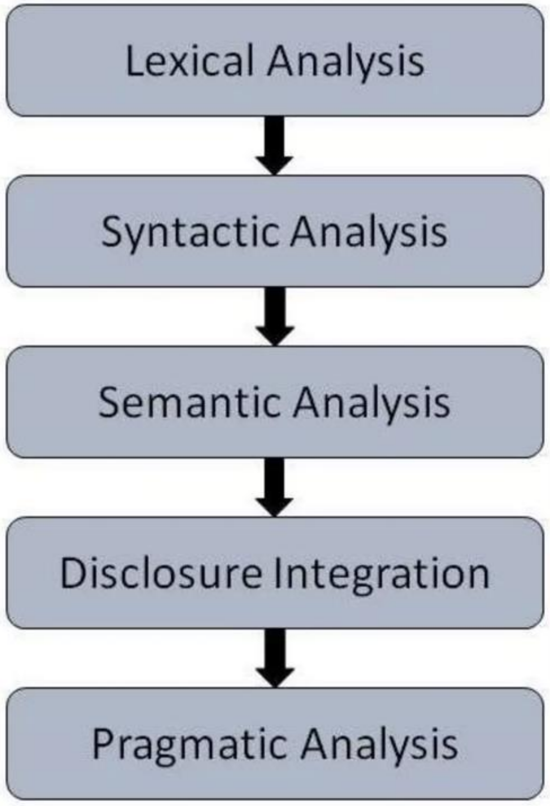
- NG is much simpler to accomplish. NLG translates a computer's artificial language into text, and can also go a step further by translating that text into audible speech with text-to- speech.

NLG v/s NLU

- NG may be viewed as the opposite NLU.
- **The difference can be :**
  - in natural language understanding the system needs to disambiguate the input sentence to produce the machine representation language,.
  - in NLG the system needs to make decisions about how to put a concept into words.
  - The NLU is harder than NLG

Steps in Natural Language Processing

- There are general five steps as shown in figure below:



Contd..

- **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 text into paragraphs, sentences, and words

Contd..

- **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

Contd..

- **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"

Contd..

- **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

Contd..

- **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

Applications of NLP

- Text-to-Speech & Speech recognition
- Information Retrieval
- Document Classification
- Automatic Summarization
- Machine Translation
- Story understanding systems
- Question answering
- Spelling correction
- Caption Generation

Thank You