Introduction to soft Computing and Neural Network

Dr. Bhakti Palkar

Content

1.1 Concept of computing systems, "Soft" computing versus "Hard" computing, Characteristics of Soft computing, Some applications of Soft computing techniques.

1.2 Biological neurons and its working, ANN – Terminologies, Basic Models, Linearly and non-linearly separable classification, McCulloch Pitts Neuron Model

Concept of Computing

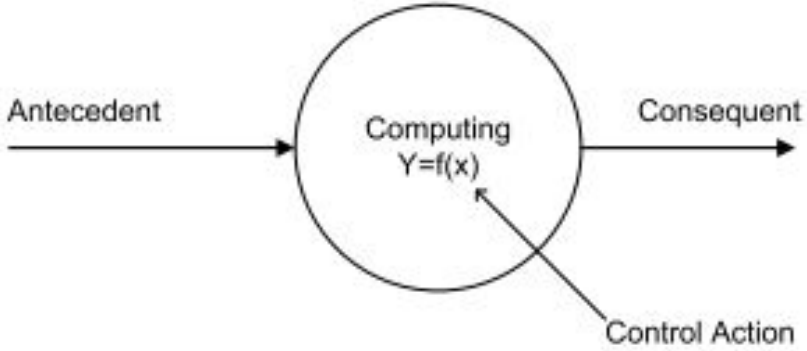


Figure : Basic of computing

$y = f(x)$, f is a mapping function
f is also called a formal method or an algorithm to solve a problem.

Important Characteristics of Computing

1. Should provide **precise solution**.
2. Control action should be **unambiguous** and **accurate**.
3. Suitable for problem, which is **easy to model** mathematically.

Hard Computing

In 1996, LA Zade (LAZ) introduced the term hard computing.

According to LAZ: We term a computing as ”Hard” computing, if

1. Precise result is guaranteed
2. Control action is unambiguous
3. Control action is formally defined (i.e. with mathematical model or algorithm)

Examples of Hard Computing

Example:

Solving numerical problems (e.g. Roots of polynomials, Integration etc.) I

Searching and sorting techniques

Solving ”Computational Geometry” problems (e.g. Shortest tour in Graph theory, Finding closest pair of points etc.)

Soft Computing

- The term soft computing was proposed by the inventor of fuzzy logic, Lotfi A.Zadeh. He describes it as follows.

Definition: Soft computing

Soft computing is a collection of methodologies that aim to exploit the tolerance for imprecision and uncertainty to achieve tractability, robustness and low solution cost. It principal constituents are fuzzy logic, neuro-computing and probablistic reasoning. The role model for soft computing is human mind.

Soft Computing

- It does not require any mathematical modeling of problem solving
- It may not yield the precise solution
- Algorithms are adaptive (i.e. it can adjust to the change of dynamic environment)
- Use some biological inspired methodologies such as genetics, evolution, Ant’s behaviors, particles swarming, human nervous systems etc.

Soft computing

- Soft Computing is the fusion of methodologies designed to model and enable solutions to real world problems, which are not modeled or too difficult to model mathematically.
- Soft Computing is a term used in computer science to refer to problems in whose solutions are unpredictable, uncertain and between 0 and 1.

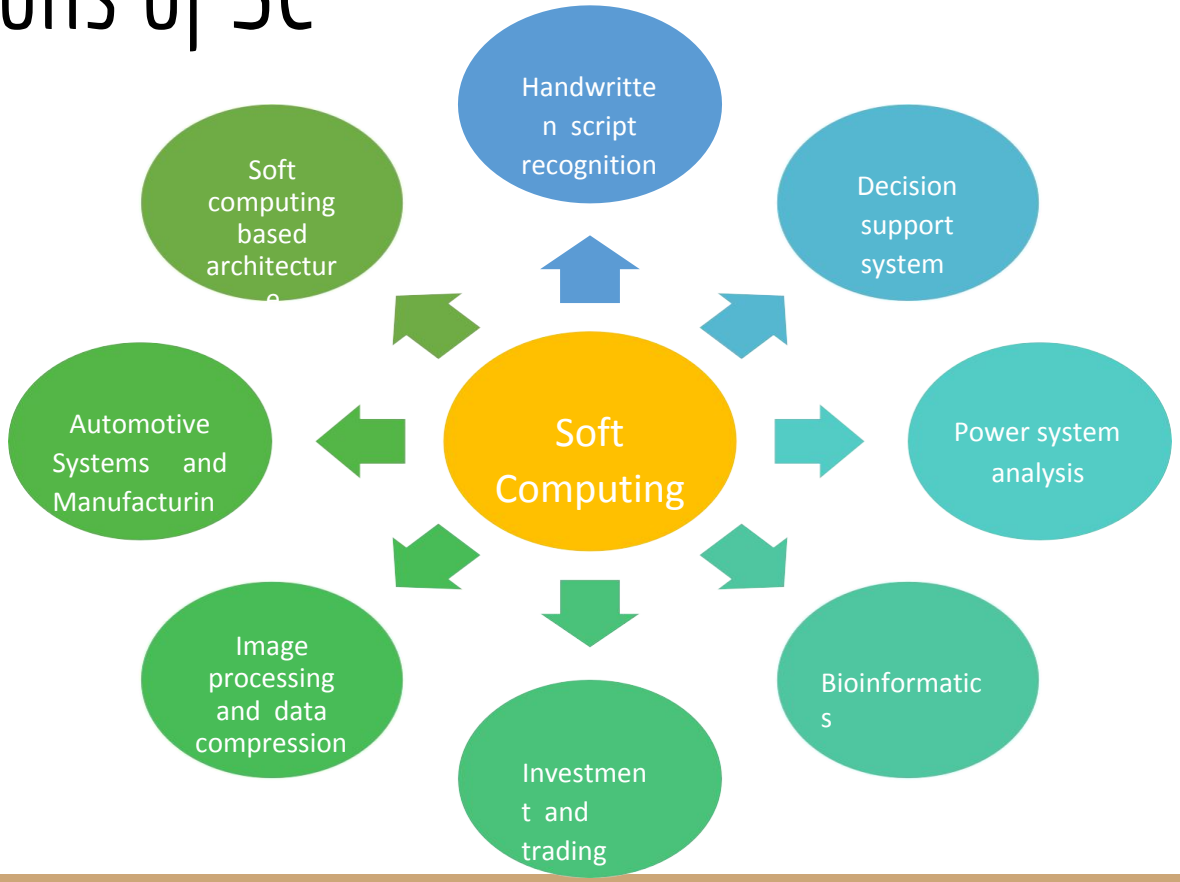
Goals of soft computing

- To develop intelligent machines to provide solutions to real world problems, which are not modelled or too difficult to model mathematically.
- Well suited for real world problems where ideal solutions are not there.
- To exploit the tolerance for approximation, uncertainty, imprecision and partial truth in order to achieve close resemblance with human like decision making.

contd..

- imprecision – the model features (quantities) are not the same as that of the real ones, but close to them.
- uncertainty – we are not sure that the features of the model are the same as that of the entity (belief).
- Approximate Reasoning – the model features are similar to the real ones, but not the same.
- The guiding principle of soft computing is to exploit these tolerance to achieve ***tractability, robustness and low solution cost.***
- The role model for soft computing is the human mind.

Applications of SC



Difference

Sr. No.	Hard Computing	Soft Computing
1	It requires precisely stated analytic model	SC techniques are tolerant of imprecision, uncertainty, partial truth and approximation
2	based on binary logic, crisp system, numerical analysis and crisp software	based on fuzzy logic, neural sets, and probabilistic reasoning
3	uses two-valued logic.	can use multivalued or fuzzy logic
4	is deterministic	incorporates stochasticity
5	requires exact input data	can deal with ambiguous and noisy data
6	is strictly sequential	allows parallel computations
7	produces precise answers.	can yield approximate answers
8	Not fault tolerant	Fault tolerant

Techniques in soft computing

1. Neural Network
2. Fuzzy Logic
3. Genetic Algorithm
4. Hybrid Systems

Neural Network

DARPA Neural Network Study (1988, AFCEA International Press, p. 60):

... a neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes.

Neural Network

According to Haykin (1994), p. 2:

A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

Knowledge is acquired by the network through a learning process.

Interneuron connection strengths known as synaptic weights are used to store the knowledge

Neural Network

According to Nigrin (1993), p. 11:

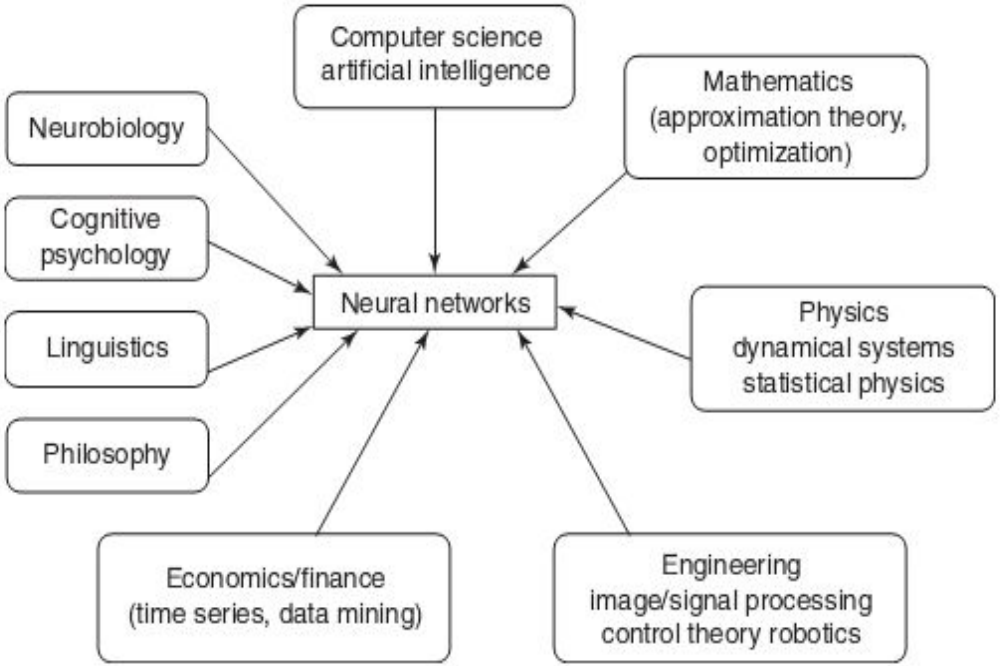
A neural network is a circuit composed of a very large number of simple processing elements that are neurally based. Each element operates only on local information.

Furthermore each element operates asynchronously; thus there is no overall system clock.

According to Zurada (1992):

Artificial neural systems, or neural networks, are physical cellular systems which can acquire, store and utilize experiential knowledge.

Multidisciplinary view of neural network



Fuzzy Logic

- It was developed in 1965, by Professor Lofti Zadeh, at University of California, Berkley. The first application was to perform computer data processing based on natural values.
- In more simple words, **A Fuzzy logic stat can be 0, 1 or in between these numbers i.e. 0.17 or 0.54**.
- **For example,** In **Boolean**, we may say glass of hot water (i.e 1 or High) or glass of cold water i.e. (0 or low), **but in Fuzzy logic,** We may say glass of warm water (neither hot nor cold).

Advantages of Fuzzy Logic

1. A Fuzzy Logic System is flexible and allow modification in the rules.
2. Even imprecise, distorted and error input information is also accepted by the system.
3. The systems can be easily constructed.
4. Since these systems involve human reasoning and decision making, they are useful in providing solutions to complex solutions in different types of applications.

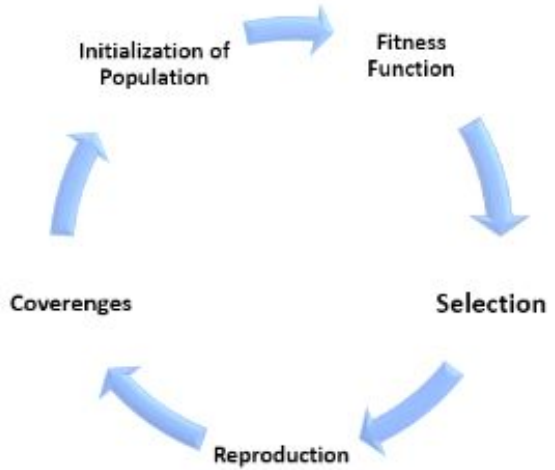
Applications of Fuzzy Logic

- Fuzzy Logic system can be used in Automotive systems, for applications like 4-Wheel steering, automatic gearboxes etc.
- Applications in the field of Domestic Applications include Microwave Ovens, Air Conditioners, Washing Machines, Televisions, Refrigerators, Vacuum Cleaners etc.
- Other applications include Hi-Fi Systems, Photo-Copiers, Humidifiers etc.

Genetic Algorithm

- A genetic algorithm is an adaptive heuristic search algorithm inspired by "Darwin's theory of evolution in Nature."
- In GAs, we have a **pool or a population of possible solutions** to the given problem. These solutions then undergo recombination and mutation (like in natural genetics), producing new children, and the process is repeated over various generations.
- During this procedure a certain strings of symbols, known as chromosomes, evaluate toward better solution.

Genetic Algorithm



- In this way we keep “evolving” better individuals or solutions over generations, till we reach a stopping criterion.

Applications of Genetic Algorithms

- Data mining and clustering.
- Image processing.
- Wireless sensor network.
- Traveling salesman problem (TSP)
- Vehicle routing problems.
- Mechanical engineering design.
- Manufacturing system.

Hybrid Computing

It is a combination of the conventional hard computing and emerging soft computing

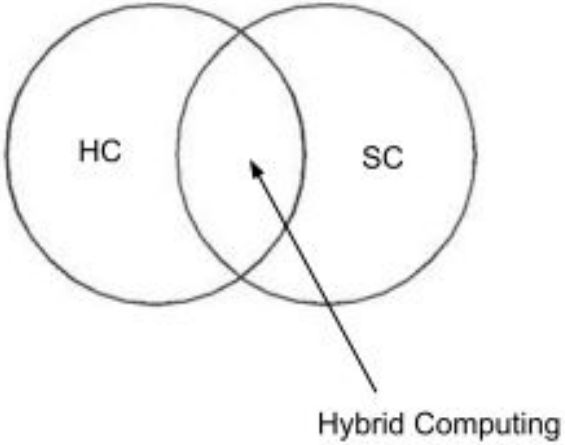


Figure : Concept of Hybrid Computing

Hybrid Systems

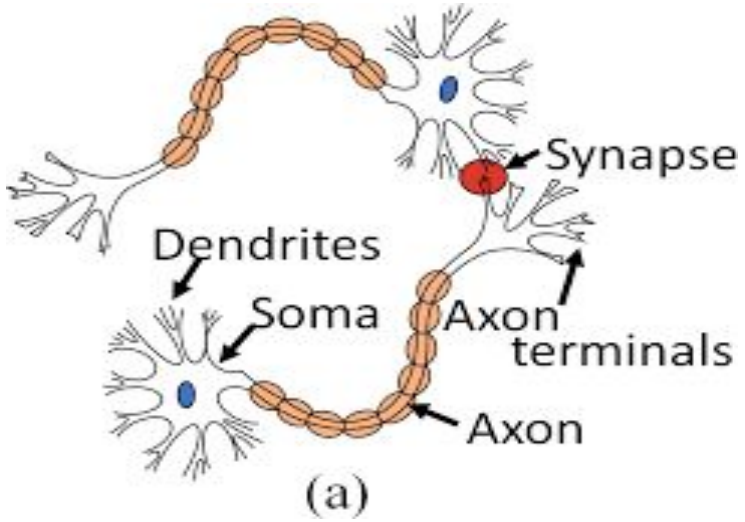
- Hybrid systems enables one to combine various soft computing paradigms and result in a best solution. The major three hybrid systems are as follows:
- Hybrid Fuzzy Logic (FL) Systems
- Hybrid Neural Network (NN) Systems
- Hybrid Evolutionary Algorithm (EA) Systems

Neural Network

- Biological nervous system **is the most important part** of many living things, in particular, human beings.
- There is a part called **brain** at the center of human nervous system.
- In fact, any biological nervous system consists of a large number of interconnected processing units called **neurons**.
- Each neuron is approximately 10μm long and they can operate in parallel.
- Typically, a human brain consists of approximately 10^{11} neurons communicating with each other with the help of **electrical impulses**.

Neuron:Basic unit of nervous system

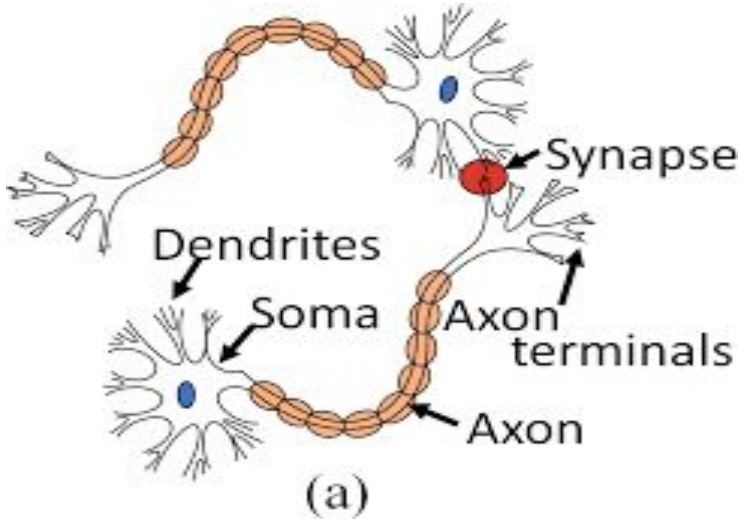
- **Dendrite** : A bush of very thin fibre.
- **Axon** : A long cylindrical fibre.
- **Soma** : It is also called a cell body, and just like as a nucleus of cell.
- **Synapse** : It is a junction where axon makes contact with the dendrites of neighbouring dendrites.



Different parts of a biological neuron

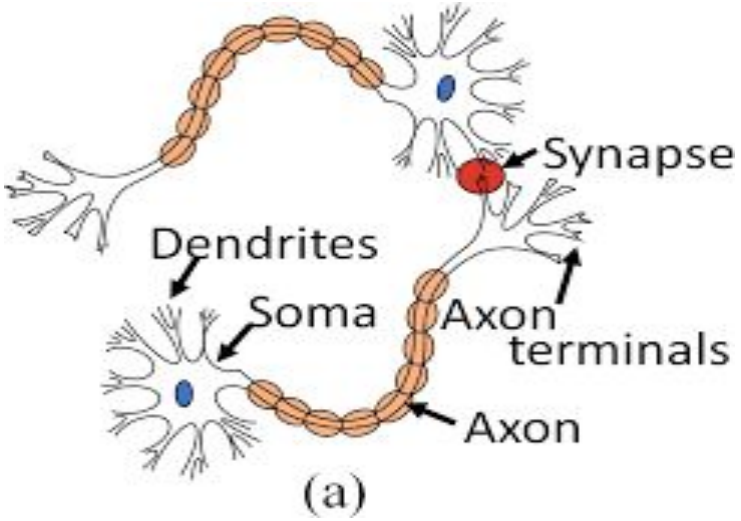
Neuron and its working

- **Dendrite** (Input): receives signals from other neurons
- **Soma** (cell body – processing unit): sums up all signals. It also consists of threshold unit.



Neuron and its working

- **Synapse** (weighted connections): The point of interconnection of one neuron with other neurons. The amount of signal transmitted depends upon the strength (synaptic weight) of the connection. Connection can be Inhibitory (decreasing strength) or Excitatory (increasing strength)
- **Axon** (output): when the sum reaches to the threshold value, neuron fires and generates an output signal.



Biological Neural Network

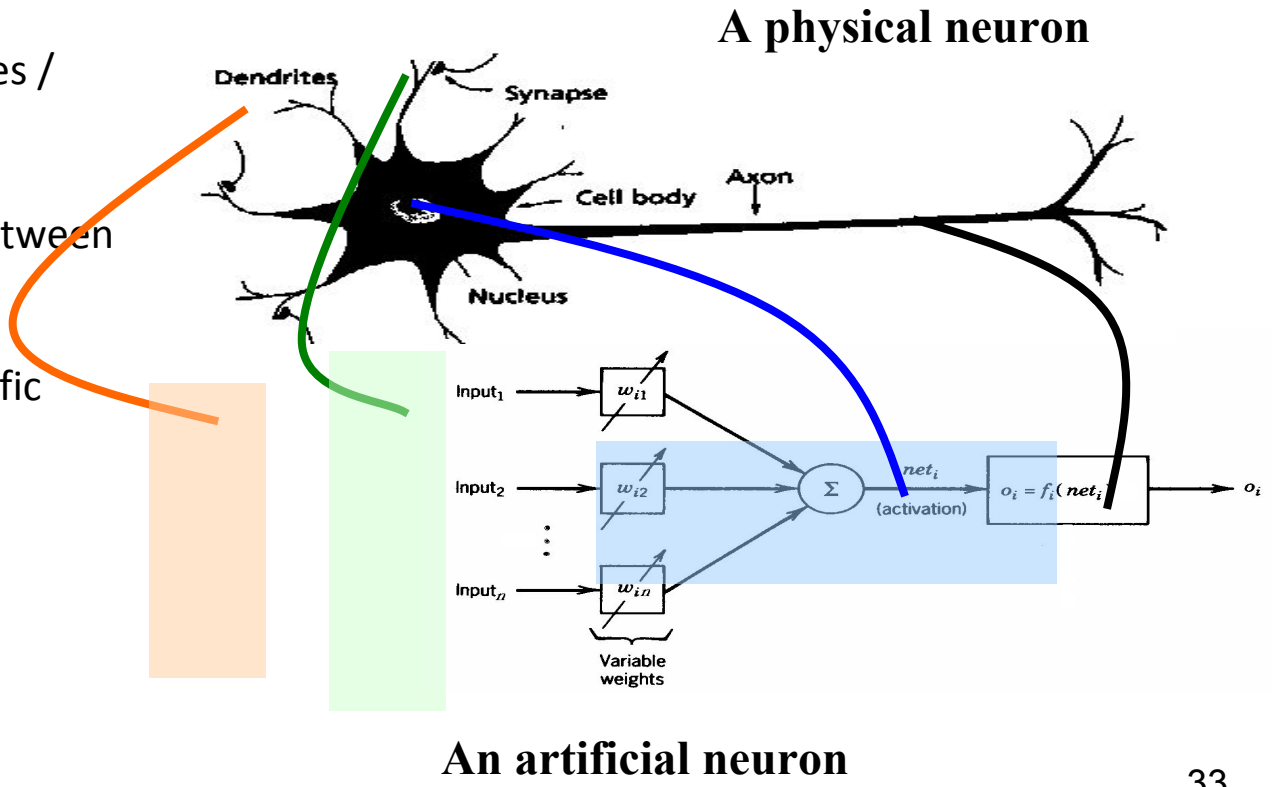
- *Has three main parts*
  - *Soma or cell body-where cell nucleus is located*
  - *Dendrites-where the nerve is connected to the cell body*
  - *Axon-which carries the impulses of the neuron*
- *Electric impulse is passed between synapse and dendrites.*
- *Synapse- Axon split into strands and strands terminates into small bulb like organs called as synapse.*
- *It is a chemical process which results in increase /decrease in the electric potential inside the body of the receiving cell.*
- *If the electric potential reaches a thresh hold value, receiving cell fires & pulse / action potential of fixed strength and duration is send through the axon to synaptic junction of the cell.*
- *After that, cell has to wait for a period called* ***refractory period.***

Artificial Neural Network

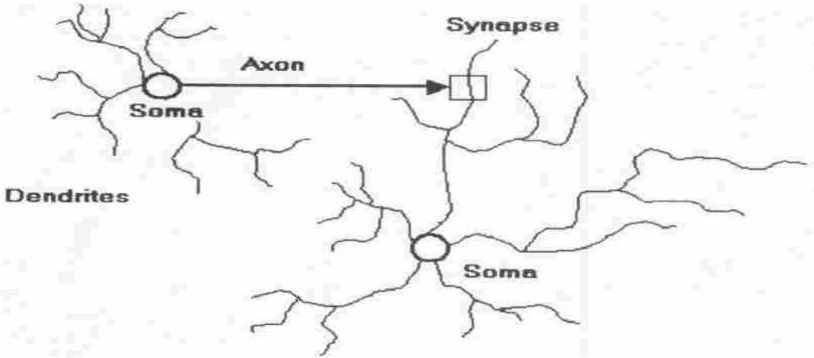
- ANN is an information processing system that has certain performance characteristics in common with biological nets.
- Several key features of the processing elements of ANN are suggested by the properties of biological neurons:
  - The processing element receives many signals.
  - Signals may be modified by a weight at the receiving synapse.
  - The processing element sums the weighted inputs.
  - Under appropriate circumstances (sufficient input), the neuron transmits a single output.
  - The output from a particular neuron may go to many other neurons.

Artificial Neurons

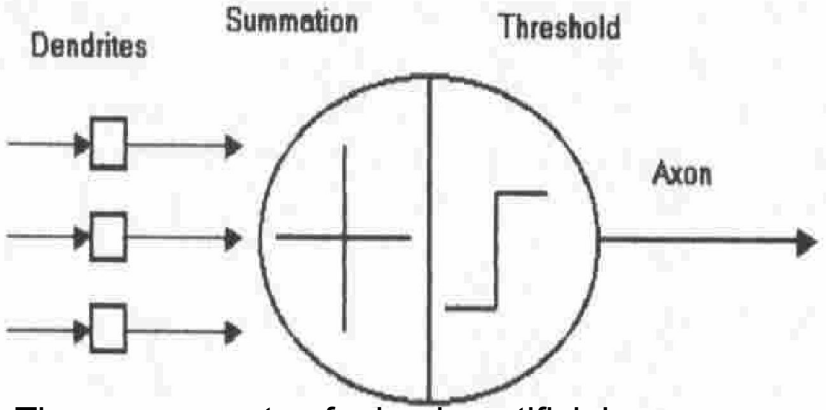
- From experience: examples / training data
- Strength of connection between the neurons is stored as a weight-value for the specific connection.
- Learning the solution to a problem = changing the connection weights



Artificial Neurons



Four basic components of a human biological neuron

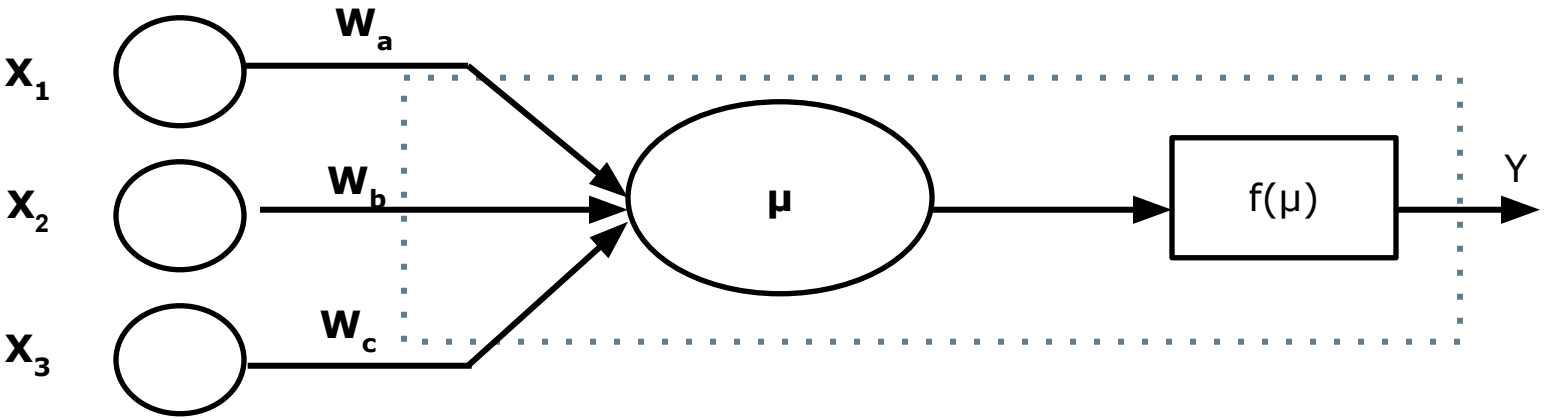


The components of a basic artificial neuron

Terminology Relation Between Biological And Artificial Neuron

Biological Neuron	Artificial Neuron
Cell	Neuron
Dendrites	Weights or interconnections
Soma	Net input
Axon	Output

Model Of A Neuron



Input units

Connection weights

Summing function

computation

(dendrite)

(synapse)

(soma)

(axon)

ANN

- A neural net consists of a large number of simple processing elements called neurons, units, cells or nodes.
- Each neuron is connected to other neurons by means of directed communication links, each with associated weight.
- The weight represent information being used by the net to solve a problem.
- Each neuron has an internal state, called its activation or activity level, which is a function of the inputs it has received. Typically, a neuron sends its activation as a signal to several other neurons.

Advantages of Neural Networks

- A Neural Network can be an ***"expert"*** in analyzing the category of information given to it.
- Answers " what-if" questions
- Adaptive learning-Ability to learn how to do tasks based on the data given for training or initial experience.
- Self organization-Creates its own organization or representation of information it receives during learning time.
- Real time operation-Computations can be carried out in parallel.
- Fault tolerance via redundant information coding
  - Partial destruction of neural network cause degradation of performance.
  - In some cases, it can be retained even after major network damage.
- In future, it can also used to give spoken words as instructions for machine.

Application Scope of Neural Networks

- Air traffic control
- Animal behavior
- Appraisal and valuation of property, etc.,
- Betting on horse races, stock markets
- Criminal sentencing
- Complex physical and chemical process
- Data mining, cleaning and validation
- Direct mail advertisers
- Echo patterns
- Economic modeling
- Employee hiring
- Expert consultants
- Fraud detection
- Hand writing and typewriting
- Lake water levels
- Machinery controls
- Medical diagnosis
- Music composition
- Photos and finger prints
- Recipes and chemical formulation
- Traffic flows
- Weather prediction

Brain Vs Computer

Term	Brain	Computer
Speed	Execution time is few milliseconds	Execution time is few nano seconds
Processing	Perform massive parallel operations simultaneously	Perform several parallel operations simultaneously. It is faster than the biological neuron
Size and complexity	Number of Neuron is 10^{11} and number of interconnections is 10^{15}. So complexity of brain is higher than computer	It depends on the chosen application and network designer.
Storage capacity	i) Information is stored in interconnections or in synapse strength. ii) New information is stored without destroying old one. iii) Sometimes fails to recollect information	i) Stored in continuous memory location. ii) Overloading may destroy older locations. iii) Can be easily retrieved

Contd...

Term	Brain	Computer
Tolerance	i) Fault tolerant ii) Store and retrieve information even interconnections fails iii) Accept redundancies	i) No fault tolerance ii) Information corrupted if the network connections disconnected. iii) No redundancies
Control mechanism	Depends on active chemicals and neuron connections are strong or weak	CPU Control mechanism is very simple

Evolution of neural networks

Year	Neural network	Designer	Description
1943	McCulloch and Pitts neuron	McCulloch and Pitts	Arrangement of neurons is combination of logic gate. Unique feature is thresh hold
1949	Hebb network	Hebb	If two neurons are active, then their connection strengths should be increased.
1958,1959,1962,1988,1960	Perceptron Adaline	Frank Rosenblatt, Block, Minsky and Papert Widrow and Hoff	Weights are adjusted to reduce the difference between the net input to the output unit and the desired output

Contd...

Year	Neural network	Designer	Description
1972	Kohonen self-organizing feature map	Kohonen	Inputs are clustered to obtain a fired output neuron.
1982, 1984, 1985, 1986, 1987	Hopfield network	John Hopfield and Tank	Based on fixed weights. Can act as associative memory nets
1986	Back propagation network	Rumelhart, Hinton and Williams	i) Multilayered ii) Error propagated backward from output to the hidden units

Contd..

1988	Counter propagation network	Grossberg	Similar to kohonen network
1987-1990	Adaptive resonance Theory(ART)	Carpenter and Grossberg	Designed for binary and analog inputs.
1988	Radial basis function network	Broomhead and Lowe	Resemble back propagation network , but activation function used is Gaussian function
1988	Neo cognitron	Fukushima	For character recogniton.

Basic models of ANN

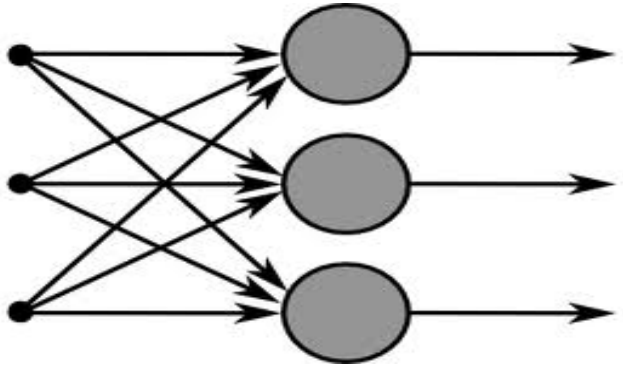
- Models are based on three entities
  - ***The model's synaptic interconnections.***
  - ***The training or learning rules adopted for updating and adjusting the connection weights.***
  - ***Their activation functions***
- The *arrangement of neurons to form layers* and the *connection pattern formed within and between layers* is called the ***network architecture.***

Five types of ANN

1. ***Single layer feed forward network***
2. ***Multilayer feed-forward network***
3. ***Single node with its own feedback***
4. ***Single-layer recurrent network***
5. ***Multilayer recurrent network***

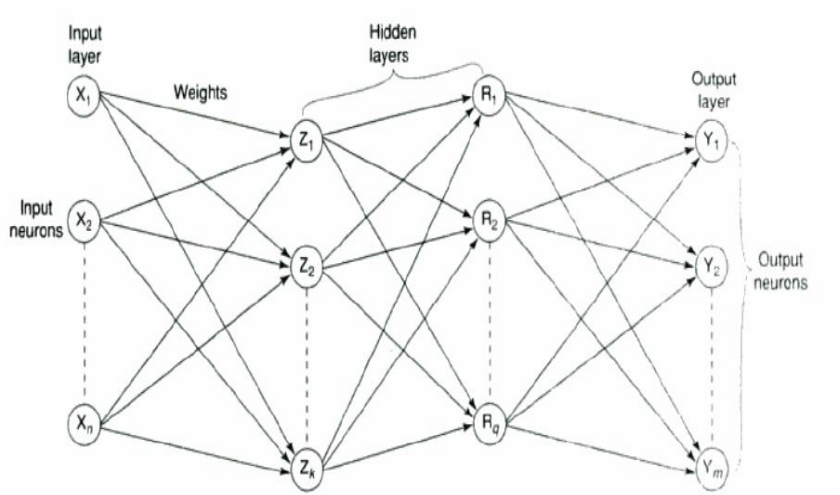
Single layer Feed- Forward Network

- Layer is formed by taking processing elements and combining it with other processing elements.
- Input and output are linked with each other
- Inputs are connected to the processing nodes with various weights, resulting in series of outputs one per node.



Multilayer feed-forward network

- Formed by the interconnection of several layers.
- Input layer receives input and buffers input signal.
- Output layer generates output.
- Layer between input and output is called *hidden layer*.
- Hidden layer is internal to the network.
- Zero to several hidden layers in a network.
- More the hidden layer, more is the complexity of network, but efficient output is produced.



Feed back network

- If no neuron in the output layer is an input to a node in the same layer / proceeding layer – ***feed forward network.***
- If outputs are directed back as input to the processing elements in the same layer/proceeding layer –***feedback network.***
- If the output are directed back to the input of the same layer then it is ***lateral feedback.***

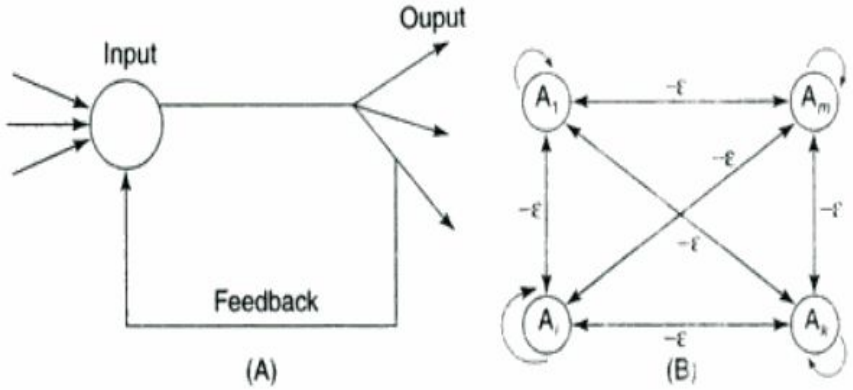


Figure 2-8 (A) Single node with own feedback. (B) Competitive nets.

Recurrent networks are networks with feedback networks with closed loop.
Fig 2.8 (A) –simple recurrent neural network having a single neuron with feedback to itself.

Single layer recurrent layer

Processing element output can be directed to processing element itself or to other processing element in the same layer.

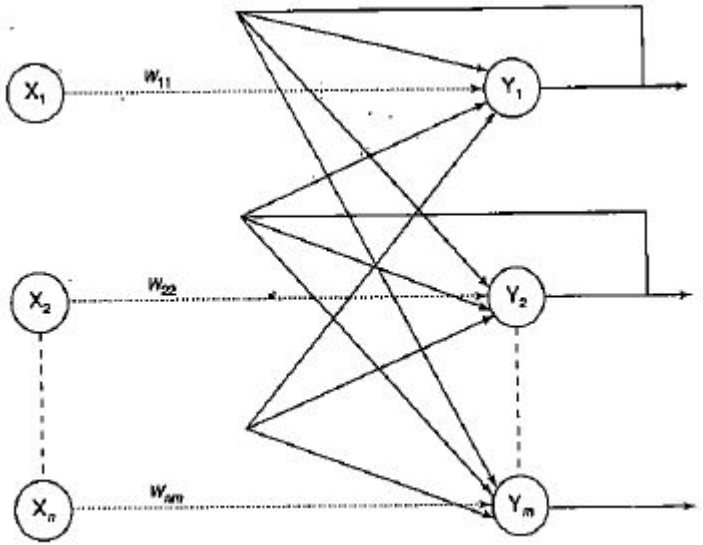


Figure 2-9 Single-layer recurrent network.

Multilayer recurrent network

Processing element output can be directed back to the nodes in the preceding layer, forming a ***multilayer recurrent network.***

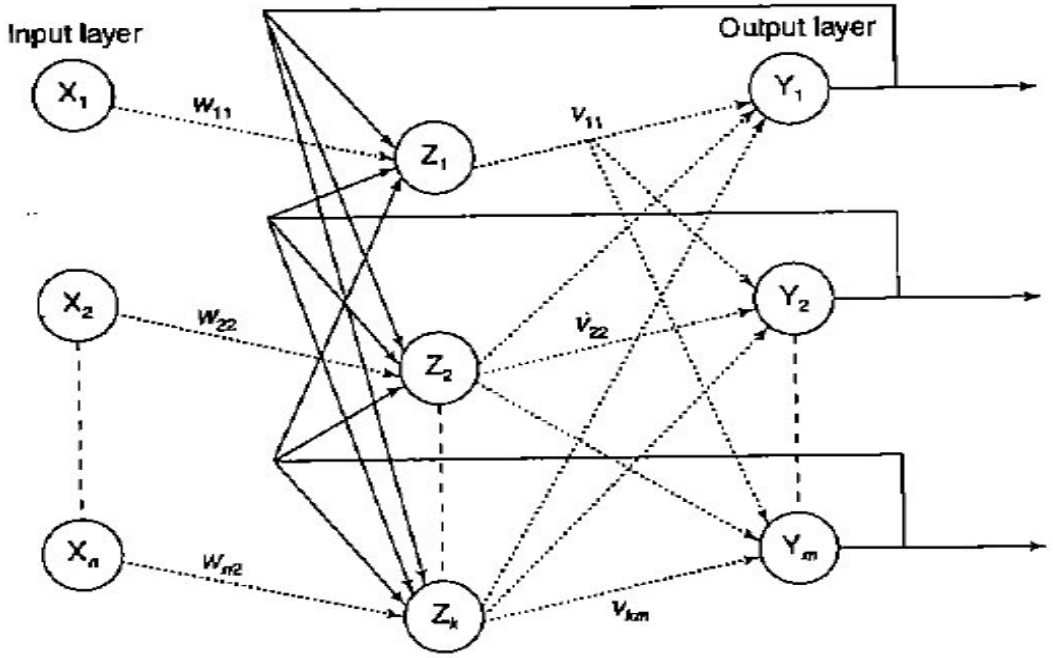


Figure 2-10 Multilayer recurrent network.

Learning

Two broad kinds of learning in ANNs is :

i) parameter learning – updates connecting weights in a neural net.

ii) Structure learning – focus on change in the network.

Apart from these, learning in ANN is classified into three categories as

i) supervised learning

ii) unsupervised learning

Iii) reinforcement learning

Supervised learning

In ANN, each input vector requires a corresponding target vector, which represents the desired output.

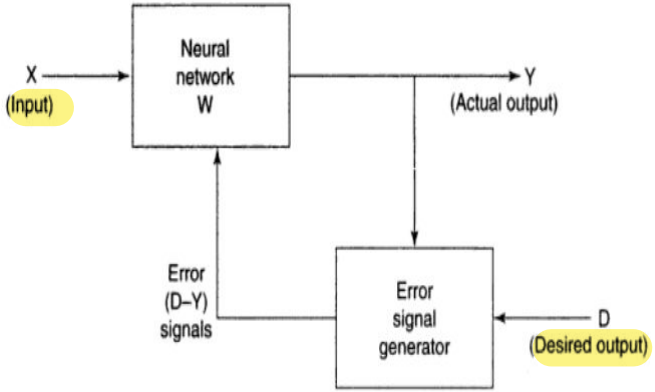
The input vector along with target vector is called ***training pair.***

The input vector results in output vector.

The actual output vector is compared with desired output vector.

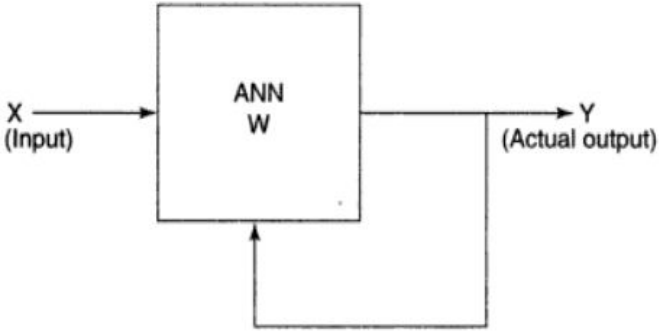
If there is a difference means an error signal is generated by the network.

It is used for adjustment of weights until actual output matches desired output.



Unsupervised learning

- Learning is performed without the help of a teacher.
- Example: tadpole – learn to swim by itself.
- In ANN, during training process, network receives input patterns and organize it to form clusters.
- From the Fig. it is observed that no feedback is applied from environment to inform what output should be or whether they are correct.
- The network itself discover patterns, regularities, features/ categories from the input data and relations for the input data over the output.
- Exact clusters are formed by discovering similarities & dissimilarities so called as *self – organizing.*



Reinforcement learning

- Similar to supervised learning.
- For example, the network might be told that its actual output *is* only "50% correct" or so. Thus, here only critic information is available, nor the exact information.
- Learning based on *critic information* is called *reinforcement learning* & the feedback sent is called *reinforcement signal.*
- The network receives some feedback from the environment.
- Feedback is only evaluative.

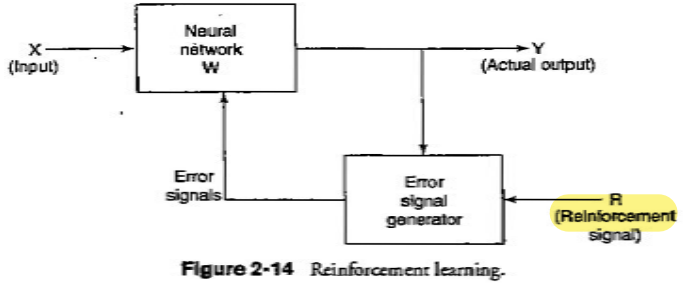


Figure 2-14 Reinforcement learning.

The external reinforcement signals are processed in the critic signal generator, and the obtained critic signals are sent to the ANN for adjustment of weights properly to get critic feedback in future.

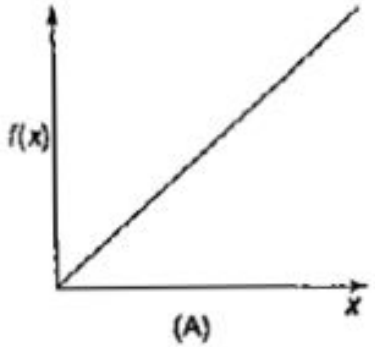
Activation functions

- To make work more efficient and for exact output, some force or activation is given.
- Activation function is applied over the net input to calculate the output of an ANN. Information processing of processing element has two major parts: input and output. An integration function (f) is associated with input of processing element.

Activation functions

1. **Identity function:**
   - it is a linear function which is defined as

     $f(x) = x$ *for all* x
   - The output is same as the input.



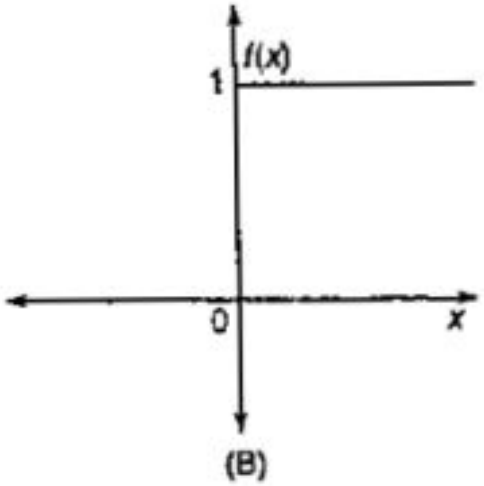
Activation functions

2. Binary step function

$$f(x) = \begin{cases} 1 & if\ x \geq \theta \\ 0 & if\ x < \theta \end{cases}$$

where θ represents thresh hold value.

It is used in single layer nets to convert the input to an output that is binary. (0 or 1)



(B)

Activation functions

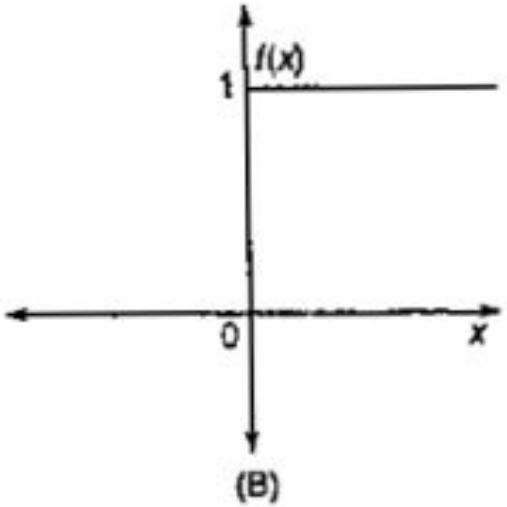
3. Bipolar step function:

It is defined as

$$f(x) = \begin{cases} 1 & if\ x \geq \theta \\ -1 & if\ x < \theta \end{cases}$$

where θ represents threshold value.

Used in single layer nets to convert the net input to an output that is bipolar (+1 or -1).



(B)

Activation functions

4. Sigmoid function:

used in Back propagation nets.

a) ***binary sigmoid function—***

logistic sigmoid function or unipolar sigmoid function.-it is defined as

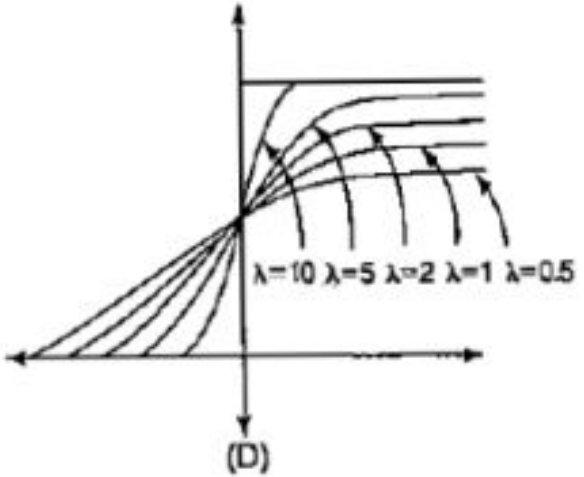
$$f(x) = \frac{1}{1 + e^{-\lambda x}}$$

where λ – steepness parameter.

-The derivative of this function is

f'(x) = λ f(x)[1-f(x)].

The range of sigmoid function is 0 to 1.

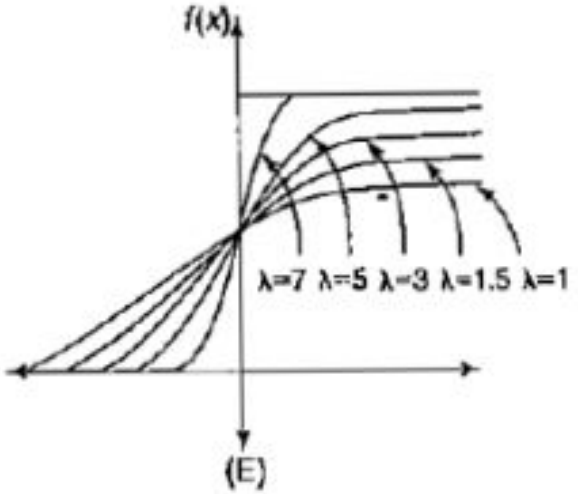


Activation functions

b) *Bipolar sigmoid function*

$$f(x) = \frac{2}{1 + e^{-\lambda x}} - 1 = \frac{1 - e^{-\lambda x}}{1 + e^{-\lambda x}}$$

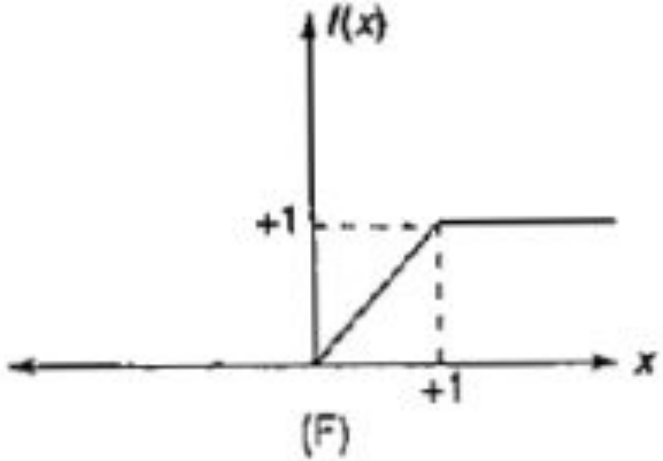
where λ- steepness parameter and the sigmoid range is between -1 and +1.



Activation functions

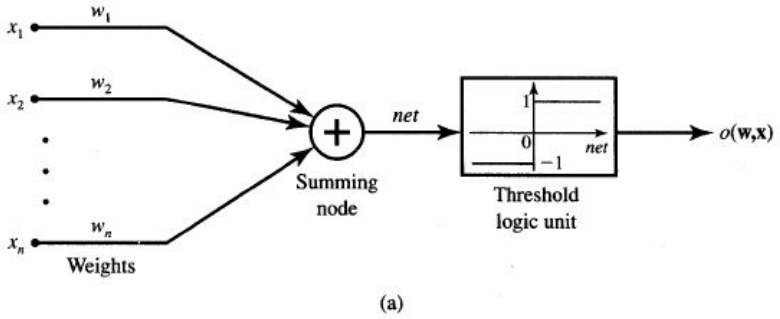
5. Ramp function:

$$f(x) = \begin{Bmatrix} 1 & if\ x > 1 \\ x & if\ 0 \leq x \leq 1 \\ 0 & if\ x < 0 \end{Bmatrix}$$

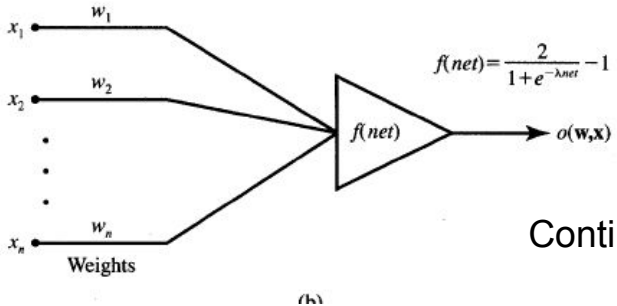


(F)

Common models of neurons



Binary perceptrons



Continuous perceptrons

Weights

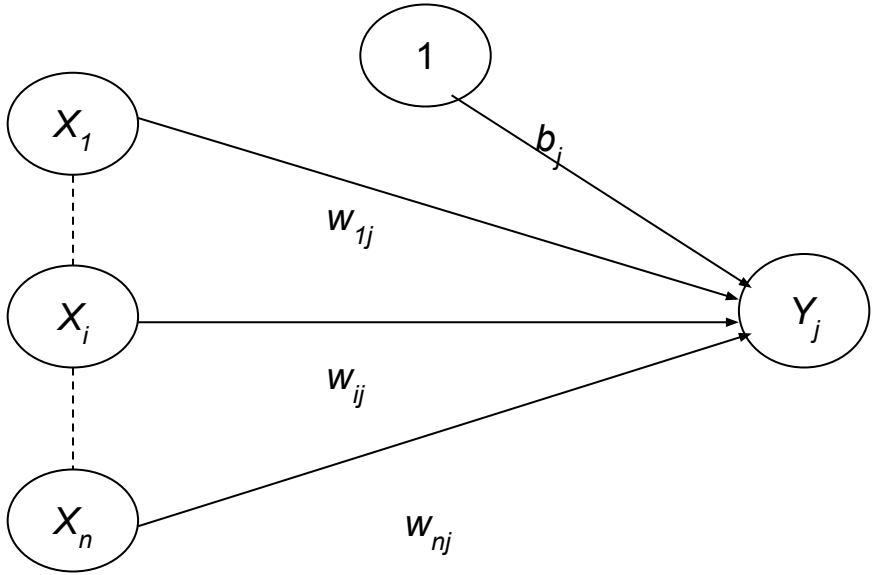
- Each neuron is connected to every other neuron by means of directed links
- Links are associated with weights
- Weights contain information about the input signal and is represented as a matrix
- Weight matrix also called connection matrix

Weight matrix

$$W = \begin{bmatrix} w_1^T \\ w_2^T \\ w_3^T \\ . \\ . \\ . \\ . \\ . \\ w_n^T \end{bmatrix} = \begin{bmatrix} w_{11} w_{12} w_{13} \cdots w_{1m} \\ w_{21} w_{22} w_{23} \cdots w_{2m} \\ \cdots\cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots\cdots \\ w_{n1} w_{n2} w_{n3} \cdots w_{nm} \end{bmatrix}$$

Weights contd...

- w_{ij} _is the weight from processing element *"i"* (source node) to processing element *"j"* (destination node)



$$y_{inj} = \sum_{i=0}^{n} x_i w_{ij}$$

$$= x_0 w_{0j} + x_1 w_{1j} + x_2 w_{2j} + ... + x_n w_{nj}$$

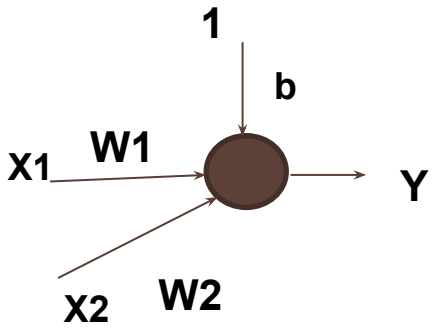
$$= w_{0j} + \sum_{i=1}^{n} x_i w_{ij}$$

$$y_{inj} = b_j + \sum_{i=1}^{n} x_i w_{ij}$$

Bias

- Bias has an impact in calculating net input.
- Bias is included by adding x_0 to the input vector x.
- The net output is calculated by

$$y_{inj} = \sum_{i=0}^{n} x_i w_{ij} \qquad y_{inj} = b_j + \sum_{i=0}^{n} x_i w_{ij}$$



- The bias is of two types
  - Positive bias -Increase the net input
  - Negative bias-Decrease the net input

Threshold

- It is a set value based upon which the final output is calculated.
- Calculated net input and threshold is compared to get the network output.
- The activation function of threshold is defined as

$$f(net) = \begin{cases} 1 & if\ net \geq \theta \\ -1 & if\ net < \theta \end{cases}$$

where θ is the fixed threshold value

Learning rate

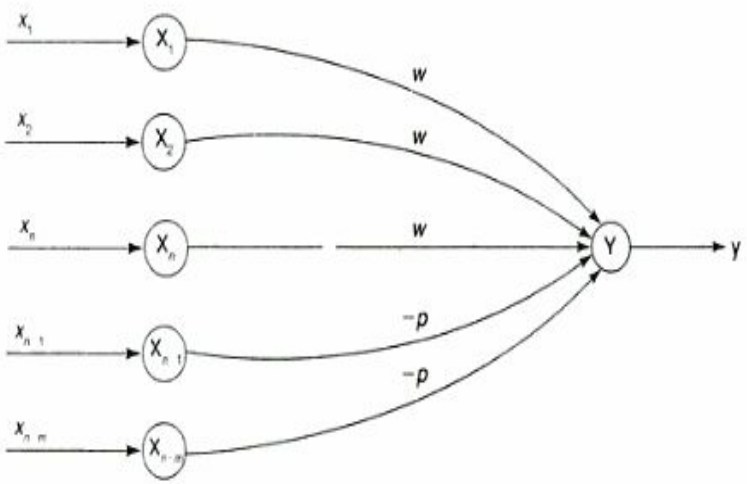
- Denoted by α.
- Used to control the amount of weight adjustment at each step of training
- Learning rate ranging from 0 to 1 determines the rate of learning in each time step

Other terminologies

- Momentum factor:
  - used for convergence when momentum factor is added to weight updation process.
- Vigilance parameter:
  - Denoted by ρ
  - Used to control the degree of similarity required for patterns to be assigned to the same cluster

Mcculloch-pitts neuron

- Discovered in 1943.
- Usually called as *M-P neuron.*
- M-P neurons are connected by directed weighted paths.
- Activation of M-P neurons is binary (i.e) at any time step the neuron may fire or may not fire.
- Weights associated with communication links may be excitatory(wgts are positive)/inhibitory(wgts are negative).
- Threshold plays major role here. There is a fixed threshold for each neuron and if the net input to the neuron is greater than the threshold then the neuron fires.
- They are widely used in logic functions.



- A simple M-P neuron is shown in the figure.
- It is excitatory with weight (*w>0)* / inhibitory with weight *–p (p<0)*.

  In the Fig., inputs from x_1 to x_n possess excitatory weighted connection and X_{n+1} to x_{n+m} has inhibitory weighted interconnections.

  Since the firing of neuron is based on threshold, activation function is defined as

$$f(y_{in}) = \begin{cases} 1 & if\ y_{in} \geq \theta \\ 0 & if\ y_{in} < \theta \end{cases}$$

Mcculloch-pitts neuron

- For inhibition to be absolute, the threshold with the activation function should satisfy the following condition:

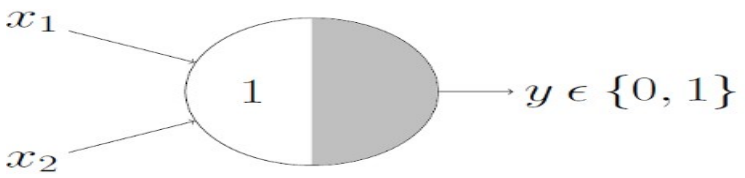
$$\theta > nw - p$$

- Output will fire if it receives "k" or more excitatory inputs but no inhibitory inputs where

$$kw \geq \theta > (k-1)\, w$$

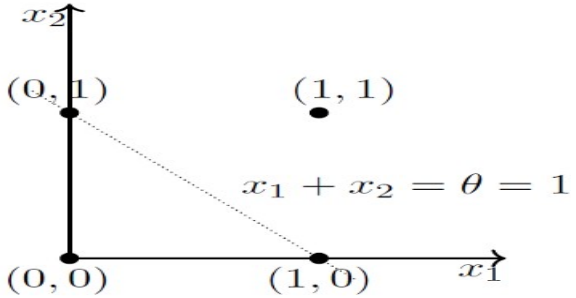
- The M-P neuron has no particular training algorithm.
- An analysis is performed to determine the weights and the threshold.
- It is used as a building block where any function or phenomenon is modeled based on a logic function.

Geometrical interpretation of OR function MP-neuron



OR function

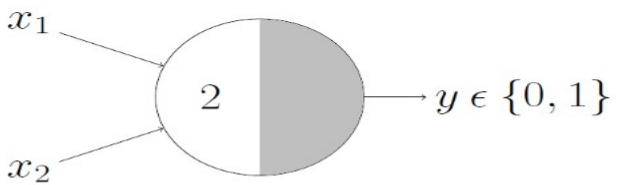
$$x_1 + x_2 = \sum_{i=1}^{2} x_i \geq 1$$



- A single MP neuron splits the input points (4points for 2 binary inputs) into two halves
- Points lying on or above the line $\sum_{i-1}^{n} x_i$-θ =0 and points lying below this line.

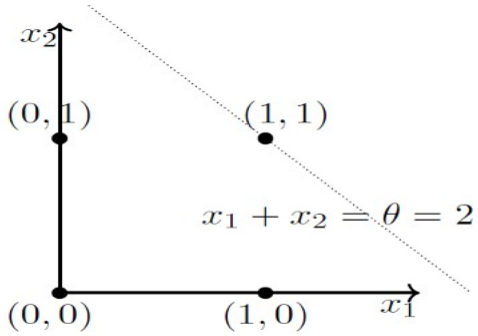
- all inputs which produce an output 0 will be on one side ($\sum_{i=1}^{n} x_i$< θ) of the line and all inputs which produce an output 1 will lie on the other side ($\sum_{i=1}^{n} x_i$> θ) of this line

Geometrical interpretation of AND function MP-neuron

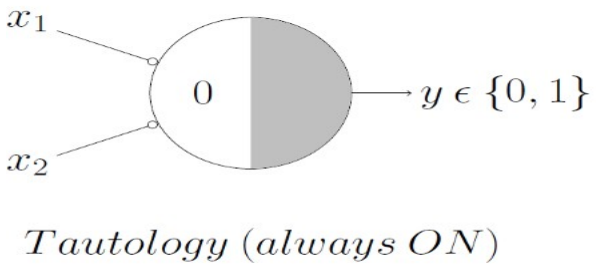


AND function

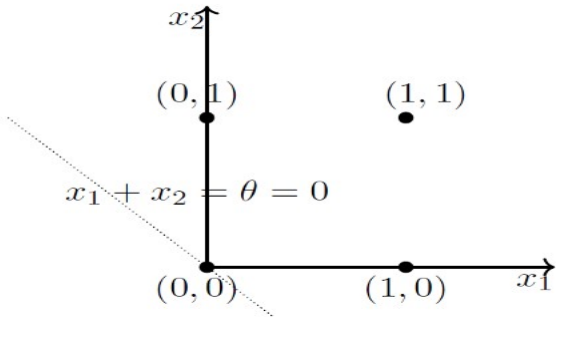
$$x_1 + x_2 = \sum_{i=1}^{2} x_i \geq 2$$



Geometrical interpretation of Tautology MP-neuron

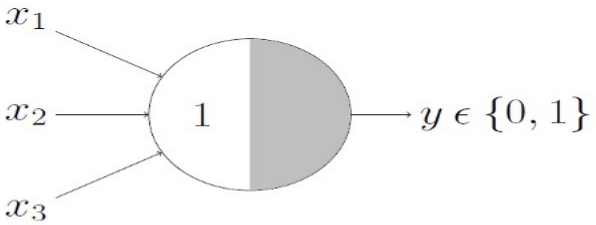


Tautology (always ON)



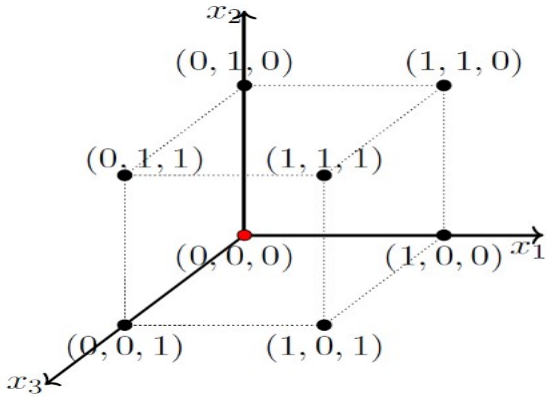
Geometrical interpretation of OR function

MP-neuron with 3 inputs



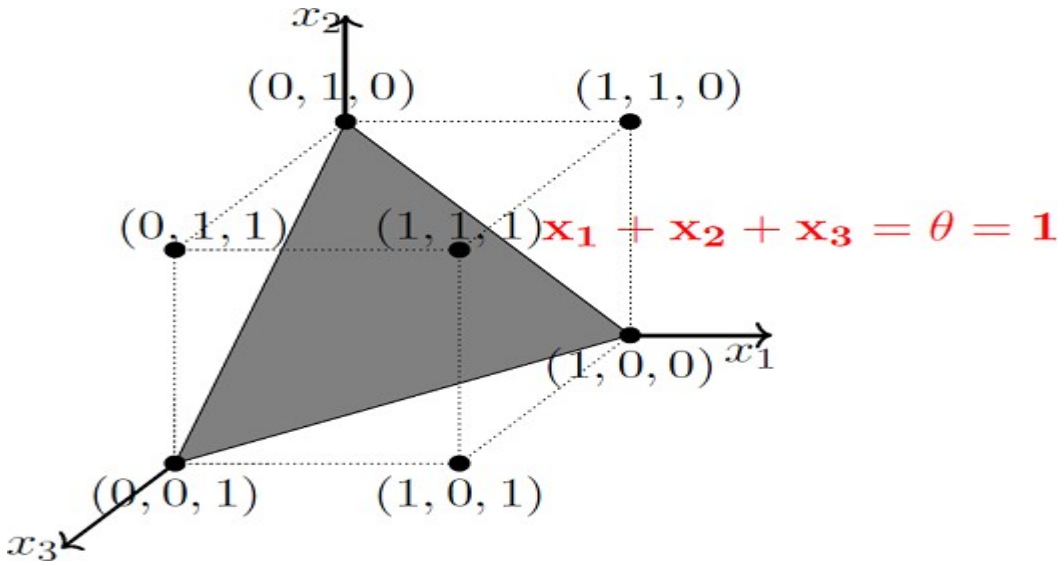
OR function

$$x_1 + x_2 + x_3 = \sum_{i=1}^{3} x_i \geq 1$$



Geometrical interpretation of OR function

MP-neuron with 3 inputs



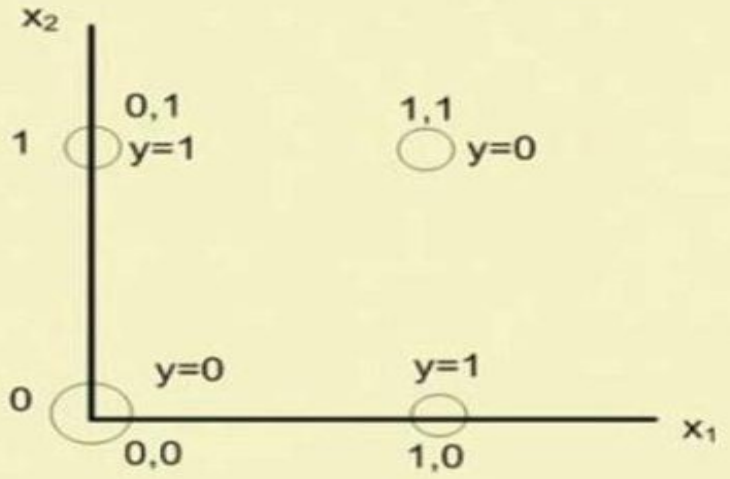
For the OR function, we want a plane such that the point (0,0,0) lies on one side and the remaining 7 points lie on the other side of the plane

XOR Problem

XOR PROBLEM IS LINEARLY NON-SEPARABLE

x_1	x_2	Output (y)
0	0	0
0	1	1
1	0	1
1	1	0

XOR Problem

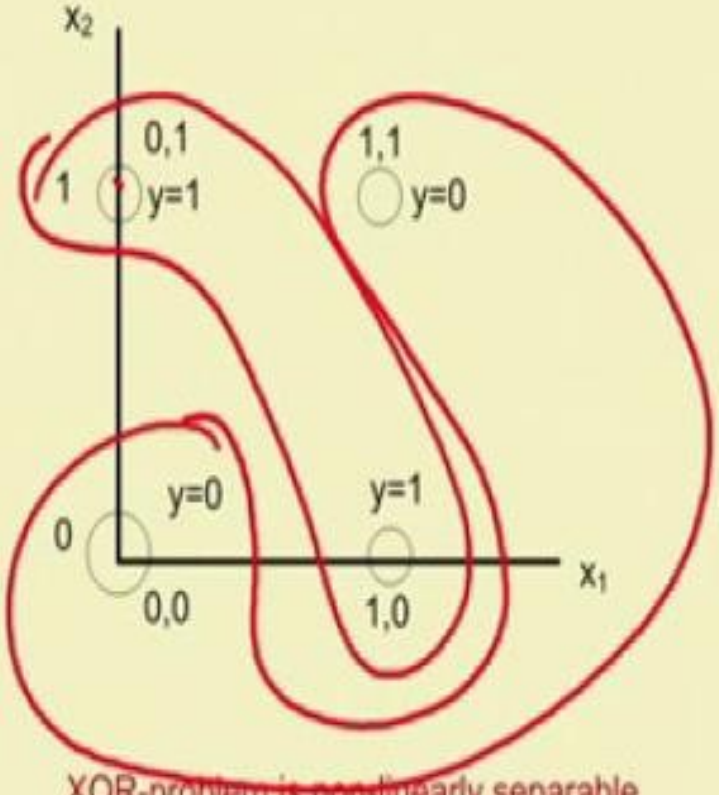


XOR-problem is non-linearly separable

XOR problem is linearly non-separable

x_1	x_2	Output (y)
0	0	0
0	1	1
1	0	1
1	1	0

XOR Problem



XOR-problem is non-linearly separable

Thus..

- A single McCulloch Pitts Neuron can be used to represent boolean functions which are linearly separable
- **Linear separability (for boolean functions)** : There exists a line (plane) such that all inputs which produce a 1 lie on one side of the line (plane) and all inputs which produce a 0 lie on other side of the line (plane)

Limitation of MP-neuron

- What about non-boolean (say, real) inputs?
- Do we always need to hand code the threshold?
- Are all inputs equal? What if we want to assign more importance to some inputs?
- What about functions which are not linearly separable? Say XOR function.

Linear separability

- It is a concept wherein the separation of the input space into regions is based on whether the network response is positive or negative.
- A decision line is drawn to separate positive or negative response.
- The decision line is also called as *decision-making line or decision-support line or linear-separable line.*
- The net input calculation to the output unit is given as

$$y_{in} = b_j + \sum_{i=1}^{n} xiwi$$

- The region which is called as *decision boundary* is determined by the relation

$$b + \sum_{i=1}^{n} xiwi = 0$$

Contd..

- Linear separability of the network is based on decision-boundary line.
- If there exists weights(bias) for which training input vectors have positive response (+1 ve) input lie on one side of line and other having negative response(-ve) lie on other side of line, we conclude the problem is linearly separable

Contd..

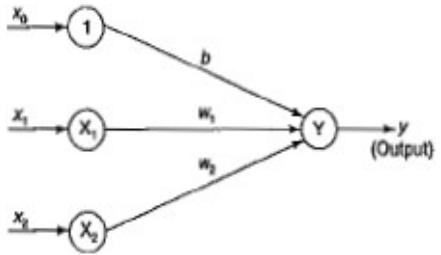


Figure 2-19 A single-layer neutal net.

Net input of the network

$$Y_{in} = b + x_1 w_1 + x_2 w_2$$

Separating line for which the boundary lies between the values of x_1 and x_2

$$b + x_1 w_1 + x_2 w_2 = 0$$

If weight of w_2 is not equal to 0 then we get

$$x2 = -w_1/w_2 - b/w_2$$

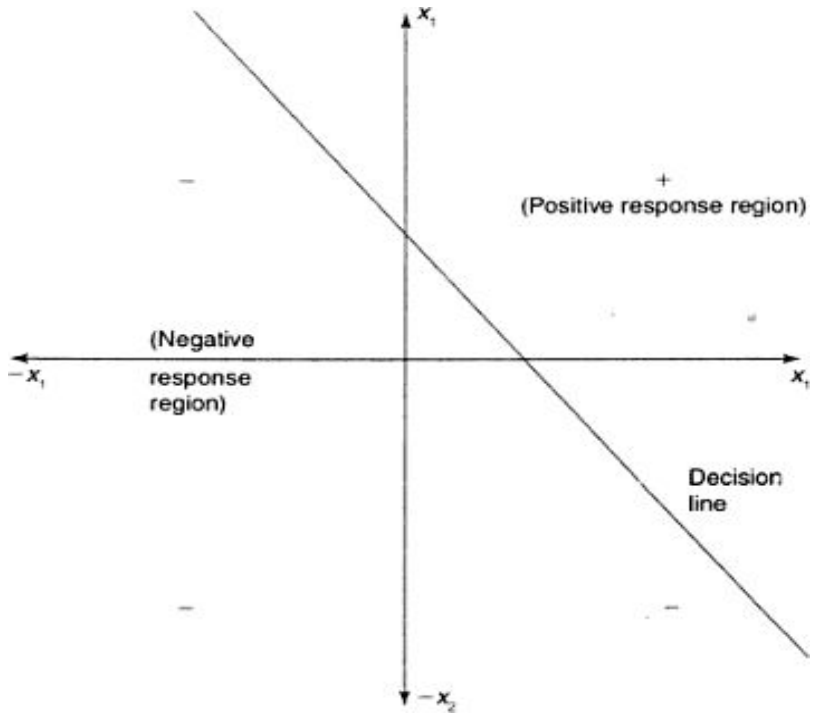
Net requirement for positive response

$$b + x_1 w_1 + x_2 w_2 > 0$$

During training process , from training data w1, w2 and b are decided.

Contd..

- Consider a network having positive response in the first quadrant and negative response in all other quadrants with either binary or bipolar data.
- Decision line is drawn separating two regions as shown in Fig.
- Using bipolar data representation, missing data can be distinguished from mistaken data. Hence bipolar data is better than binary data.
- Missing values are represented by 0 and mistakes by reversing the input values from +1 to -1 or vice versa.



Using the linear separability concept, obtain the response for OR function (rake bipolar inputs and bipolar targets).

Solution:

x_1	x_2	y
1	1	1
1	−1	1
−1	1	1
−1	−1	−1

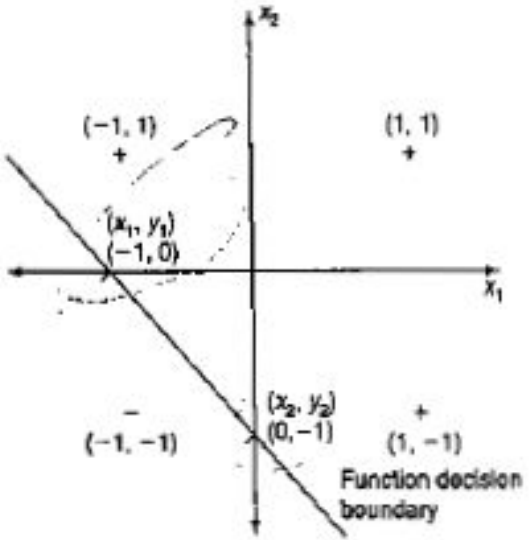


Figure 10 Graph for 'OR' function.

Assuming tbe coordinates as (-1, 0) and (0,-1); as *(x1,y1*) and *(x2,y2),* the slope *"m"* of the straight line can be obtained as

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{-1 - 0}{0 + 1} = \frac{-1}{1} = -1$$

We now calculate *c:*

'c=y1-mx1

= 0- (-1)(-1) = -1

Using this value the equation for the line is given as

y = mx+c

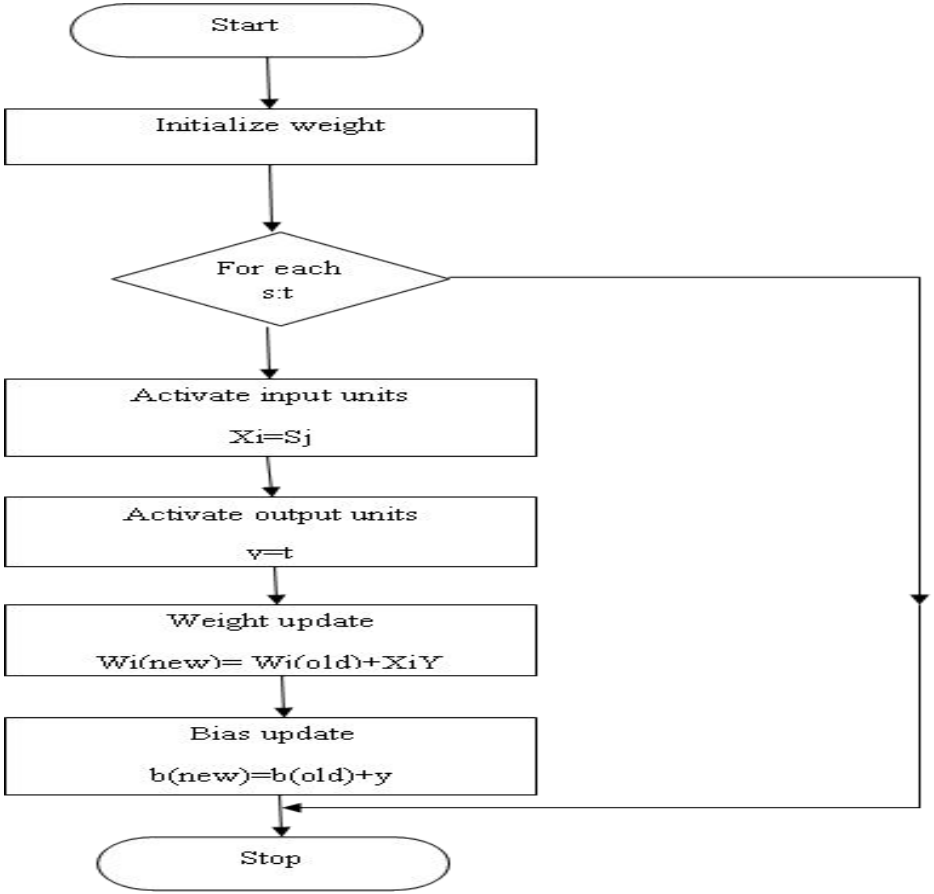
= (-1)x-l

= -x-1

Hebb network

- When an axon of cell A is near enough to excite cell B, and repeatedly or permanently takes place in firing it, some growth process or metabolic change takes place in one or both the cells such than A's efficiency, as one of the cells firing B, is increased.
- According to Hebb rule, *the weight vector is found to increase proportionately to the product of the input and the learning signal.*
- In Hebb learning, two interconnected neurons are 'on' simultaneously.
- The weight update in Hebb rule is given by
  - $$W_i(\text{new}) = w_i\,(\text{old}) + x_i\, y.$$
- It is suited more for bipolar data.

Flowchart


Start
Initialize weight
For each
s:t
Activate input units
Xi=Sj
Activate output units
y=t
Weight update
Wi(new)= Wi(old)+XiY
Bias update
b(new)=b(old)+y
Stop

Steps:

- 0: First initialize the weights.
- 1: Steps 2-4 have to be performed for each input training vector and target output pair, s:t
- 2: Input activations are set. The activation function for input layer is identity function.
  - $X_i = S_i$ for i=1 to n
- 3: Output activations are set.
- 4: Weight adjustment and bias adjustments are performed.
  - $W_i(new) = w_i(old)+x_iy$
  - $b(new)=b(old)+y$
- In step 4, the weight updation formula can be written in vector form as
  - *w(new)=w(old)+y.*
- Change in weight is expressed as
  - $\Delta w = xy$

  Hence,

  w(new)=w(old)+Δw

Hebb rule is used for pattern association, pattern categorization, pattern classification and over a range of other areas

Design a Hebb net to implement OR function

Use bipolar data in the place of binary data Initially the weights and bias are set to zero w1=w2=b=0

Inputs			Target
X1	X2	B	y
1	1	1	1
1	-1	1	1
-1	1	1	1
-1	-1	1	-1

- **First input [x1 x2 b] = [1 1 1] and target = 1 [i.e. y = 1],** setting the intial weights as old weights and applying the hebb rule, we get

wi (new) = wi(old) + xi*y
w1(new) = w1(old) + x1*y = 0+1*1 = 1
w2(new) = w2(old) + x2*y = 0+1*1 = 1
b(new) = b(old) +y = 0+1 = 1
The weight change here is Δwi = xi*y
Δw1 = x1*y = 1
Δw2 = x2*y = 1
Δ = b = y = 1

- **Similarly for second input [x1 x2 b] = [1 -1 1] and y = 1**

w1(new) = w1(old) + x1*y = 1+1*1 = 2 & Δw1 = 1
w2(new) = w2(old) + x2*y = 1+(-1)*1 = 0 & Δw2 = -1
b(new) = b(old) +y = 1+1 = 2 & Δb = 1

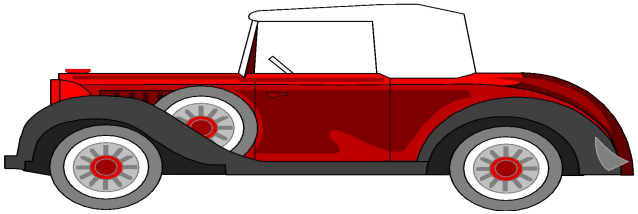
- **Similarly for third and fourth input, output can be calculated.**

Inputs			y	Weight changes			Weights		
X1	X2	b	Y	ΔW1	ΔW2	Δb	W1	W2	b
1	1	1	1	1	1	1	1	1	1
1	-1	1	1	1	-1	1	2	0	2
-1	1	1	1	-1	1	1	1	1	3
-1	-1	1	-1	1	1	-1	2	2	2

Applications of NNs

- **classification**
  - in marketing: consumer spending pattern classification
  - In defence: radar and sonar image classification
  - In agriculture & fishing: fruit and catch grading
  - In medicine: ultrasound and electrocardiogram image classification, EEGs, medical diagnosis
- **recognition and identification**
  - In general computing and telecommunications: speech, vision and handwriting recognition
  - In finance: signature verification and bank note verification
- **assessment**
  - In engineering: product inspection monitoring and control
  - In defence: target tracking
  - In security: motion detection, surveillance image analysis and fingerprint matching
- **forecasting and prediction**
  - In finance: foreign exchange rate and stock market forecasting
  - In agriculture: crop yield forecasting
  - In marketing: sales forecasting
  - In meteorology: weather prediction

Engine Management



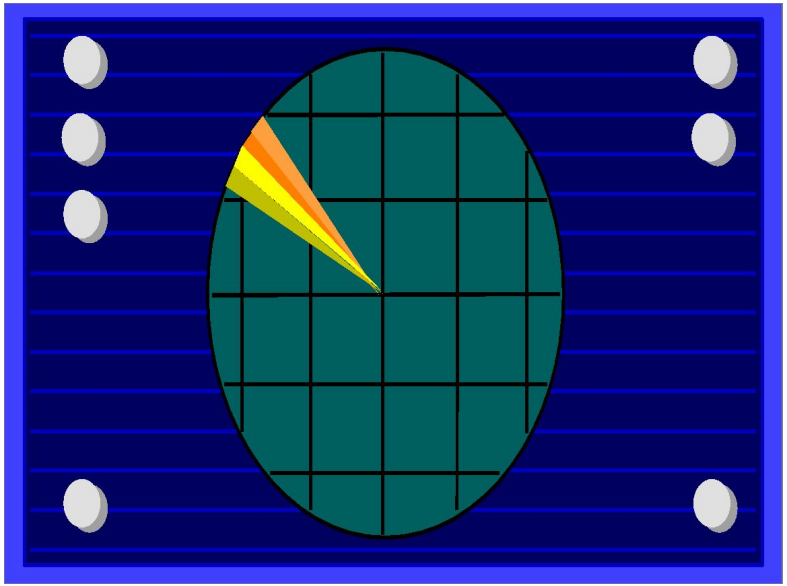
- The behaviour of a car engine is influenced by a large number of parameters
  - temperature at various points
  - fuel/air mixture
  - lubricant viscosity.
- Major companies have used neural networks to dynamically tune an engine depending on current settings.

Signature Recognition

- Each person's signature is different.
- There are structural similarities which are difficult to quantify.
- One company has manufactured a machine which recognizes signatures to within a high level of accuracy.
  - Considers speed in addition to gross shape.
  - Makes forgery even more difficult.

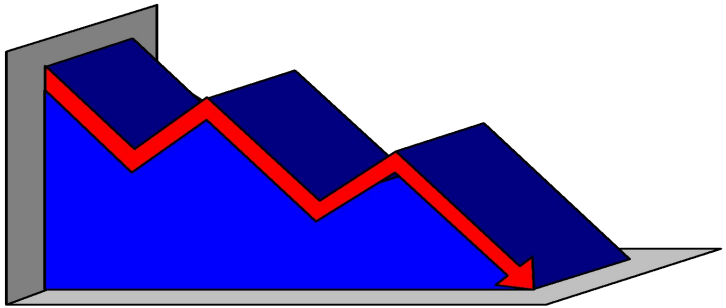
Sonar Target Recognition

- Distinguish mines from rocks on sea-bed
- The neural network is provided with a large number of parameters which are extracted from the sonar signal.
- The training set consists of sets of signals from rocks and mines.



Stock Market Prediction

- "Technical trading" refers to trading based solely on known statistical parameters; e.g. previous price
- Neural networks have been used to attempt to predict changes in prices.
- Difficult to assess success since companies using these techniques are reluctant to disclose information.



Classification of Neural Networks

		Learning Methods			
		Gradient descent	Hebbian	Competitive	Stochastic
Types of Architecture	Single-layer feed-forward	ADALINE, Hopfield, Percepton,	AM, Hopfield,	LVQ, SOFM	-
	Multi-layer feed- forward	CCM, MLFF, RBF	Neocognition		
	Recurrent Networks	RNN	BAM, BSB, Hopfield,	ART	Boltzmann and Cauchy machines

Table : Classification of Neural Network Systems with respect to learning methods and Architecture types