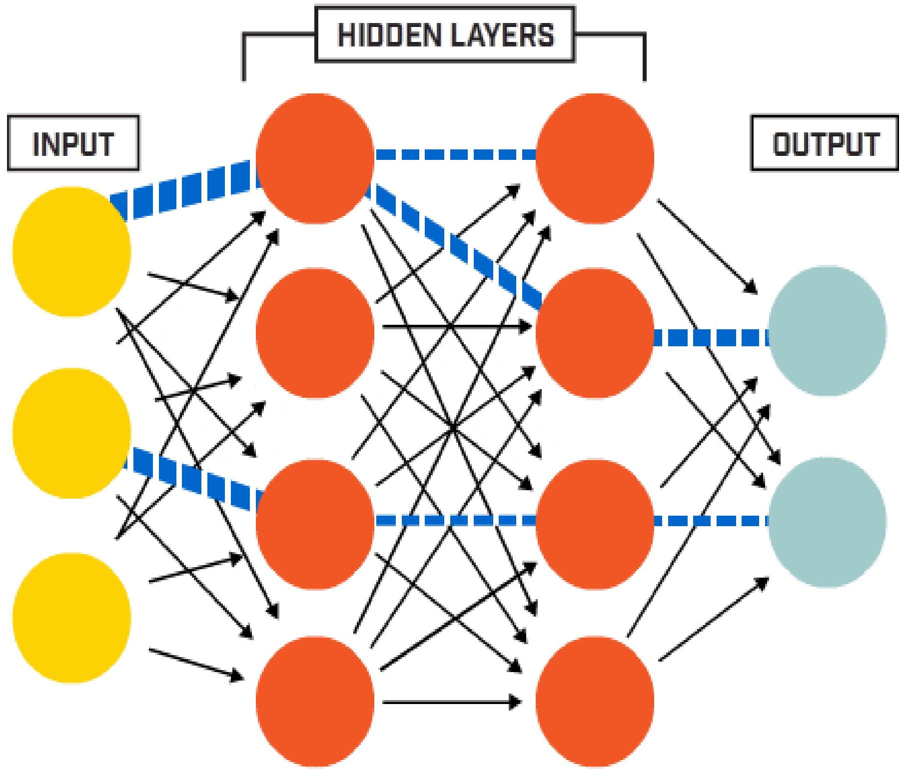
ARTIFICIAL NEURAL NETWORK AND DEEP LEARNING



Dr. Somesh Nandi
Dept of AIML
RVCE

Unit -1

- **Neural Networks**
  - What is NN
  - Models
  - Neural networks as directed graphs
  - Architectures
- **Learning Processes**
  - Different Types of Learning Process
  - Learning with and without teacher
  - Learning tasks
  - Memory and adaptation
  - Statistical Learning

Unit -1

- **Neural Networks**

  - **A Neural network is a massively parallel distributed processor made up of simple processing units which has a natural propensity for storing experiential knowledge and making it available for use. It resembles human brain in two aspects**
    - **Knowledge is acquired by the network from its environment through a learning process**
    - **Interneuron connection strengths known as synaptic weights are used to store the acquired knowledge**

  - **Procedure to perform the learning process is called learning algorithm**

Unit -1

- **Benefits of Neural Networks**

  Neural Networks achieve power through

  - **Massively Parallel and Distributed Structure**
  - **Generalization – refers to produce reasonable outputs for inputs not encountered in training**
- **Properties of Neural Networks**
  - **Non Linearity** -- Neurons are distributed throughout the network
  - **Input – Output Mapping** -- Reordering of previously trained examples and mapping Input – output using estimation statistics
  - **Adaptivity** -- Adapt to the Surrounding Environment
  - **Evidential Response** -- Decision and Confidence , rejection of unwanted patterns
  - **Contextual Information** – Every neuron is affected by the activity of global neurons
  - **Fault Tolerance** -- hardware is fault tolerant, robust in nature , performance degrades in adverse conditions, but not catastrophic failure
  - **VLSI Implement ability** -- well suited due to massive parallelism
  - Uniformity of Analysis and Design -- commonality, share and integrity
  - Neurobiological Analogy

Unit -1

Models of a Neuron

Synapses are the connecting junction between axon and dendrites. The majority of synapses send signals from the axon of a neuron to the dendrite of another neuron

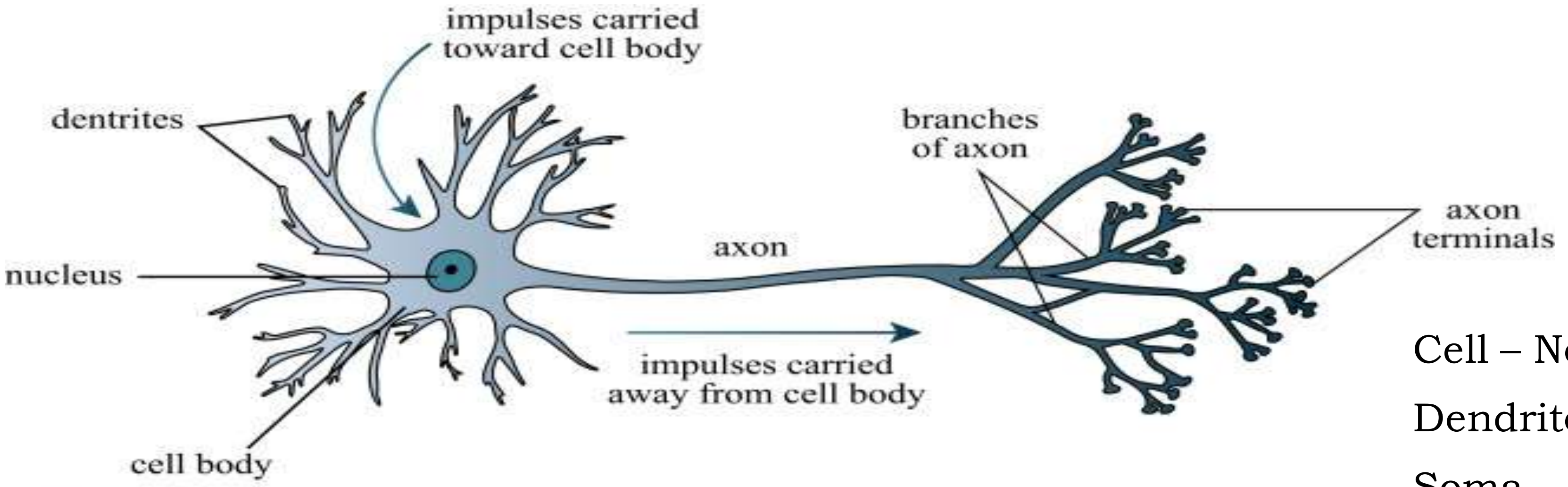
Dendrites have fibers branching out from the soma in a bushy network around the nerve cell.

- Dendrites allow the cell to receive signals from connected neighboring neurons and each dendrite is able to perform multiplication by that dendrite’s weight value

Axons are the single, long fibers extending from the main soma.

- The axon will branch and connect to other dendrites

Unit -1

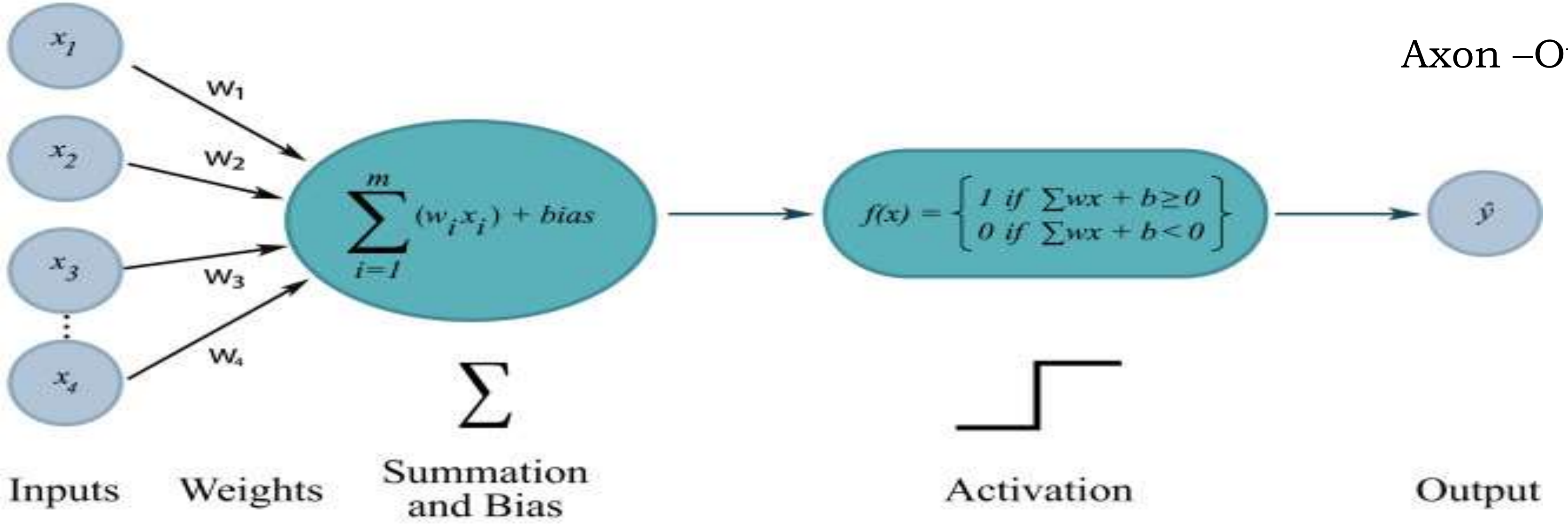


Cell – Neuron

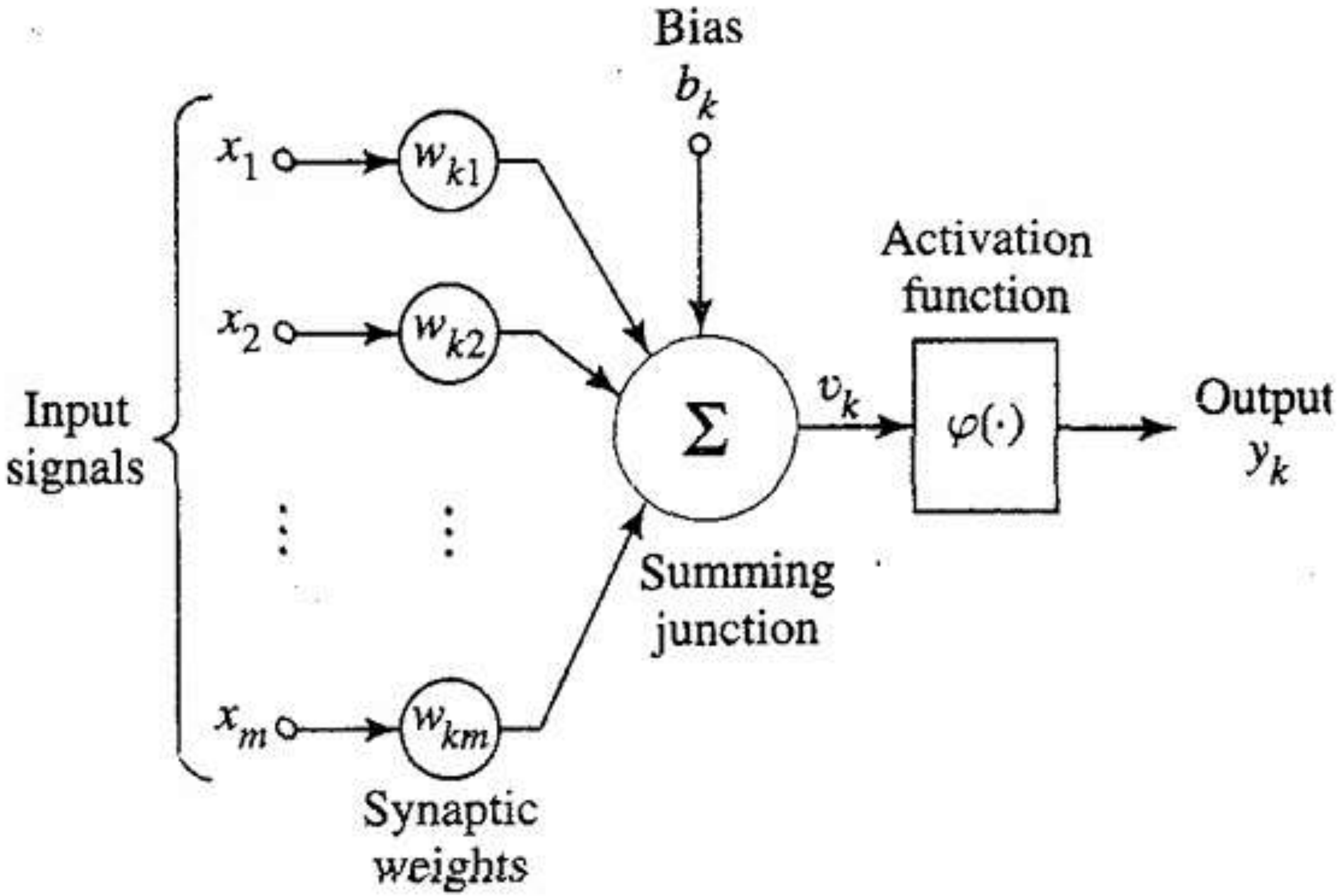
Dendrites –Weight or interconnections

Soma – Net Input

Axon –Output

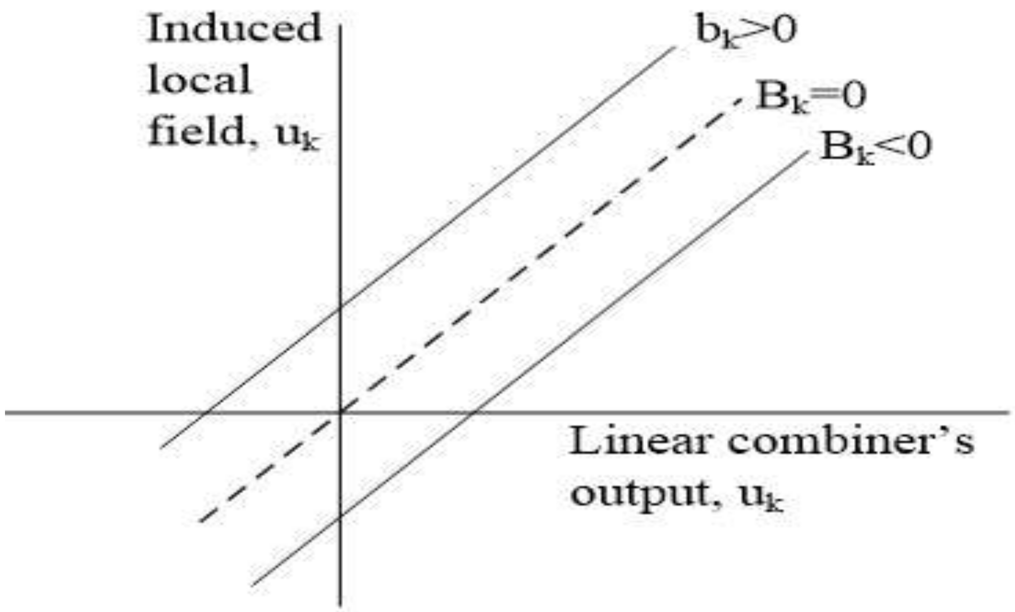


Unit -1

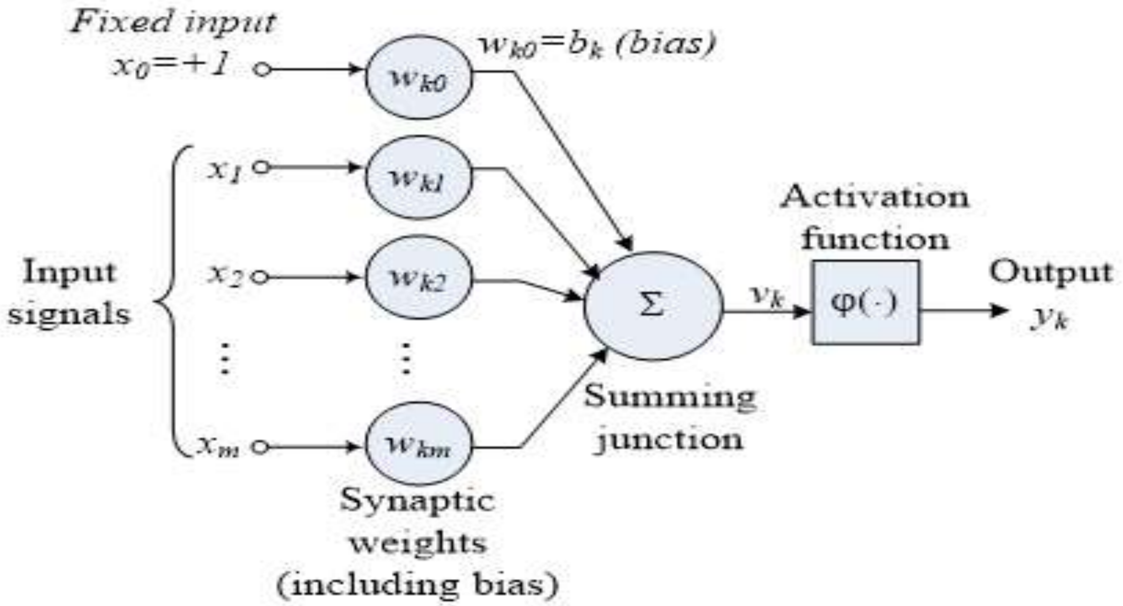

Bias
b_k
x_1
w_{k1}
x_2
w_{k2}
Input signals
x_m
w_{km}
Synaptic weights
Σ
Summing junction
v_k
Activation function
$\varphi(\cdot)$
Output
y_k

Unit -1

Non Linear Model of a Neuron



Affine transformation produced by the presence of a bias



Another Nonlinear model of a neuron

Functional Parameters of a Neuron

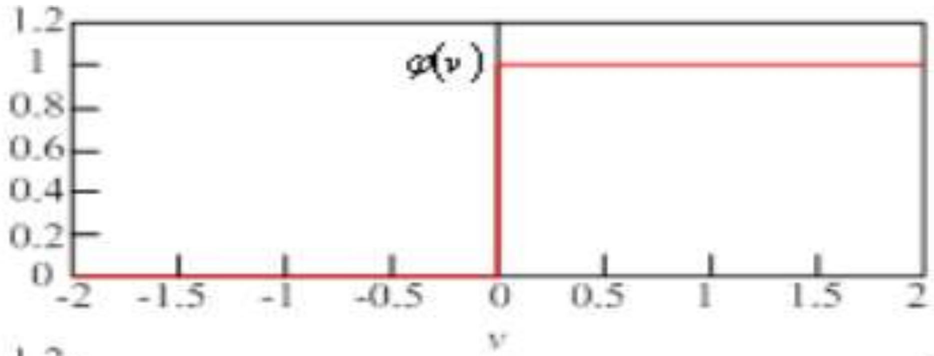
Function parameter	Description
w	Vector of real-valued weights on the connections
w · x	Dot product ($\sum_{i=1}^{n} w_i x_i$)
n	Number of inputs to the perceptron
b	The bias term (input value does not affect its value; shifts decision boundary away from origin)

Unit -1

Activation Function denoted by $\varphi(v)$ defines the output of a neuron in terms of induced local field v.

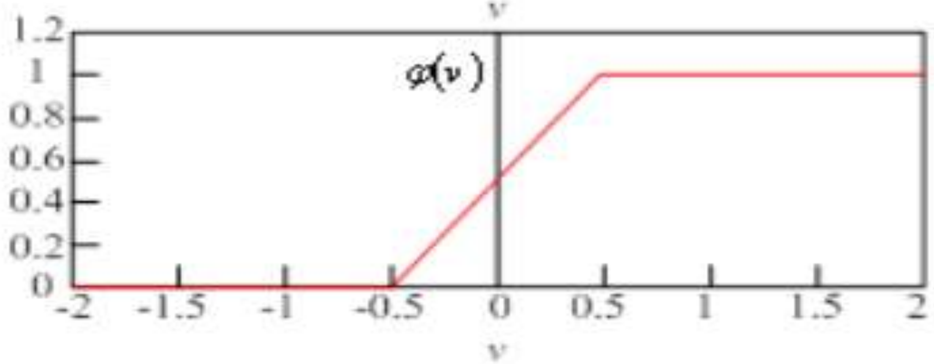
Threshold Function

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases}$$



Piecewise-Linear Function

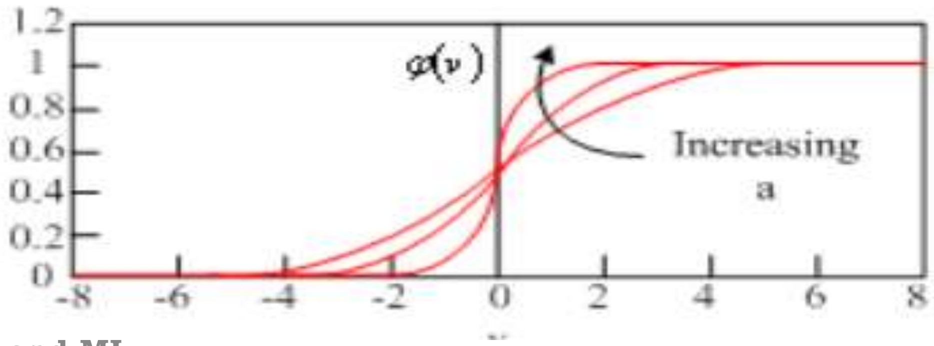
$$\varphi(v) = \begin{cases} 1 & v \geq 1/2 \\ v & -1/2 < v < 1/2 \\ 0 & v \leq -1/2 \end{cases}$$



Sigmoid Function

$$\varphi(v) = \frac{1}{1+\exp(-av)}$$

a is the slope parameter



Unit -1

- **Signum Function :** Activation function ranging from -1 to +1 where the activation function assumes an antisymmetric form with respect to origin, the threshold can be defined as

$$\varphi(v) = \begin{cases} 1 & \text{if } v > 0 \\ 0 & \text{if } v = 0 \\ -1 & \text{if } v < 0 \end{cases}$$

The hyperbolic tangent function of the same can be defined as $\varphi(v) = \tanh(v)$

Unit -1

Neural Networks as Directed Graphs – Signal Flow Diagrams

- Block diagram providing functional description of the network
- Signal Flow graph providing a complete description of signal flow in the network
- Architectural graph describing the network layout

Description of the graph

1. *Source nodes* supply input signals to the graph.
2. Each neuron is represented by a single node called a *computation node*.
3. The *communication links* interconnecting the source and computation nodes of the graph carry no weight; they merely provide directions of signal flow in the graph.

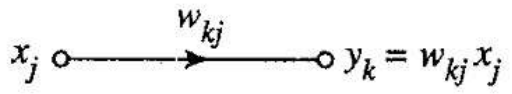
Unit -1

Neural Networks as Directed Graphs – Signal Flow Diagrams

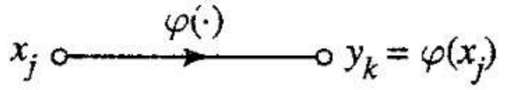
Rule 1. A signal flows along a link only in the direction defined by the arrow on the link.

Two different types of links may be distinguished:

- *Synaptic links*, whose behavior is governed by a *linear* input–output relation. Specifically, the node signal x_j is multiplied by the synaptic weight w_{kj} to produce the node signal y_k, as illustrated in Fig. 1.9a.
- *Activation links*, whose behavior is governed in general by a *nonlinear* input–output relation. This form of relationship is illustrated in Fig 1.9b, where φ(·) is the nonlinear activation function.



(a)



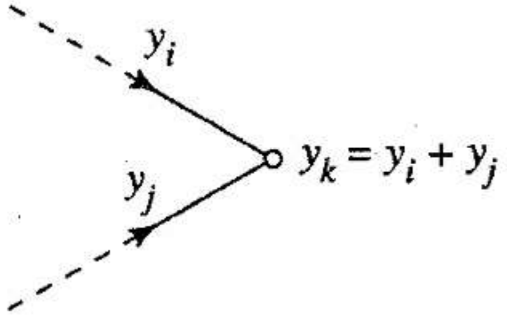
(b)

Unit -1

Neural Networks as Directed Graphs

Rule 2. A node signal equals the algebraic sum of all signals entering the pertinent node via the incoming links.

This second rule is illustrated in Fig. 1.9c for the case of *synaptic convergence* or *fan-in*.



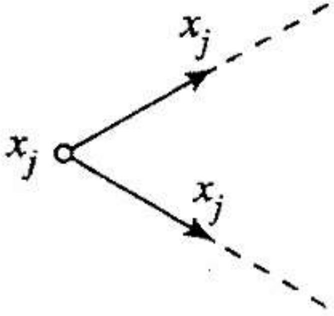
(c)

Unit -1

Neural Networks as Directed Graphs

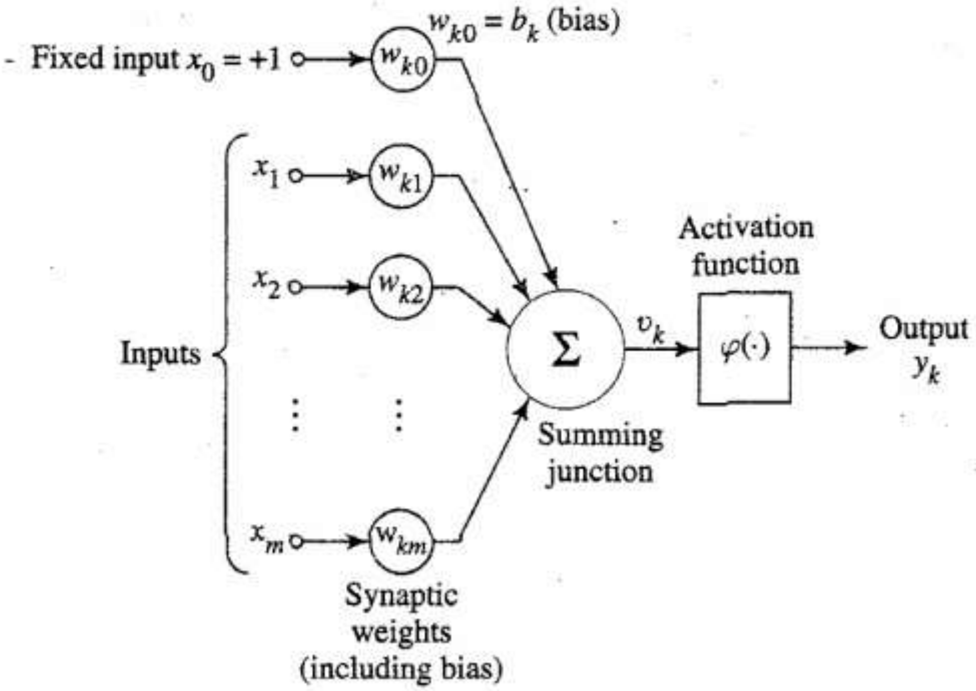
Rule 3. The signal at a node is transmitted to each outgoing link originating from that node, with the transmission being entirely independent of the transfer functions of the outgoing links.

This graph illustrates the synaptic divergence or fanout

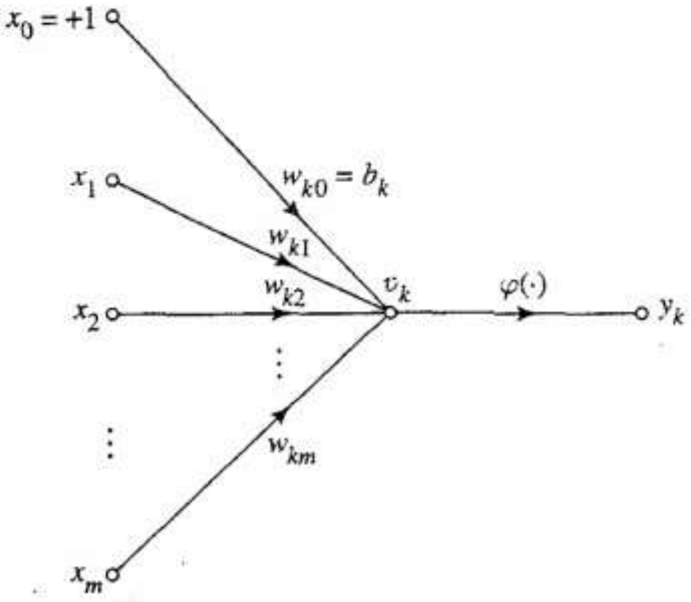


(d)

Neural Networks as Directed Graphs



Neural Network Diagram

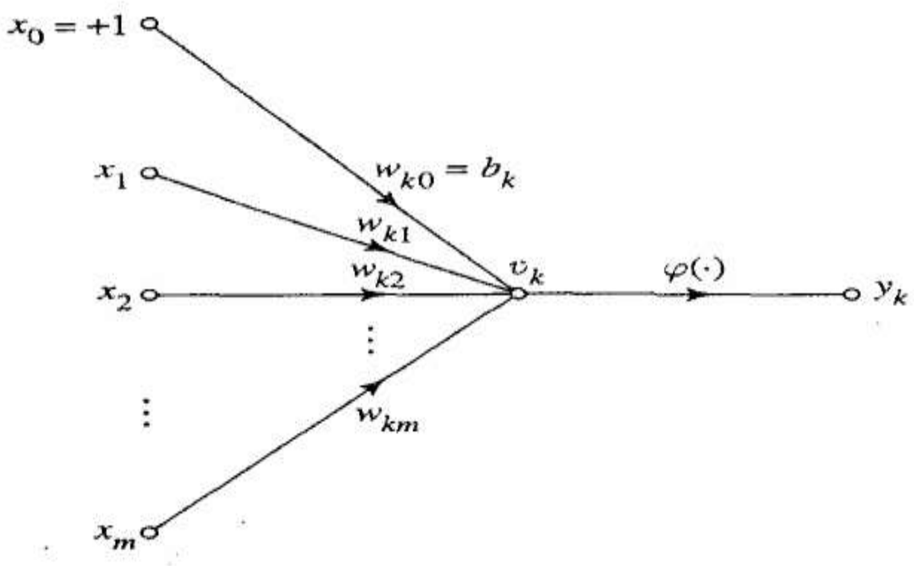


Signal Flow Diagram

Unit -1

Neural Networks as Directed Graphs

Mathematical definition of a NN



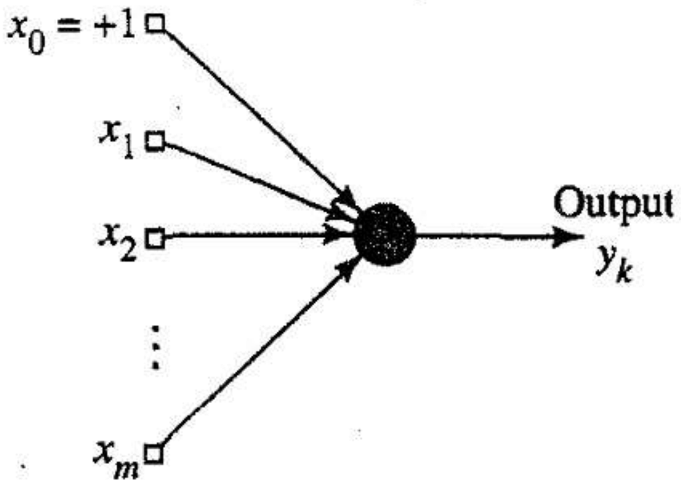
A neural network is a directed graph consisting of nodes with interconnecting synaptic and activation links, and is characterized by four properties:

1. *Each neuron is represented by a set of linear synaptic links, an externally applied bias, and a possibly nonlinear activation link. The bias is represented by a synaptic link connected to an input fixed at +1.*
2. *The synaptic links of a neuron weight their respective input signals.*
3. *The weighted sum of the input signals defines the induced local field of the neuron in question.*
4. *The activation link squashes the induced local field of the neuron to produce an output.*

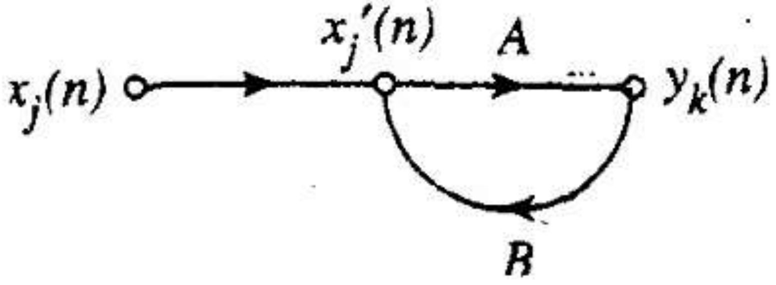
Unit -1

Feedback

Feedbacks are special class of neural networks called as Recurrent Neural Networks (RNNs)



Architectural Graph

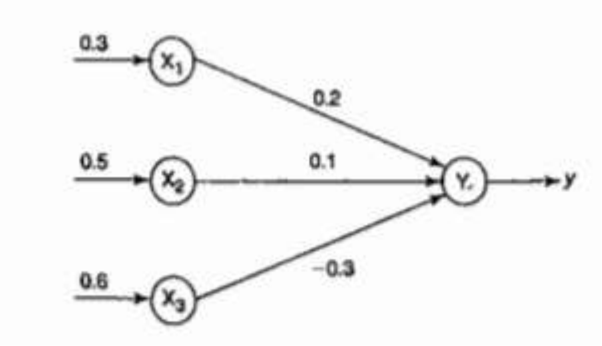


Signal Flow Graph

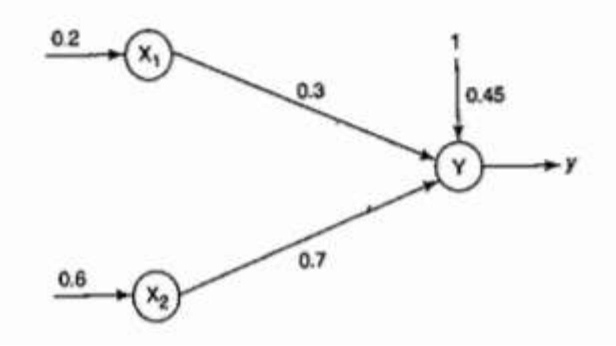
Unit -1

Problems

- For the given network, calculate the net input to the output neuron



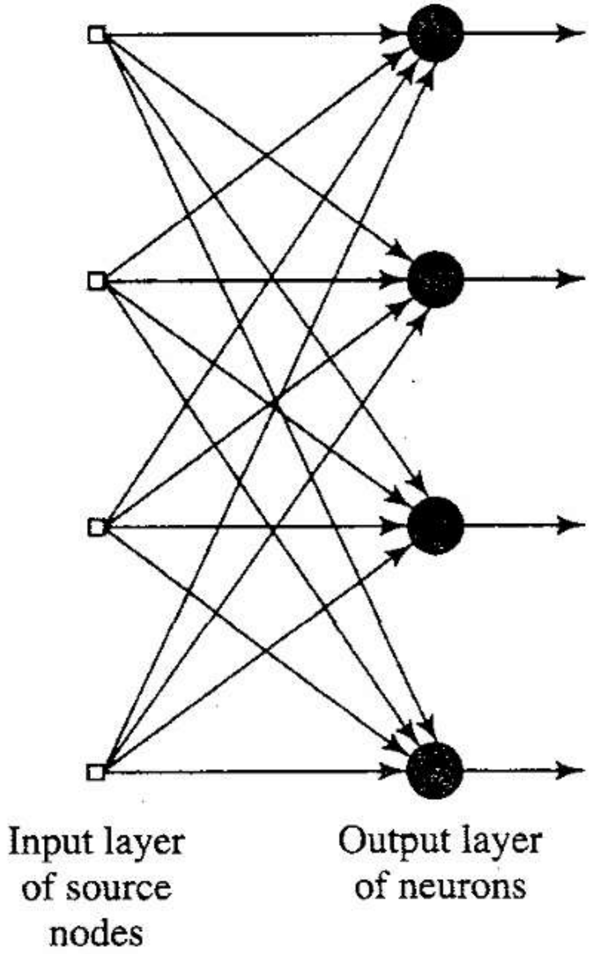
- Calculate the net input (i) Without bias and (ii) when the bias is 0.45 and observe the difference and discuss



Unit -1

Network Architeures

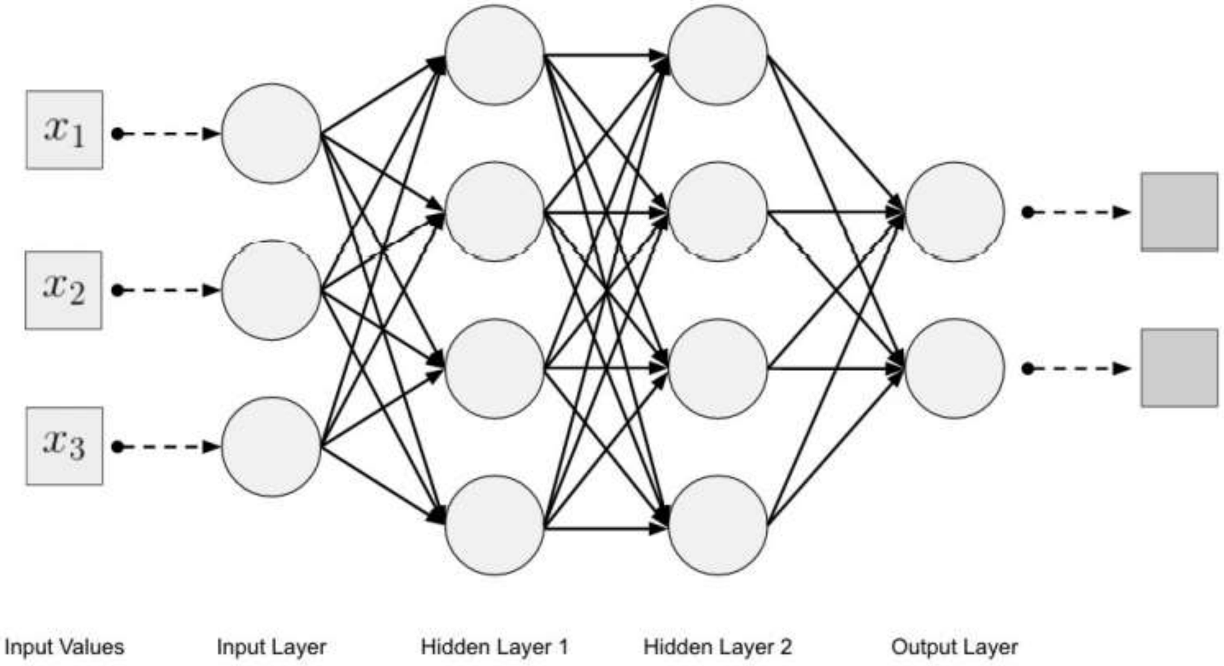
- Single Layer Feedforward network
  - Consists of input layer of source nodes that projects to output layer of computation node
  - Strictly feedforward or acylic type
  - Single output layer of computation nodes
- Layer is formed by taking a processing element and combining it with other processing elements
- Interconnections lead to the formation of various network architectures



Unit -1

Network Architeures

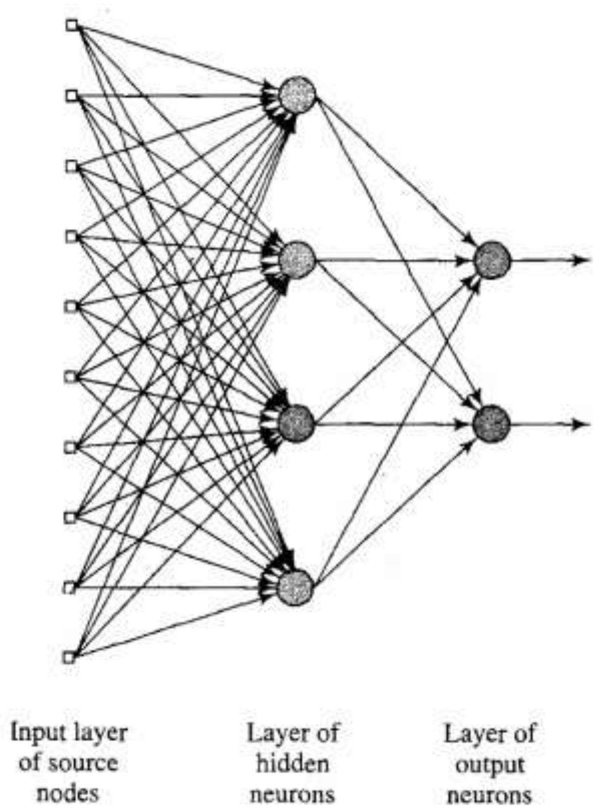
- Multi Layer Feedforward network



Unit -1

Network Architeures

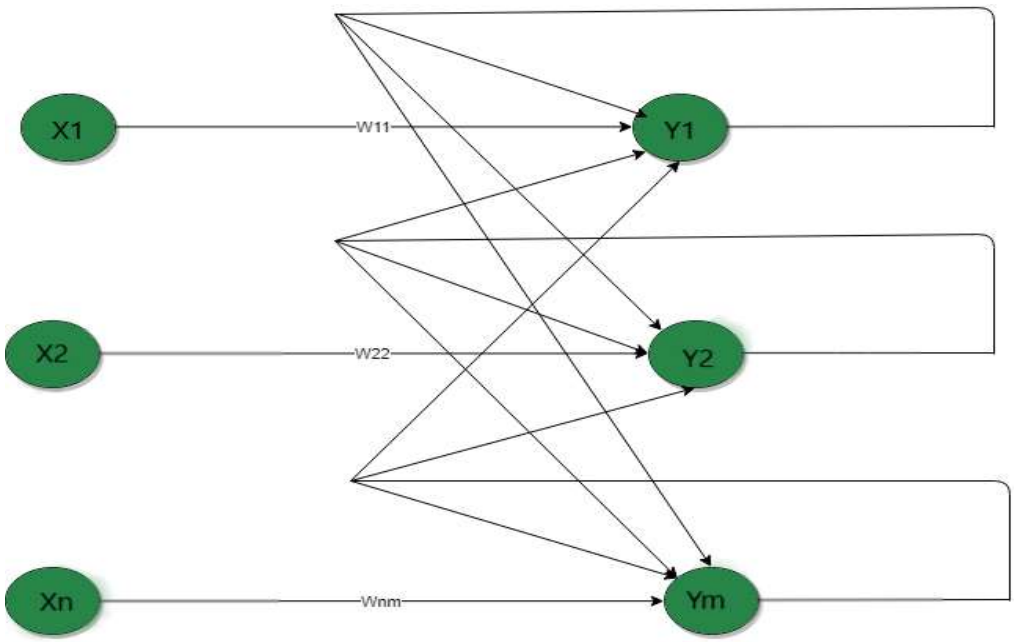
- Multi Layer Feedforward network
    - Consists of input layer , hidden layer and output layer
- Each layer can have a different number of neurons and each layer is fully connected to the adjacent layer.
- The connections between the neurons in the layers form an acyclic graph



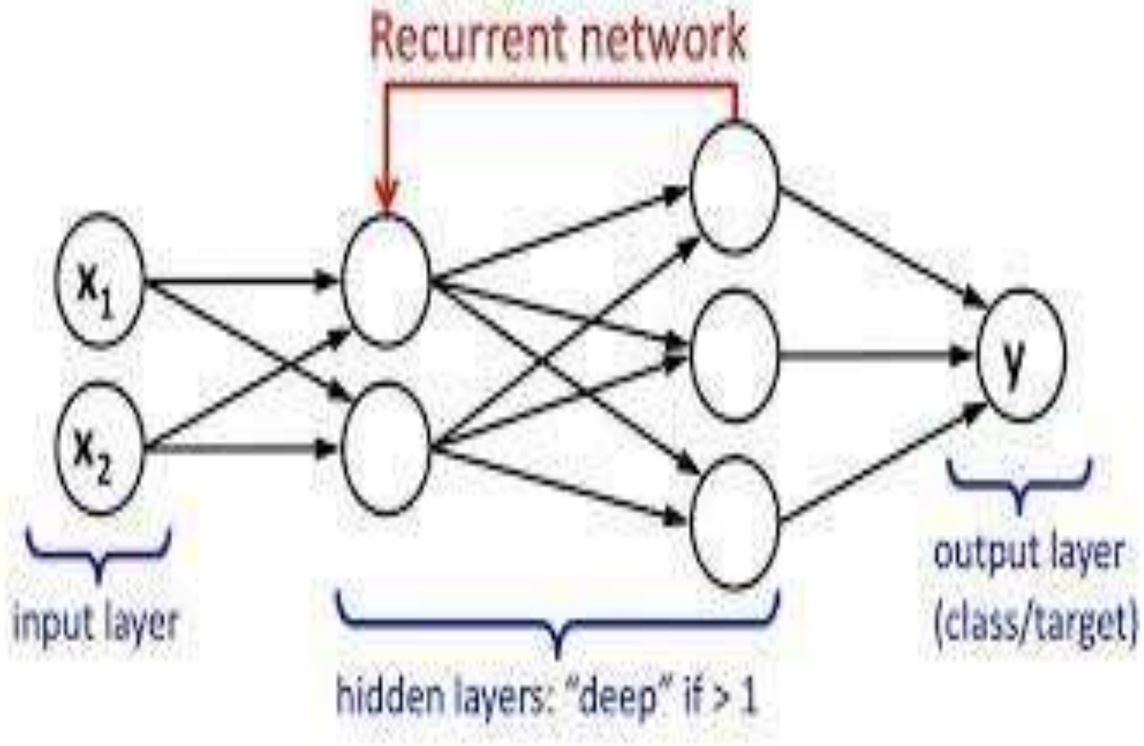
Unit -1

Recurrent Neural Network(RNN) - the output from the previous step is fed as input to the current step.

Single Layer Recurrent Network

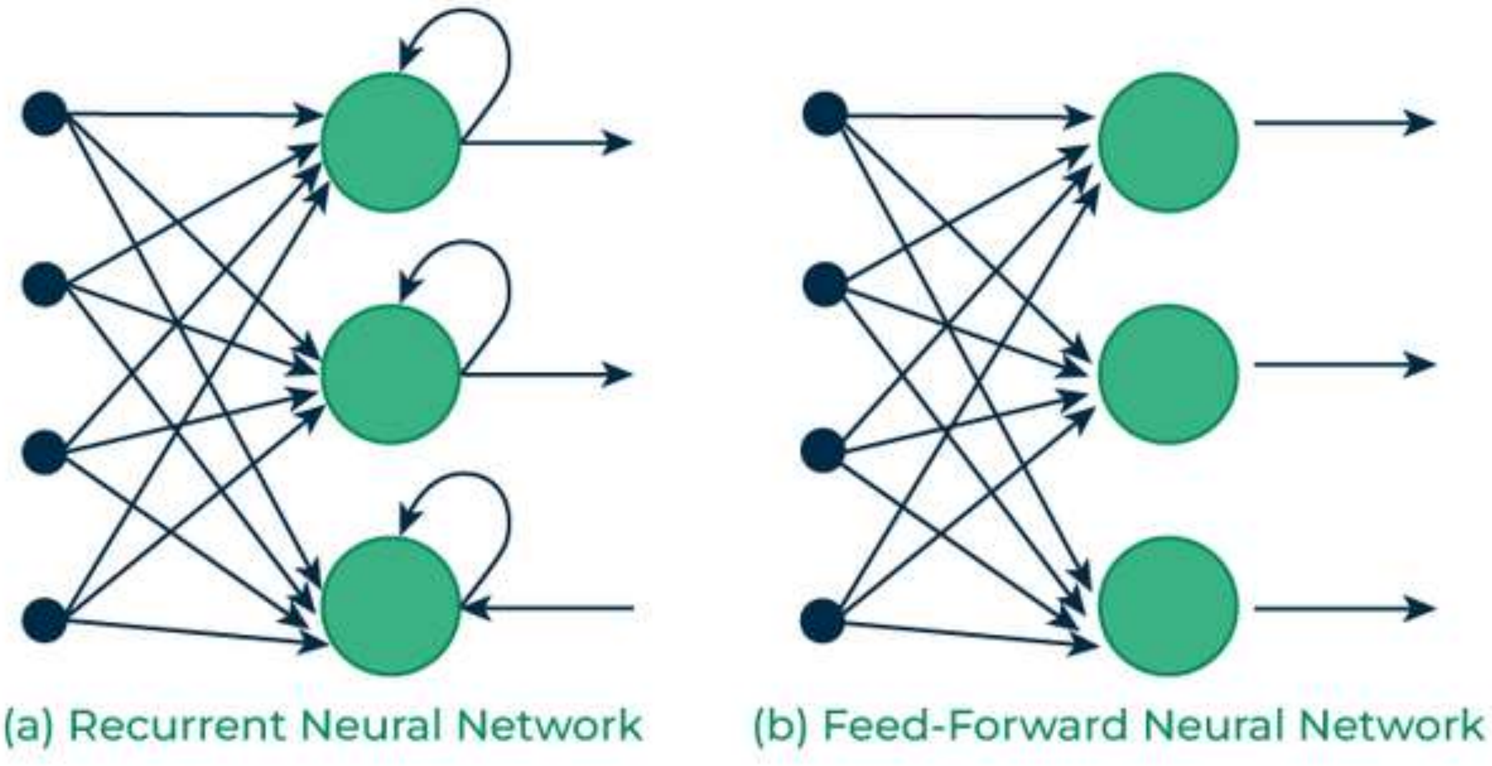


Multi Layer Recurrent Network



Unit -1

Recurrent Neural Networks Vs Feed forward Networks



Unit -1

Learning Process

Error Correction Learning

Memory Based Learning

Hebbian Learning

Competitive Learning

Boltzmann Learning

Learning with Teacher

Learning without Teacher

Unit -1

Error Correction Learning

Consider a simple case of a neuron k constituting the only computational node in the output layer of a feedforward neural network

Neuron k is driven by a signal vector x(n) produced by one or more hidden layers

Hidden layers are driven by an input vector applied from source node

The argument n denotes the discrete time , the time step of an iterative process involved in adjusting the synaptic weights of neuron k

The output signal of the neuron k is denoted by $y_k(n)$

The output signal is comparted to desired response or target output *dk(n).*

An error correction an error signal, denoted by *ek(n),* is produced

$$e_k(n) = d_k(n) - y_k(n)$$

Unit -1

Types of Learning

Error Correction Learning - Example

The error signal ek(n) actuates a control mechanism, apply a sequence of corrective adjustments to the synaptic weights of neuron k.

The corrective adjustments are designed to make the output signal *Yk(n)* come closer to the desired

response *dk(n)* in a step-by-step manner. This objective is achieved by minimizing a *cost function* or *index of performance dk(n)*,defined in terms of the error signal *ek(n*

Unit -1

Types of Learning

Error Correction Learning - Example

The error signal ek(n) actuates a control mechanism, apply a sequence of corrective adjustments to the synaptic weights of neuron k.

The corrective adjustments are designed to make the output signal *Yk(n)* come closer to the desired

response *dk(n)* in a step-by-step manner. This objective is achieved by minimizing a *cost function* or *index of performance dk(n)*,defined in terms of the error signal *ek(n*

Unit -1

Types of Learning

Hebbian Learning

Properties of Hebbian Synapse

1. Time Dependent Mechanism
2. Local Mechanism
3. Interactive Mechanism
4. Conjunctional or correlational Mechanism

Unit -1

Types of Learning

Mathematical Model of Hebbian Network

To formulate Hebbian learning in mathematical terms, consider a synaptic weight w_{kj} of neuron k with presynaptic and postsynaptic signals denoted by x_j and y_k, respectively. The adjustment applied to the synaptic weight w_{kj} at time step n is expressed in the general form

$$\Delta w_{kj}(n) = F(y_k(n), x_j(n))$$

where $F(\cdot,\cdot)$ is a function of both postsynaptic and presynaptic signals. The signals $x_j(n)$ and $y_k(n)$ are often treated as dimensionless.

Form 1:

Hebb's hypothesis. The simplest form of Hebbian learning is described by

$$\Delta w_{kj}(n) = \eta y_k(n) x_j(n) \tag{2.9}$$

where η is a positive constant that determines the *rate of learning*.

Unit -1

Types of Learning

Dis advantage of Hebb's Hypothesis

- The repeated application of the input signal (presynaptic activity) xj leads to an increase in yk and therefore exponential grown that drives the synaptic connection into saturation
- No information will be stored in synapse and the selectivity is lost

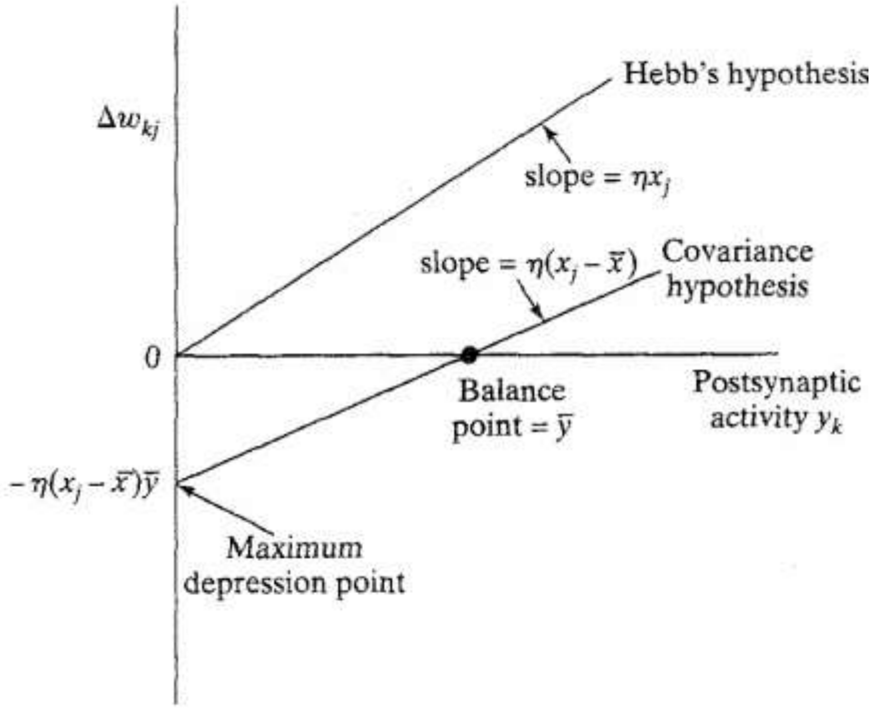
Form 2 :

Covariance hypothesis. One way of overcoming the limitation of Hebb's hypothesis is to use the *covariance hypothesis* introduced in Sejnowski (1977a, b). In this hypothesis, the presynaptic and postsynaptic signals in Eq. (2.9) are replaced by the departure of presynaptic and postsynaptic signals from their respective average values over a certain time interval. Let $\bar{x}$ and $\bar{y}$ denote the *time-averaged values* of the presynaptic signal x_j and postsynaptic signal y_k, respectively. According to the covariance hypothesis, the adjustment applied to the synaptic weight w_{kj} is defined by

$$\Delta w_{kj} = \eta(x_j - \bar{x})(y_k - \bar{y}) \tag{2.10}$$

Types of Learning

Form 2 :



covariance hypothesis allows for the following:

- Convergence to a nontrivial state, which is reached when $x_k = \bar{x}$ or $y_j = \bar{y}$.
- Prediction of both synaptic *potentiation* (i.e., increase in synaptic strength) and synaptic *depression* (i.e., decrease in synaptic strength).

Unit -1

Types of Learning

Form 2 :

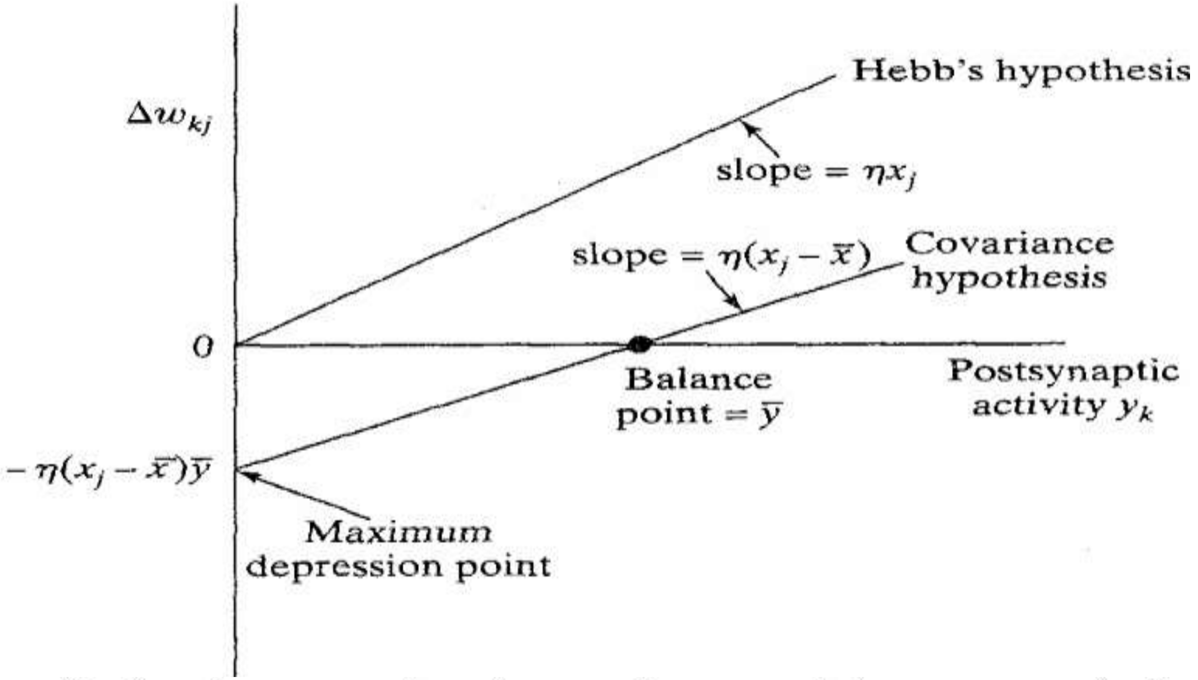
covariance hypothesis allows for the following:

- Convergence to a nontrivial state, which is reached when $x_k = \bar{x}$ or $y_j = \bar{y}$.
- Prediction of both synaptic *potentiation* (i.e., increase in synaptic strength) and synaptic *depression* (i.e., decrease in synaptic strength).

Unit -1

Types of Learning

Difference between Hebb's and Covariance hypothesis:



In both cases the dependence of Δw_{kj} on y_k is linear; however, the intercept with the y_k-axis in Hebb's hypothesis is at the origin, whereas in the covariance hypothesis it is at $y_k = \bar{y}$.

Unit -1

Types of Learning

Observations from eqn 2.10

1. Synaptic weight w_{kj} is enhanced if there are sufficient levels of presynaptic and postsynaptic activities, that is, the conditions $x_j > \bar{x}$ and $y_k > \bar{y}$ are both satisfied.
2. Synaptic weight w_{kj} is depressed if there is either
   - a presynaptic activation (i.e., $x_j > \bar{x}$) in the absence of sufficient postsynaptic activation (i.e., $y_k < \bar{y}$), or
   - a postsynaptic activation (i.e., $y_k > \bar{y}$) in the absence of sufficient presynaptic activation (i.e., $x_j < \bar{x}$).

Unit -1

Types of Learning

Competitive Learning

Basic Elements of Competitive Learning

- A set of neurons that are all the same except for some randomly distributed synaptic weights, and which therefore *respond differently* to a given set of input patterns.
- A *limit* imposed on the "strength" of each neuron.
- A mechanism that permits the neurons to *compete* for the right to respond to a given subset of inputs, such that only *one* output neuron, or only one neuron per group, is active (i.e., "on") at a time. The neuron that wins the competition is called a *winner-takes-all neuron.*

Accordingly the individual neurons of the network learn to specialize on ensembles of similar patterns; in so doing they become *feature detectors* for different classes of input patterns.

Unit -1

Types of Learning

Competitive Learning

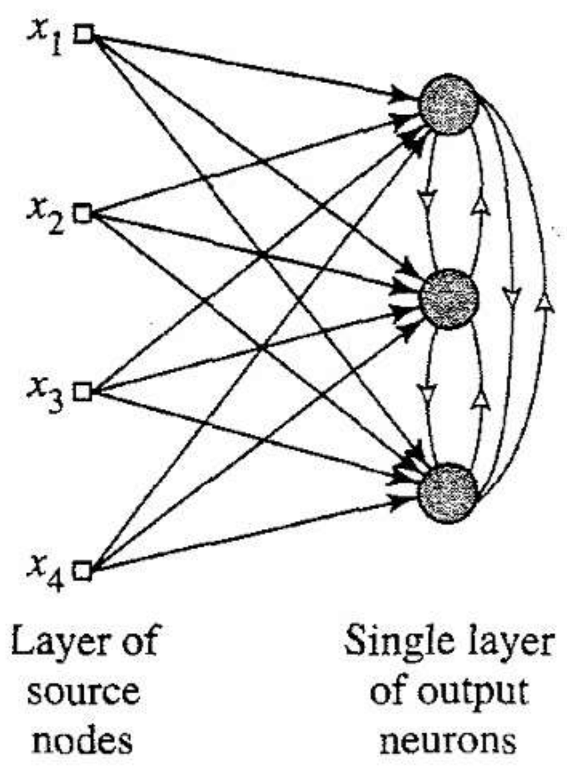


FIGURE 2.4 Architectural graph of a simple competitive learning network with feedforward (excitatory) connections from the source nodes to the neurons, and lateral (inhibitory) connections among the neurons; the lateral connections are signified by open arrows.

Unit -1

Types of Learning

Competitive Learning

- Single Layer of output neurons each is connected to input nodes
- May include feedback connections among neurons
- Preform lateral inhibition with each neuron tending to inhibit the neuron that is laterally connected

$$y_k = \begin{cases} 1 & \text{if } v_k > v_j \text{ for all } j, j \neq k \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_j w_{kj} = 1 \qquad \text{for all } k$$

According to the standard *competitive learning rule*, the change Δw_{kj} applied to synaptic weight w_{kj} is defined by

$$\Delta w_{kj} = \begin{cases} \eta(x_j - w_{kj}) & \text{if neuron } k \text{ wins the competition} \\ 0 & \text{if neuron } k \text{ loses the competition} \end{cases} \qquad (2.13)$$

where η is the learning-rate parameter. This rule has the overall effect of moving the synaptic weight vector $\mathbf{w}_k$ of winning neuron k toward the input pattern $\mathbf{x}$.

Unit -1

Types of Learning

Boltzmann Learning

- Neural network built in the name of Boltzman is called as Boltzmann Learning
- The neurons constitute a recurrent structure and operate in binary manner
- They are either in “on” state or “off” state.
- The energy function is given by

$$E = -\frac{1}{2}\sum_{\substack{j \\ j \neq k}} \sum_{k} w_{kj} x_k x_j$$

where xj is the state of the neuron j and wkj is the synaptic weight connecting neuron j to k.

Unit -1

Types of Learning

Boltzmann Learning

. The machine operates by choosing a neuron at random—for example, neuron k—at some step of the learning process, then flipping the state of neuron k from state x_k to state $-x_k$ at some temperature T with probability

$$P(x_k \rightarrow -x_k) = \frac{1}{1 + \exp(-\Delta E_k / T)} \tag{2.16}$$

where ΔE_k is the *energy change* (i.e., the change in the energy function of the machine)

Unit -1

Types of Learning

Boltzmann Learning – two functional groups : visible and hidden

- Visible neurons provide an interface between the network and the environment which it operates
- Hidden neurons always operate freely
- Two modes of operation
- Clamped Condition – in which visible neurons are all clamped onto specific states determined by environment
- Free running condition : in which all neurons are allowed to operate freely

Unit -1

Types of Learning

Boltzmann Learning – two functional groups : visible and hidden

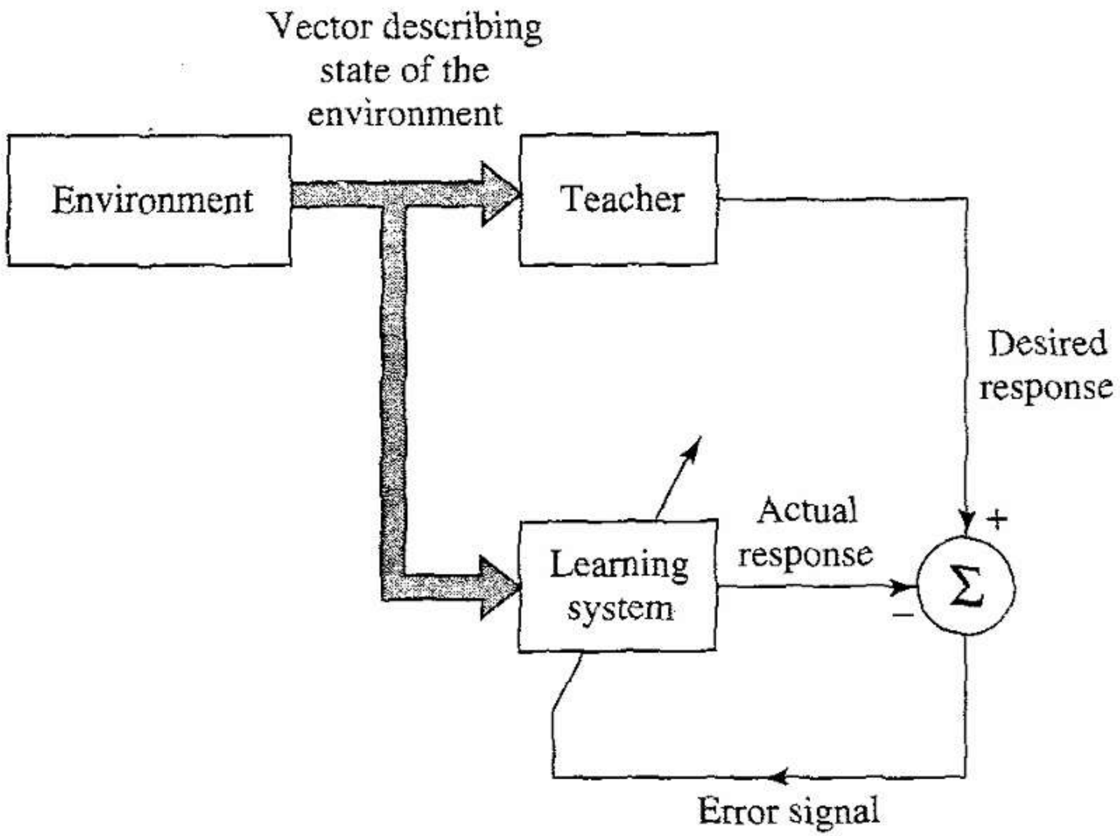
Let ρ_{kj}^{+} denote the *correlation* between the states of neurons j and k, with the network in its clamped condition. Let ρ_{kj}^{-} denote the *correlation* between the states of neurons j and k with the network in its free-running condition. Both correlations are averaged over all possible states of the machine when it is in thermal equilibrium. Then, according to the *Boltzmann learning rule*, the change Δw_{kj} applied to the synaptic weight w_{kj} from neuron j to neuron k is defined by (Hinton and Sejnowski, 1986)

$$\Delta w_{kj} = \eta(\rho_{kj}^{+} - \rho_{kj}^{-}), \qquad j \neq k \qquad (2.17)$$

Unit -1

Types of Learning

Learning with a Teacher

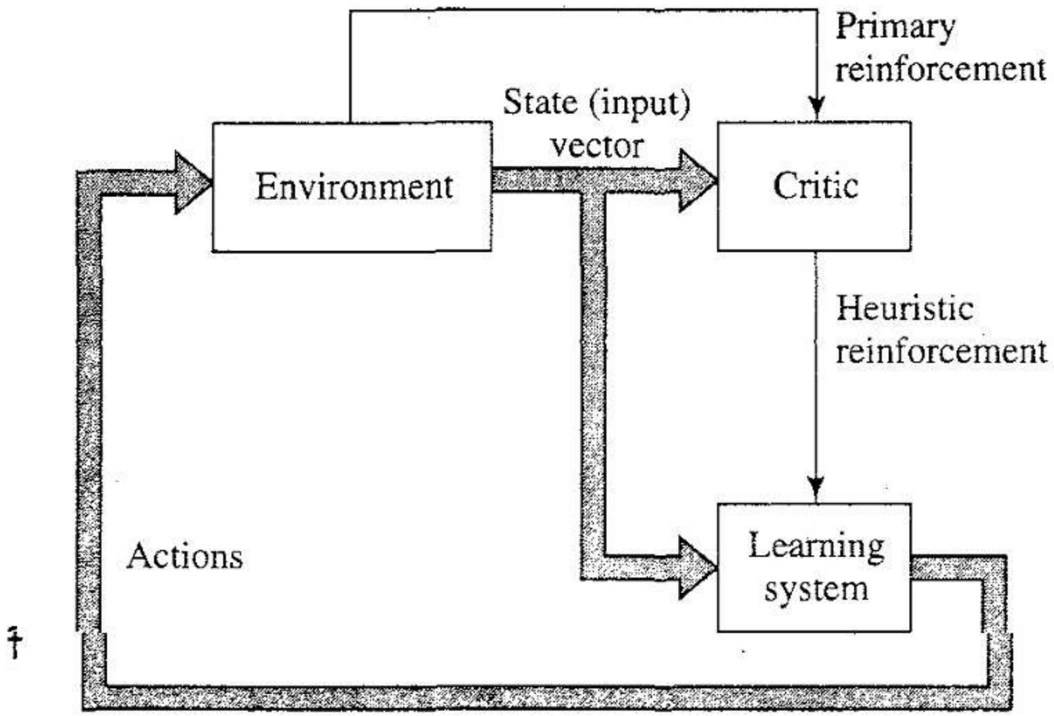


Unit -1

Types of Learning

Learning without a Teacher

1. Reinforcement Learning



Unit -1

Types of Learning

Reinforcement Learning

- The system is designed to learn under delayed reinforcement , ie the system observes a temporal sequence of stimul recd from environment
- Goal of learning is to minimize a cost-to-go function , defined as the expectation of cumulative cost of actions taken over a sequence of steps instead of immediate cost

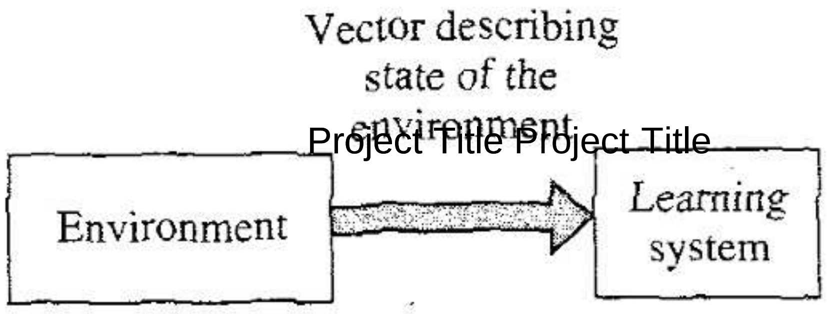
Delayed-reinforcement learning is difficult to perform for two basic reasons:

- There is no teacher to provide a desired response at each step of the learning process.
- The delay incurred in the generation of the primary reinforcement signal implies that the learning machine must solve a *temporal credit assignment problem*. By this we mean that the learning machine must be able to assign credit and blame individually to each action in the sequence of time steps that led to the final outcome, while the primary reinforcement may only evaluate the outcome.

Unit -1

Types of Learning

Unsupervised Learning

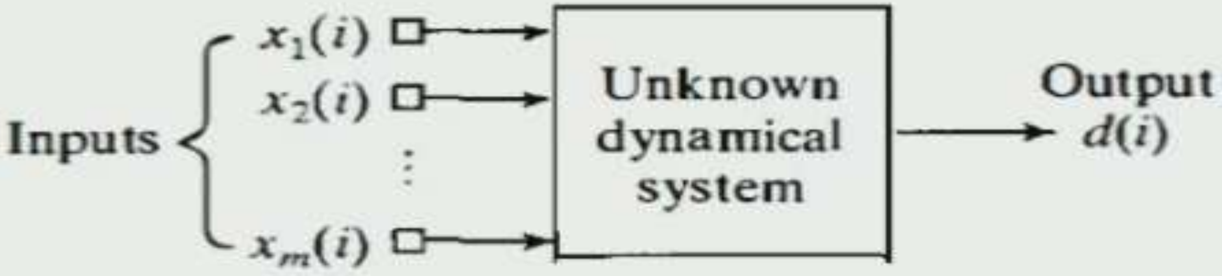


Project Title Project Title

Adaptive Filter Problem

Consider a dynamic system with an unknown mathematical characterization

A set of labeled input-output data generated by the system at discrete instants of time at some uniform rate is available



Thus, the external behaviour of the system is described by the data set

$$\mathcal{T}: \{\mathbf{x}(i), d(i); i = 1, 2, \ldots, n, \ldots\}$$

Where,

$$\mathbf{x}(i) = [x_1(i), x_2(i), \ldots, x_m(i)]^T$$

The stimulus x(i) can arise in one of two fundamentally different ways. one spatial and the other temporal:

- The m elements of $\mathbf{x}(i)$ originate at different points in space; in this case we speak of $\mathbf{x}(i)$ as a *snapshot* of data.
- The m elements of $\mathbf{x}(i)$ represent the set of present and $(m - 1)$ past values of some excitation that are *uniformly spaced in time*.

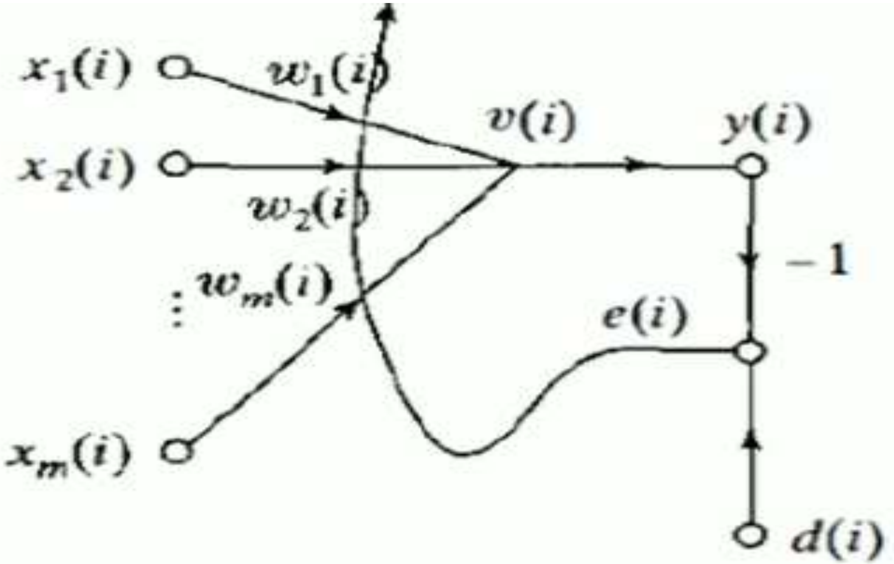
How to design a multiple input-single output model of the unknown dynamical system by building it around a single linear neuron.

The neuronal model operates under the influence of an algorithm that controls necessary adjustments to the synaptic weights of the neuron

- The algorithm starts from an arbitrary setting of the neuron's synaptic weights.

- Adjustments to the synaptic weights. in response to statistical variations in the system's behaviour, are made on a continuous basis.

- Computations of adjustments to the synaptic weights are completed inside a time interval that is one sampling period long.



Its operation consists of two continuous processes:

1. Filtering process

2. Adaptive process

Since the neuron is linear, the output y(i) is exactly the same as the induced local field v(i); that is,

$$y(i) = v(i) = \sum_{k=1}^{m} w_k(i)x_k(i)$$

$$y(i) = \mathbf{x}^T(i)\mathbf{w}(i)$$

$$\mathbf{w}(i) = [w_1(i), w_2(i), \ldots, w_m(i)]^T$$

1. *Filtering process*, which involves the computation of two signals:
   - An output, denoted by $y(i)$, that is produced in response to the m elements of the stimulus vector $\mathbf{x}(i)$, namely, $x_1(i), x_2(i), \ldots, x_m(i)$.
   - An error signal, denoted by $e(i)$, that is obtained by comparing the output $y(i)$ to the corresponding output $d(i)$ produced by the unknown system. In effect, $d(i)$ acts as a *desired response* or *target signal*.
2. *Adaptive process*, which involves the automatic adjustment of the svnaptic weights of the neuron in accordance with the error signal $e(i)$.