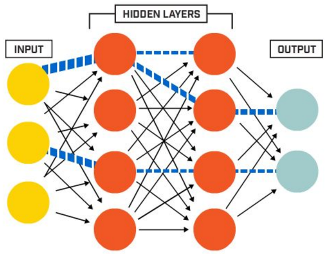
ARTIFICIAL NEURAL NETWORK AND DEEP LEARNING
21AI63



Dr S Anupama Kumar
Associate Professor
Dept of AIML
RVCE

Semester: VI						
ARTIFICIAL NEURAL NETWORK AND DEEP LEARNING **Category: Professional Core Course** **(Theory & Lab)**						
Course Code	:	**21AI63**		**CIE Marks**	:	**100+50 Marks**
Credits: L: T:P	:	**3:0:1**		**SEE Marks**	:	**100+50 Marks**
Total Hours	:	**45L+30P**		**SEE Duration**	:	**3 Hours**

Course Outcomes: After completing the course, the students will be able to:-	
CO1	Describe basic concepts of neural network, its applications and various learning models
CO2	Analyze different Network Architectures, learning tasks, convolutional networks, and deep learning models
CO3	Investigate and apply neural networks model and learning techniques to solve problems related to society and industry
CO4	Demonstrate a prototype application developed using any NN tools and APIs
CO5	Appraise the knowledge of Neural Networks and Deep Learning as an Individual /as a team member

Unit – I	9 Hrs
Neural Networks: **Introduction:** What is a Neural Network? Models of a Neuron, Network Architectures **Learning Processes:** Error-correction learning, memory-based learning, Hebbian learning, Competitive learning and Boltzmann learning, Learning with a teacher, Learning without a teacher, Learning tasks, Memory and adaptation. Statistical Learning Theory, VC dimension	
Unit – II	**9 Hrs**
Single-layer Perceptron: Adaptive Filtering Problem, Unconstrained Optimization Techniques, Steepest Descent, Least-Mean-Square Algorithm, Learning Curves, Learning rate annealing techniques, Perceptron and Convergence theorem **Multilayer Perceptron:** Back-propagation Algorithm, Sequential and Batch Modes of training, Stopping Criteria, XOR problem, Heuristics for BP algorithm to perform better	
Unit – III	**10 Hrs**
Convolutional Neural Networks: **Introduction:** Historical Perspective and Biological Inspiration, Broader Observations About Convolutional Neural Networks **The Basic Structure of a Convolutional Network:** Padding, Strides, Typical Settings, The ReLU Layer, Pooling, Fully Connected Layers , The Interleaving Between Layers , Local Response Normalization , Hierarchical Feature Engineering **Training a Convolutional Network:** Backpropagating Through Convolutions , Backpropagation as Convolution with Inverted/Transposed Filter, Convolution/Backpropagation as Matrix Multiplications, Data Augmentation **Applications of Convolutional Networks:** Content-Based Image Retrieval , Object Localization, Object Detection , Natural Language and Sequence Learning , Video Classification	

Unit – IV	10 Hrs

Recurrent Neural Networks
Introduction: Expressiveness of Recurrent Networks,
The Architecture of Recurrent Neural Networks: Language Modeling Example of RNN, Generating a Language Sample, Backpropagation Through Time, Bidirectional Recurrent Networks, Multilayer Recurrent Networks

Echo-State Networks, Long Short-Term Memory (LSTM) , Gated Recurrent Units (GRUs)
Applications of Recurrent Neural Networks: Application to Automatic Image Captioning, Sequence-to-Sequence Learning and Machine Translation , Question-Answering Systems, Application to Sentence-Level Classification , Token-Level Classification with Linguistic Features, Time-Series Forecasting and Prediction, Temporal Recommender Systems, Secondary Protein Structure Prediction
End-to-End Speech Recognition Handwriting Recognition

Unit – V	10 Hrs

Deep Reinforcement Learning : Introduction
Stateless Algorithms: Multi-Armed Bandits: Naïve Algorithm, Greedy Algorithm, Upper Bounding Methods
The Basic Framework of Reinforcement Learning: Challenges of Reinforcement Learning, Simple Reinforcement Learning for Tic-Tac-Toe, Role of Deep Learning and a Straw-Man Algorithm
Bootstrapping for Value Function Learning: Deep Learning Models as Function Approximators, Example: Neural Network for Atari Setting, On-Policy Versus Off-Policy Methods: SARSA, Modeling States Versus State-Action Pairs
Monte Carlo Tree Search
Case Studies: AlphaGo: Championship Level Play at Go, Alpha Zero: Enhancements to Zero Human Knowledge , Self-Learning Robots, Deep Learning of Locomotion Skills, Deep Learning of Visuomotor Skills, Building Conversational Systems: Deep Learning for Chatbots, Self-Driving Cars

Laboratory Component

Group of two students belongs to same batch are required to implement an engineering application using any one of the deep learning techniques, CNN and architectures, RNN or Reinforcement learning.

Examples:

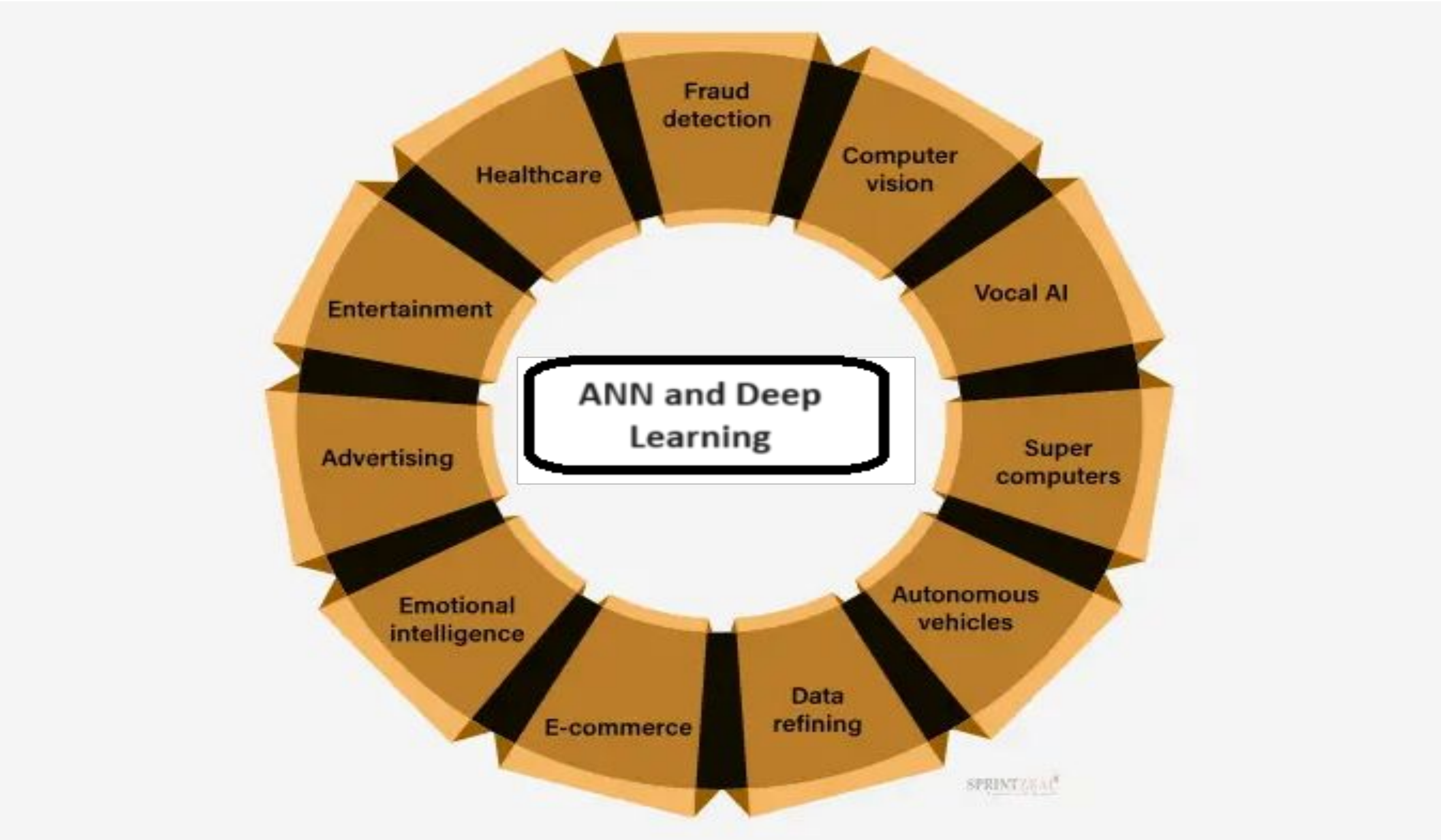
CNN: Biometric authentication using CNN, Object identification and recognition, Emotion recognition, Auto translation, document classification, etc.

RNN: Language translation, Generating image descriptions, Speech recognition, etc,

Reinforcement learning: Real-time bidding, Recommendation Systems, Traffic Control Systems, etc.

The laboratory component will be evaluated in two phases :

- **Phase I –** 25 marks which includes identification of the problem, data set collection and preprocessing, selection of appropriate algorithm and its justification
- **Phase II -** Implementation of the deep learning model with appropriate GUI, performance and evaluation of the model .


Fraud detection
Computer vision
Vocal AI
Super computers
Autonomous vehicles
Data refining
E-commerce
Emotional intelligence
Advertising
Entertainment
Healthcare
ANN and Deep Learning

Unit -1

- **Neural Networks**
  - What - Human Brain vs NN
  - Models
  - Neural networks as directed graphs
  - Feedbacks
  - Architectures
  - AI and NN
- **Learning Processes**
  - Error correction Learning
  - Memory Based Learning
  - Hebbian Learning
  - Competitive Learning
  - Boltzmann Learning
  - Learning with and without teacher
  - Learning tasks

Unit -1

- **Neural Networks**
  - Brain – complex, non linear and parallel computer - capability of organize its structural constituents called neurons so as to perform tasks like pattern recognition, perception, motor control etc
  - 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 eg speech signal
  - Input – Output Mapping -- Reordering of previously trained examples and mapping Input – output using estimation statistics
  - Adaptivity -- Stability Plasticity Dilemma
  - 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
  - VLSI Implementability -- well suited due to massive parallelism
  - Uniformity of Analysis and Design -- commonality, share and integrity

Unit -1

- ## Human Brain

The **human brain** contains about 10 billion nerve cells, or neurons. On average, each neuron is connected to other neurons through approximately 10,000 synapses.

	processing elements	element size	energy use	processing speed	style of computation	fault tolerant	learns	intelligent, conscious
	10^{14} synapses	10^{-6} m	30 W	100 Hz	parallel, distributed	yes	yes	usually
	10^{8} transistors	10^{-6} m	30 W (CPU)	10^{9} Hz	serial, centralized	no	a little	not (yet)

Unit -1

Characteristics of BNN and ANN

Characteristics	Biological Neural Network	Artificial Neural Network
Speed	Processes information at a slower rate. Response time is measured in milliseconds.	Information is processed at a faster rate. The response time is measured in nanoseconds.
Processing	Massively parallel processing.	Serial processing.
Size & Complexity	An extremely intricate and dense network of linked neurons of the order of 1011 neurons and 1015 interconnections.	Size and complexity are reduced. It is incapable of performing sophisticated pattern recognition tasks.
Storage	An extremely intricate and dense network of linked neurons with 1015 interconnections, including neurons on the order of 1011.	The term "replaceable information storage" refers to the practice of replacing fresh data with old data.
Fault tolerance	The fact that information storage is flexible means that new information may be added by altering the connectivity strengths without deleting existing information.	Intolerant of faults. In the event of a system failure, corrupt data cannot be recovered.
Control Mechanism	There is no unique control mechanism outside of the computational task.	Controlling computer activity is handled by a control unit.

Unit -1

	Biological Neuron	Artificial Neuron
Speed	Milli secs	Nano Secs
Processing	Parallel	Parallel – Can be faster than human brains
Size and Complexity	Better	Less than Biological Neuron
Storage	based on synapse strength	Contiguous Memory Allocation
Tolerance	Fault Tolerant, Accept Redundancy, performs if some cells die	Not Fault Tolerant, Do not accept Redundancy, system damages if network is disturbed
Control Mechanism	Depends of the active chemicals present in brain	Has a CPU which works very precisely

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

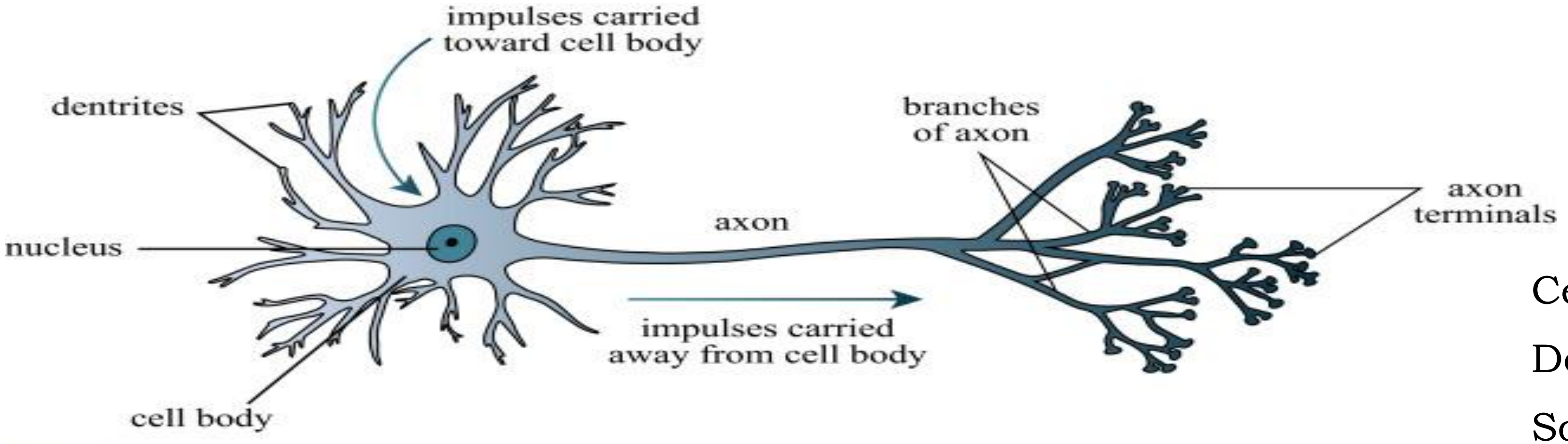
Dendrites have fibers branching out from the soma in a bushy network around the nerve cell.

- 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

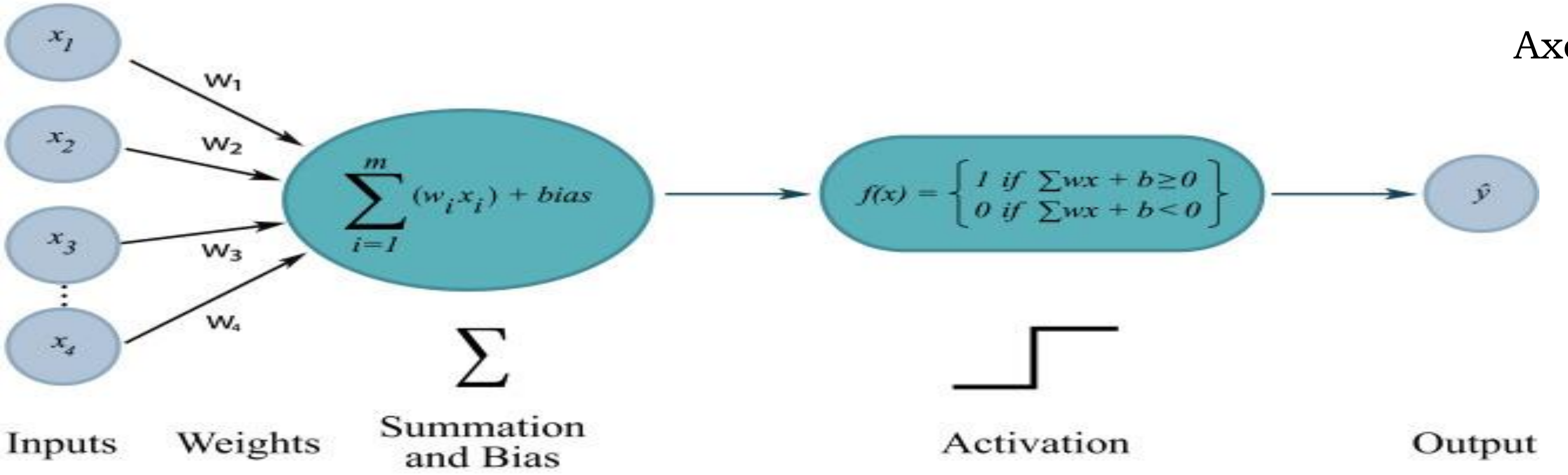


Cell – Neuron

Dendrites –Weight or interconnections

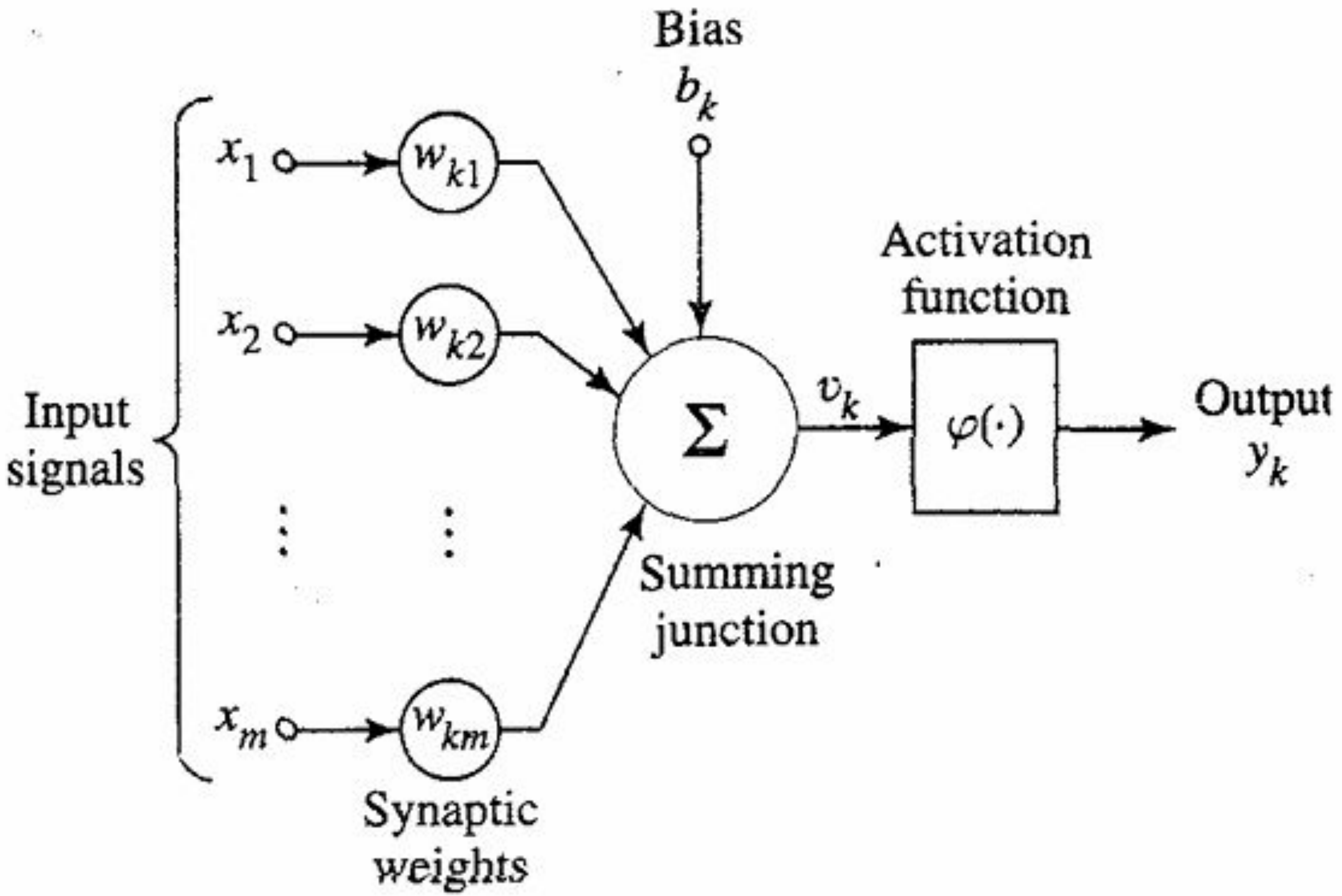
Soma – Net Input

Axon –Output



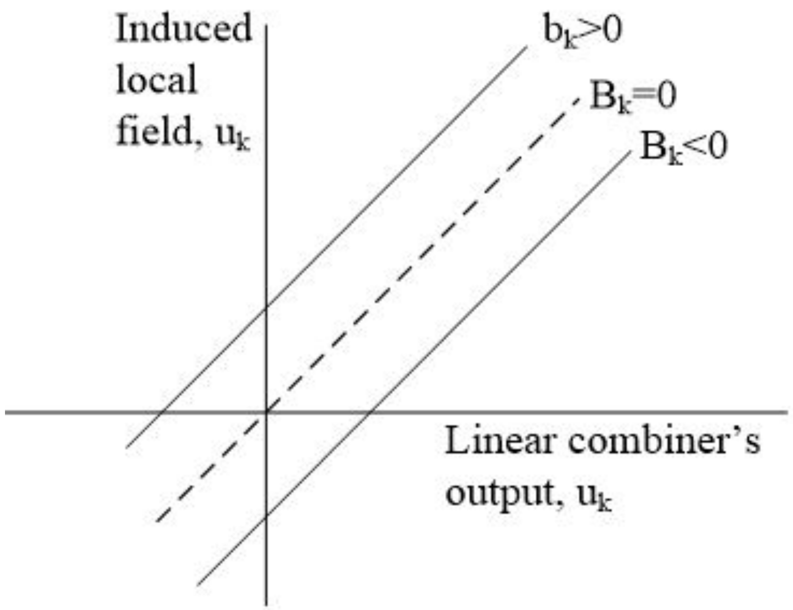
Unit -1

Non

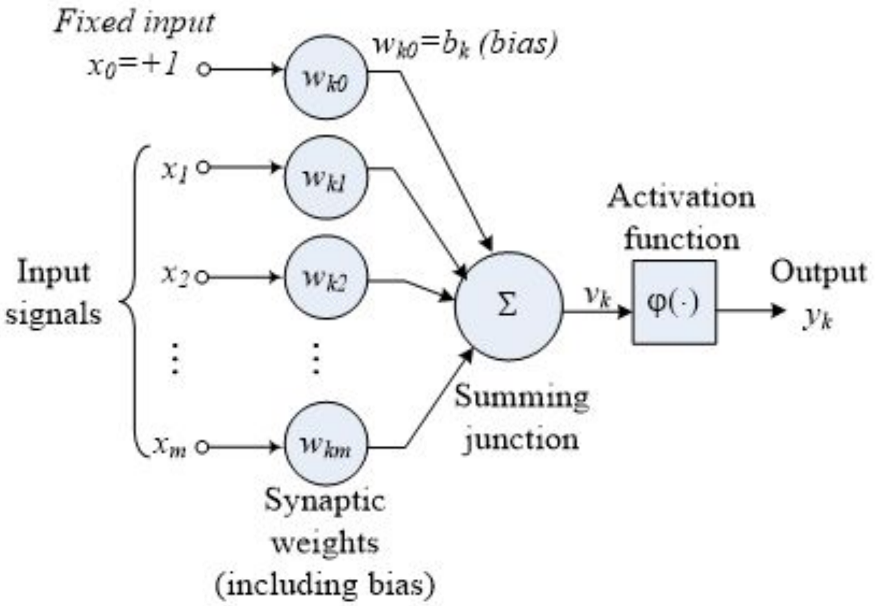

Bias
b_k
x_1
w_{k1}
x_2
w_{k2}
Input signals
x_m
w_{km}
Synaptic weights
Σ
Summing junction
v_k
Activation function
$\varphi(\cdot)$
Output
y_k

Unit -1

Non Linear Model of a Neuron



Affine transformation produced by the presence of a bias



Another Nonlinear model of a neuron

Unit -1

Functional Parameters of a Neuron

Function parameter	Description
w	Vector of real-valued weights on the connections
$\mathbf{w} \cdot \mathbf{x}$	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

Functional Parameters of a Neuron

Weight : Each Neuron is connected to other by means of communication link and each link is associated with weight.

- Weights contain information about the input signal that is used by the net to solve a problem.
- Represented in the form of a matrix and called as connection matrix.

Bias : Included in the network and has its impact in calculating the net input.

- Two types – Positive and Negative Bias
- Positive – helps to increase the net input
- Negative – Decreases the net input

Unit -1

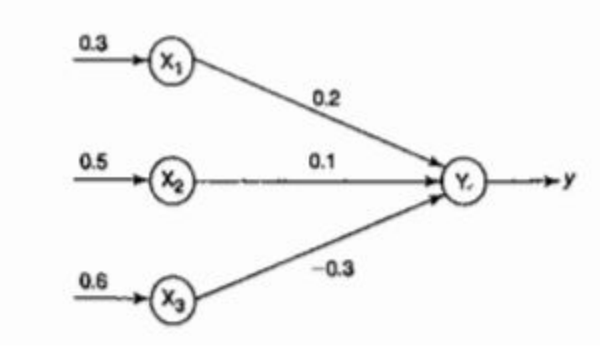
Functional Parameters of a Neuron

- Threshold : Is a set value based upon which the final output of the network is calculated.
  - Used in activation function
- Learning Rate : Denoted by ∞ . Used to control the amount of weight adjustment at each step of training
  - Ranges from 0 to 1
- Momentum Factor : Convergence is made faster if this is used.
  - Weights from one or more previous training patterns will be saved

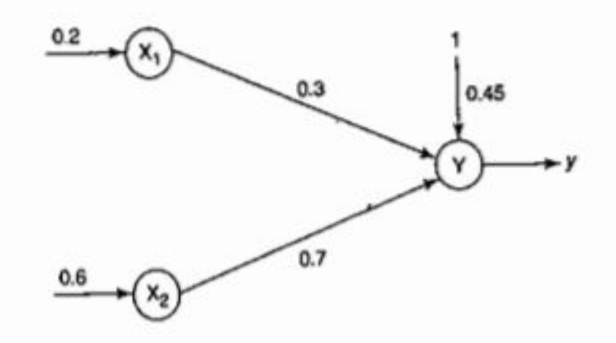
Unit -1

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

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

Threshold Function

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

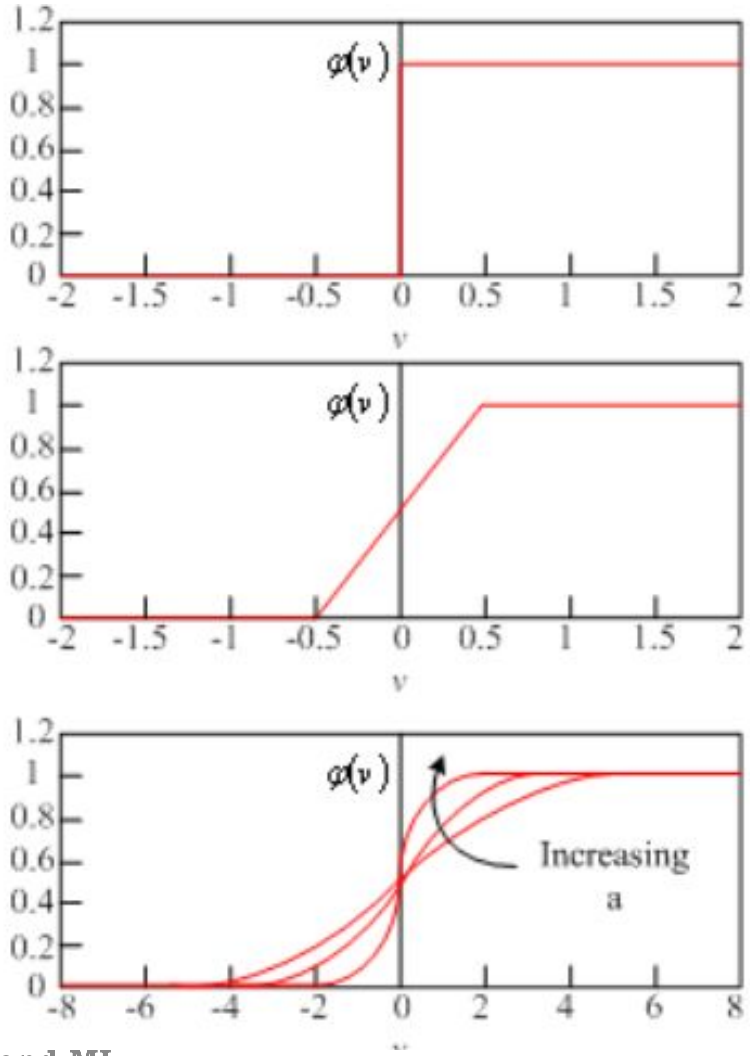
Piecewise-Linear Function

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

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.

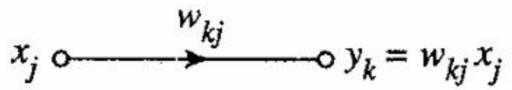
Unit -1

Neural Networks as Directed Graphs – Signal Flow Diagrams

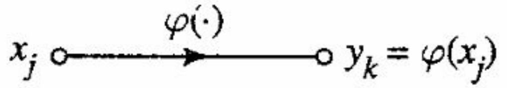
Rule 1. A signal flows along a link only in the direction defined by the arrow on the link.

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 $\varphi(\cdot)$ is the nonlinear activation function.



(a)



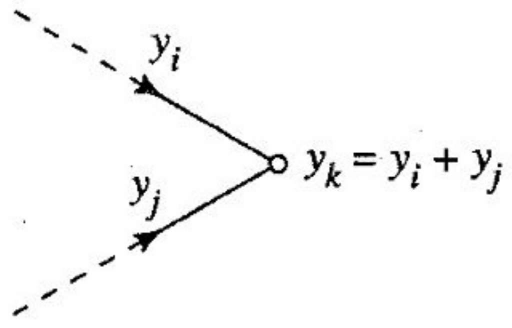
(b)

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*.



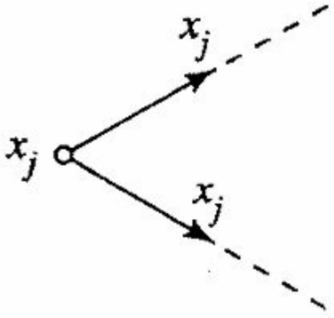
(c)

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.

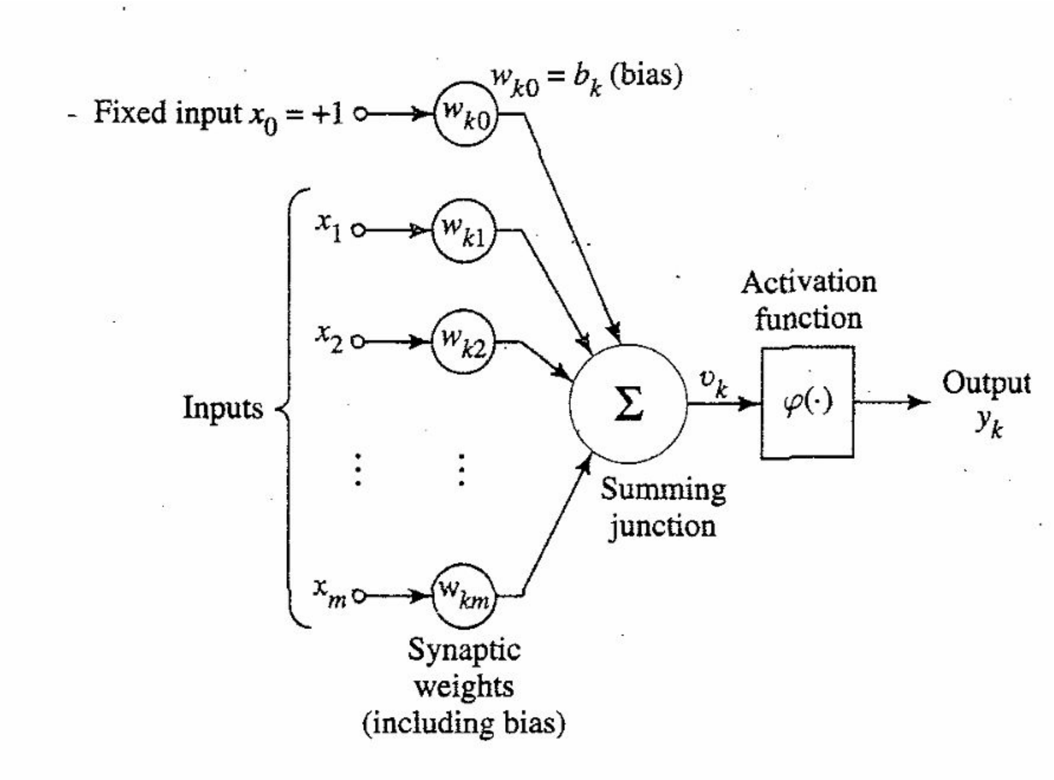
This graph illustrates the synaptic divergence or fanout



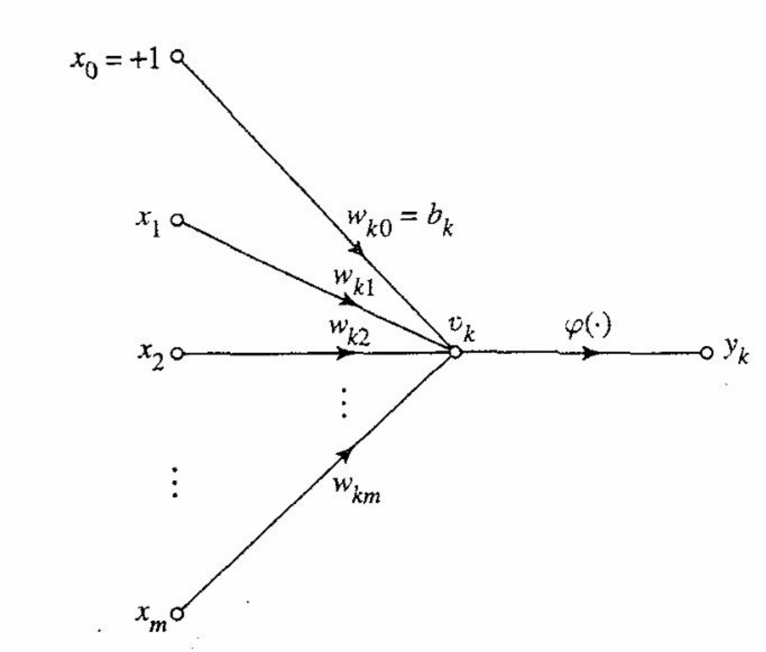
(d)

Unit -1

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

Neural Networks as Directed Graphs

Mathematical definition of a NN

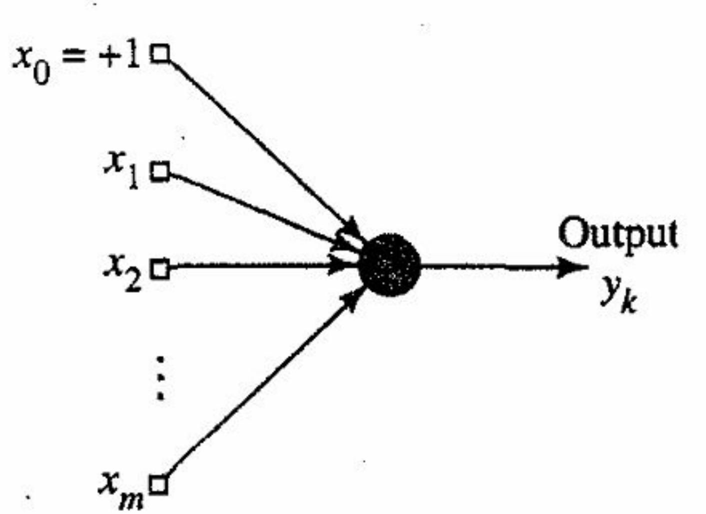
A neural network is a directed graph consisting of nodes with interconnecting synaptic and activation links, and is characterized by four properties:

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.*

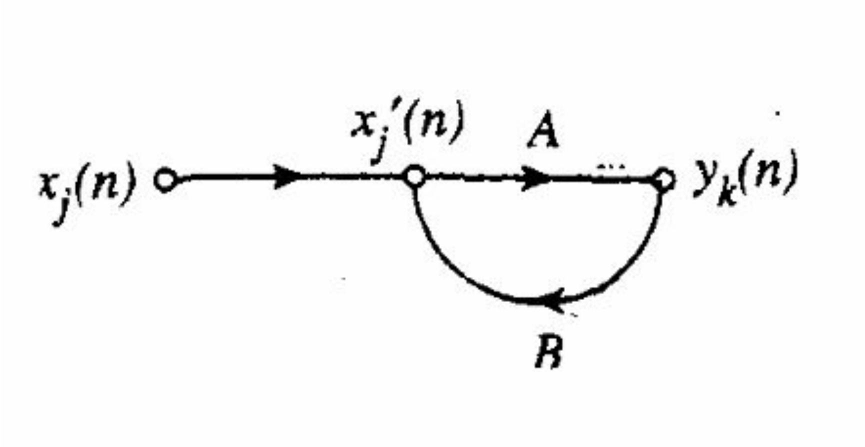
Unit -1

Feedback

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



Architectural Graph

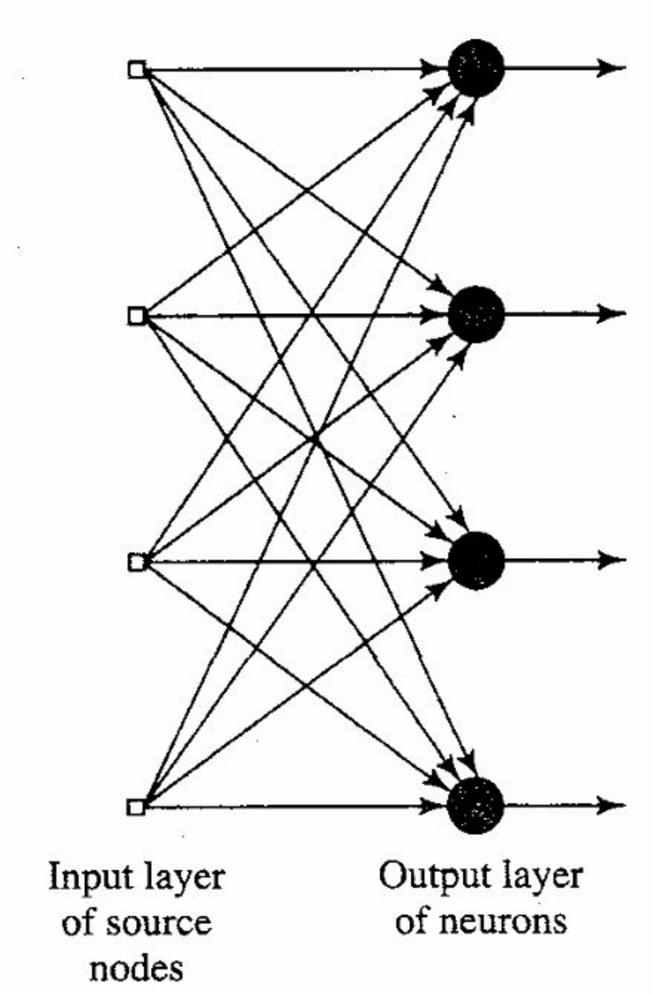


Signal Flow Graph

Unit -1

Network Architeures

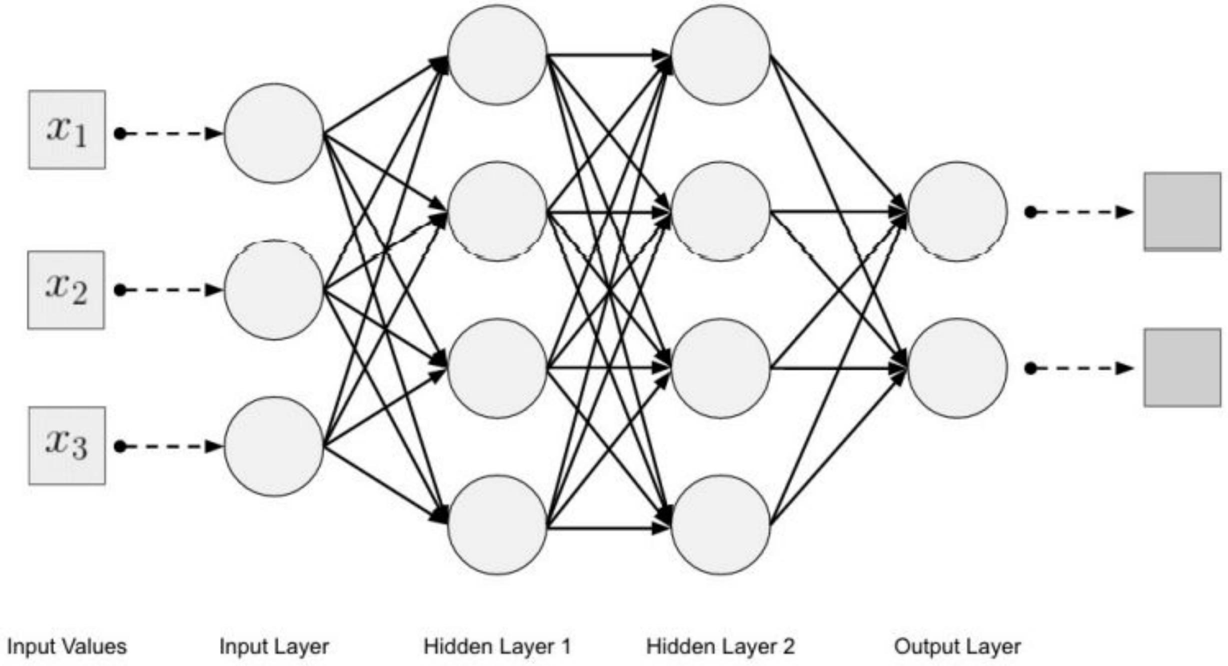
- Single Layer Feedforward network
  - Consists of input layer of source nodes that projects to output layer of computation node
  - Strictly feedforward or acylic type
  - Single output layer of computation nodes
- Layer is formed by taking a processing element and combining it with other processing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



Unit -1

Network Architeures

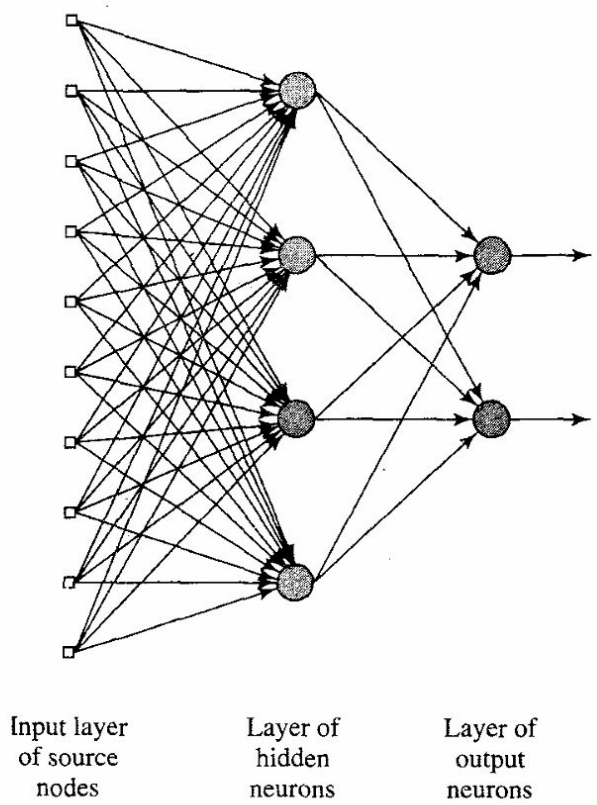
- Multi Layer Feedforward network



Unit -1

Network Architeures

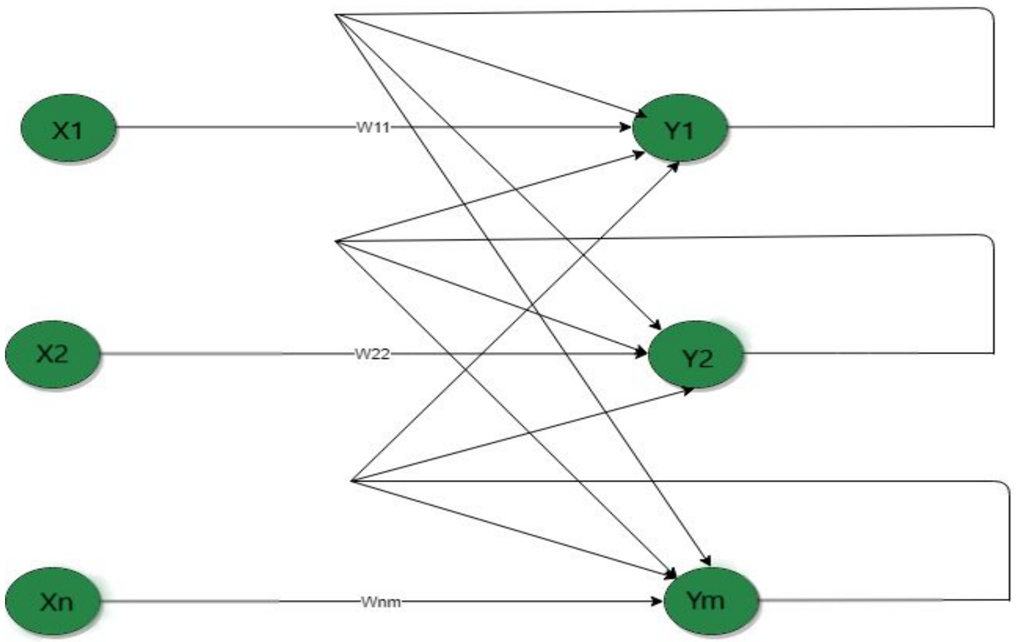
- Multi Layer Feedforward network
    - Consists of input layer , hidden layer and output layer
- Each layer can have a different number of neurons and each layer is fully connected to the adjacent layer.
- The connections between the neurons in the layers form an acyclic graph



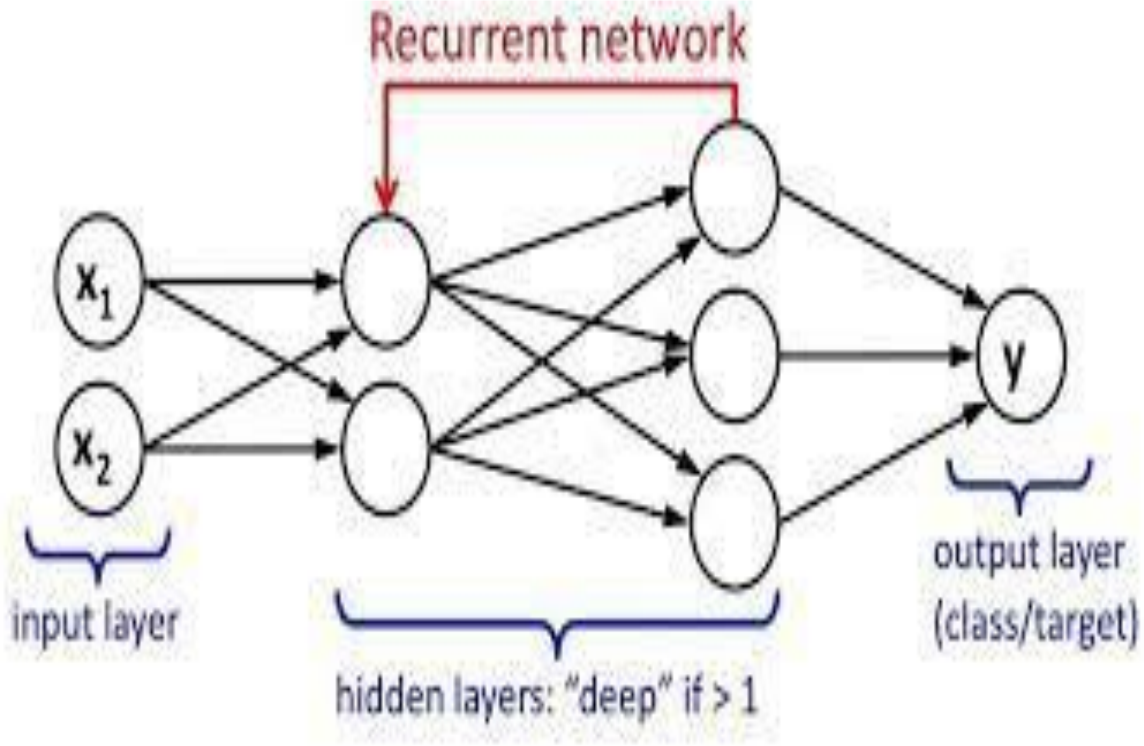
Unit -1

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

Single Layer Recurrent Network



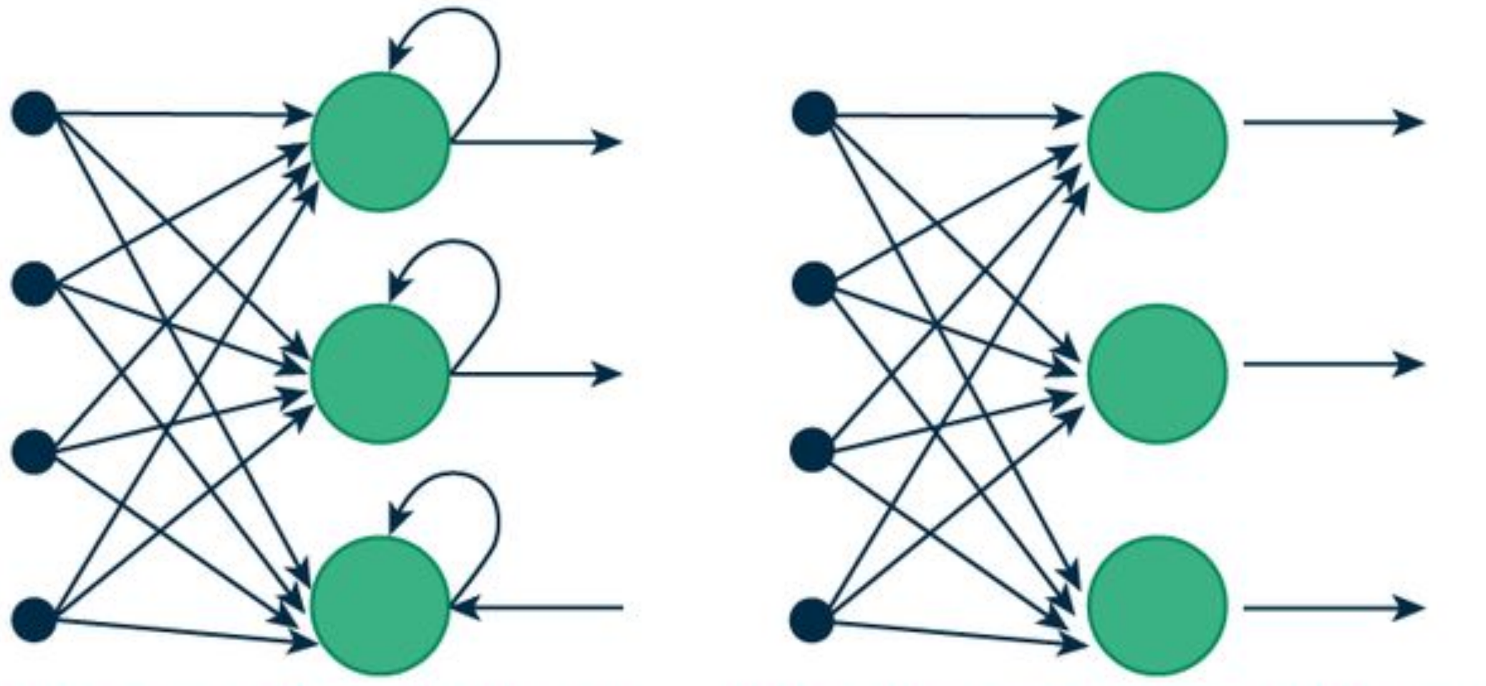
Multi Layer Recurrent Network



RV College of Engineering®
Mysore Road, RV Vidyaniketan Post,
Bengaluru - 560059, Karnataka, India

Unit -1

Recurrent Neural Networks Vs Feed forward Networks



(a) Recurrent Neural Network (b) Feed-Forward Neural Network

Unit -1

Knowledge Representation

Knowledge refers to stored information or models used by a person or machine to interpret, predict, and appropriately respond to the outside world.

Two Primary Characteristics

- What information is actually made explicit
- How the information is physically encoded for use

NN learns a model of the world (environment) and knowledge of the world consists of

1. Known world state – prior information
2. Observations (measurements) designed to probe environment where NN operates.
3. Observations can be noisy with errors due to the imperfection

Unit -1

Knowledge Representation

Observation provide a pool of information from which the examples are used to train NN

The examples can be labelled / unlabeled

Given a set of examples, the NN can be designed

Knowledge Representation

Example : Handwritten Recognition system

Phase I : Learning Phase

Select an appropriate architecture –

input layer consisting of source nodes equal in number to the pixels of an input image

Output layer consisting of 10 neurons(1 / digit)

Train the network using example

Phase II : Generalization

Apply test data which the NN has not seen before

Assess the performance with the actual identity

Knowledge Representation

Example : Handwritten Recognition system

Phase I : Learning Phase

Select an appropriate architecture –

input layer consisting of source nodes equal in number to the pixels of an input image

Output layer consisting of 10 neurons(1 / digit)

Train the network using example

Phase II : Generalization

Apply test data which the NN has not seen before

Assess the performance with the actual identity

Unit -1

Knowledge Representation

Rules for Knowledge Representation

Rule 1. Similar inputs from similar classes should usually produce similar representations inside the network, and should therefore be classified as belonging to the same category.

Rule 2. Items to be categorized as separate classes should be given widely different representations in the network.

Rule 3. If a particular feature is important, then there should be a large number of neurons involved in the representation of that item in the network.

Rule 4. Prior information and invariances should be built into the design of a neural network, thereby simplifying the network design by not having to learn them.

Unit -1

Knowledge Representation

Characteristics from Rule 4:

1. Biological visual and auditory networks are known to be very specialized.
2. A neural network with specialized structure usually has a smaller number of free parameters available for adjustment than a fully connected network. Consequently, the specialized network requires a smaller data set for training, learns faster, and often generalizes better.
3. The rate of information transmission through a specialized network (i.e., the network throughput) is accelerated.
4. The cost of building a specialized network is reduced because of its smaller size, compared to its fully connected counterpart.

Unit -1

Knowledge Representation

Inference from Rule 4 to build a specialized NN:

1. Biological visual and auditory networks are known to be very specialized.
2. A neural network with specialized structure usually has a smaller number of free parameters available for adjustment than a fully connected network. Consequently, the specialized network requires a smaller data set for training, learns faster, and often generalizes better.
3. The rate of information transmission through a specialized network (i.e., the network throughput) is accelerated.
4. The cost of building a specialized network is reduced because of its smaller size, compared to its fully connected counterpart.

Unit -1

Knowledge Representation

Building NN using Prior Information

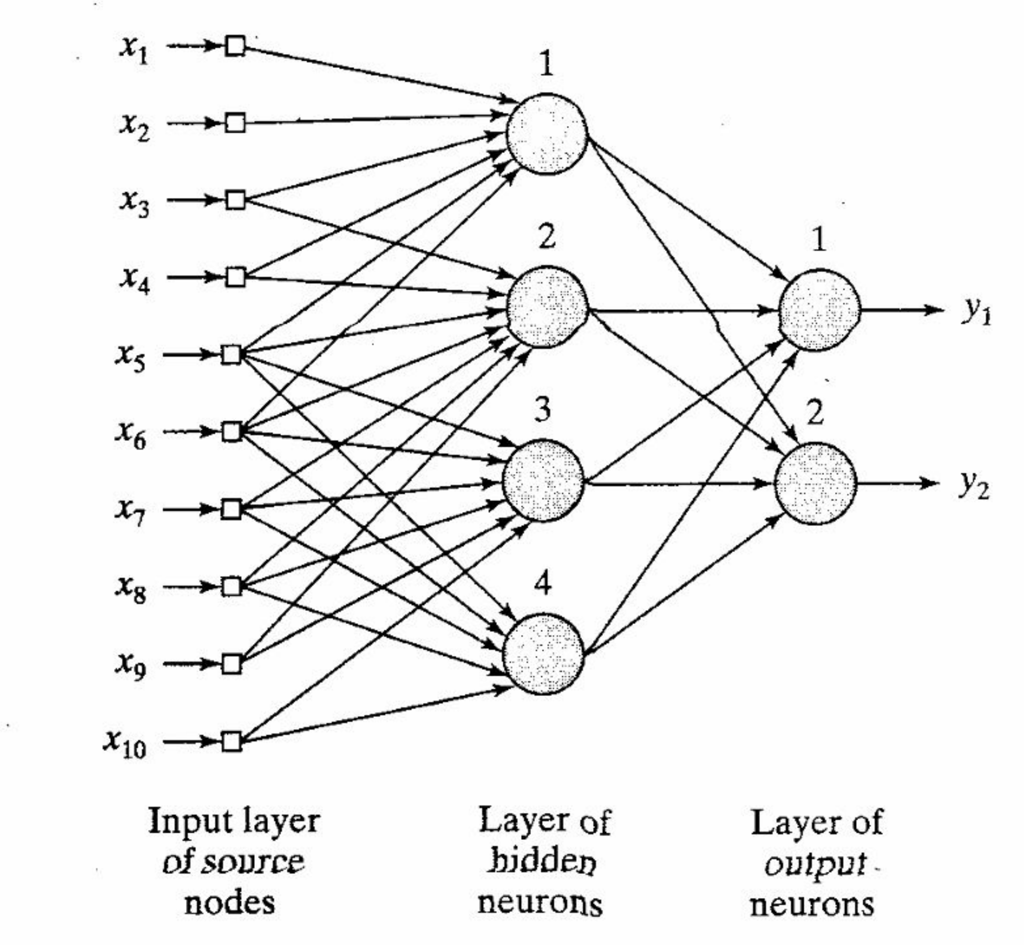
1. *Restricting the network architecture* through the use of local connections known as *receptive fields*.[5]
2. *Constraining the choice of synaptic weights* through the use of *weight-sharing*.[6]

From Rule 2, the number of free parameters is reduced,

Unit -1

Knowledge Representation

Building NN using Prior Information



Unit -1

Knowledge Representation

Error Correction Learning

Memory Based Learning

Hebbian Learning

Competitive Learning

Boltzmann Learning

Learning with Teacher

Learning without Teacher

Unit -1

Knowledge Representation

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

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

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

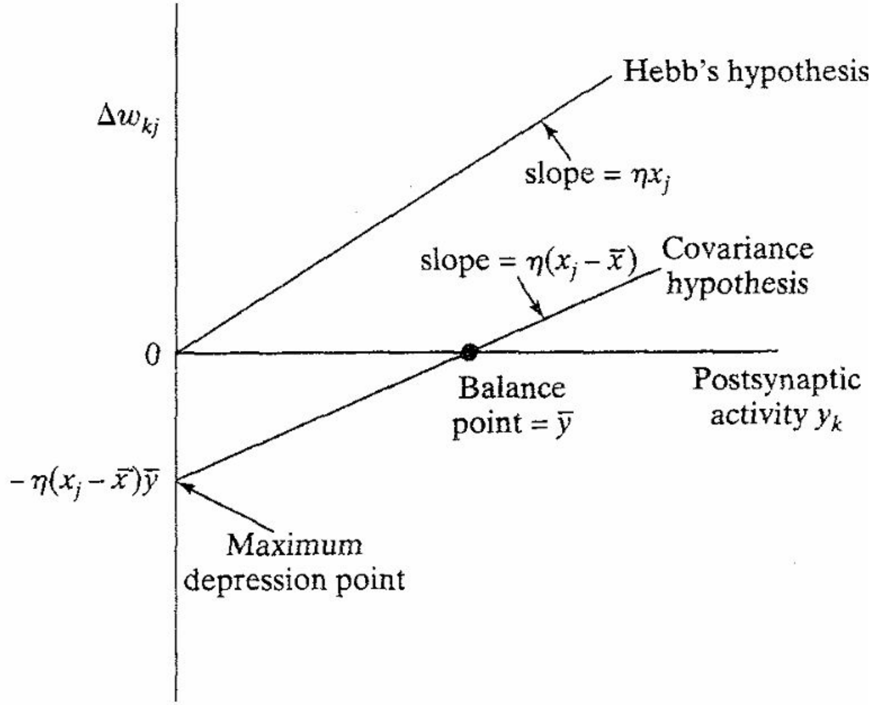
Form 2 :

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 :

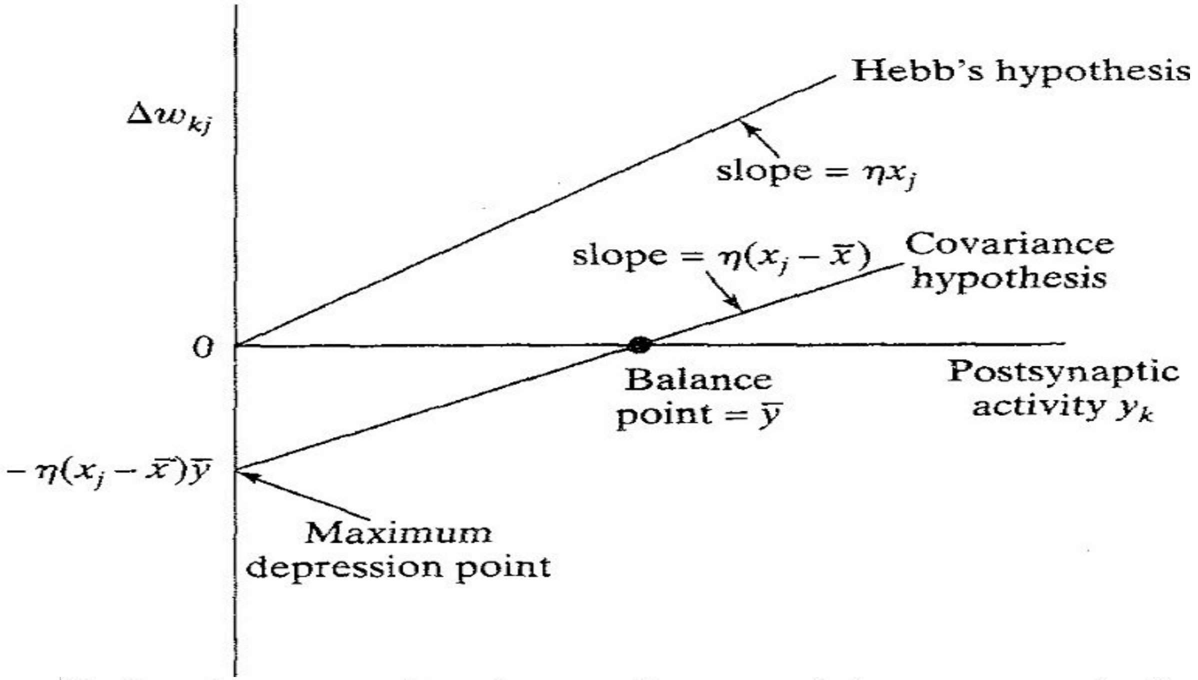
covariance hypothesis allows for the following:

- 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

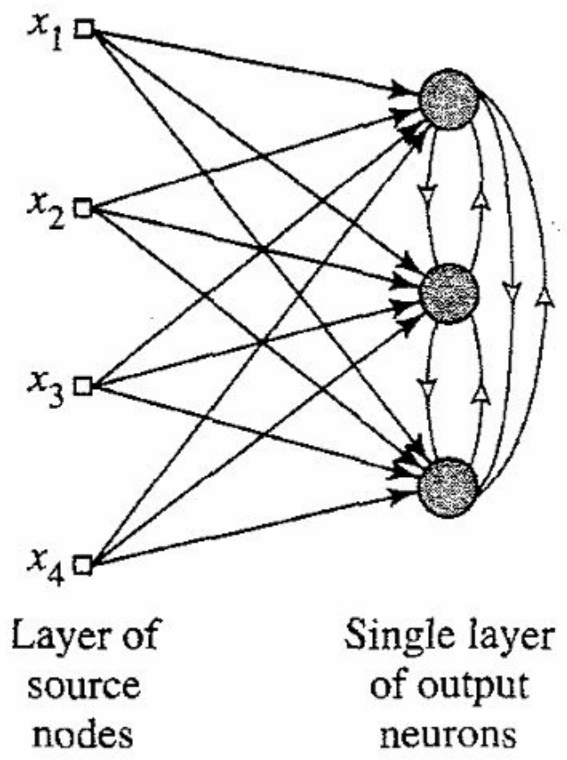


FIGURE 2.4 Architectural graph of a simple competitive learning network with feedforward (excitatory) connections from the source nodes to the neurons, and lateral (inhibitory) connections among the neurons; the lateral connections are signified by open arrows.

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

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

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

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

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

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

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

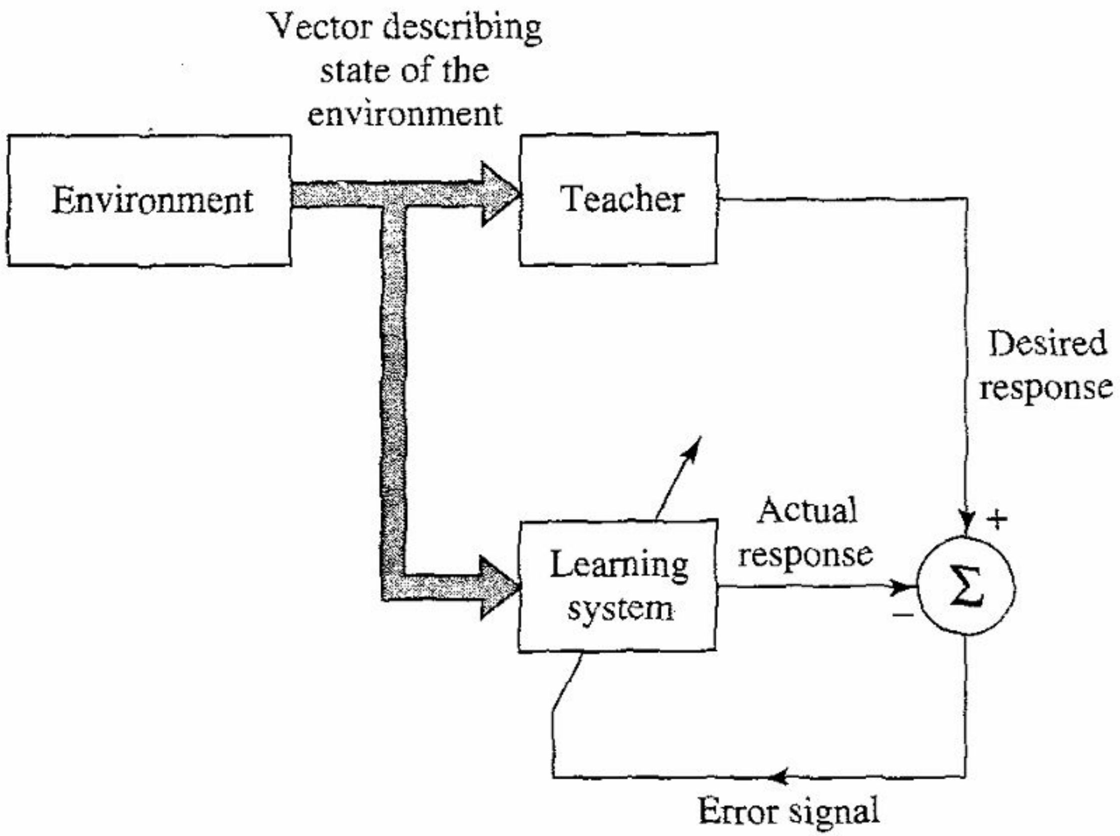
Let ρ_{kj}^{+} denote the *correlation* between the states of neurons j and k, with the network in its clamped condition. Let ρ_{kj}^{-} denote the *correlation* between the states of neurons j and k with the network in its free-running condition. Both correlations are averaged over all possible states 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)

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

Unit -1

Types of Learning

Learning with a Teacher

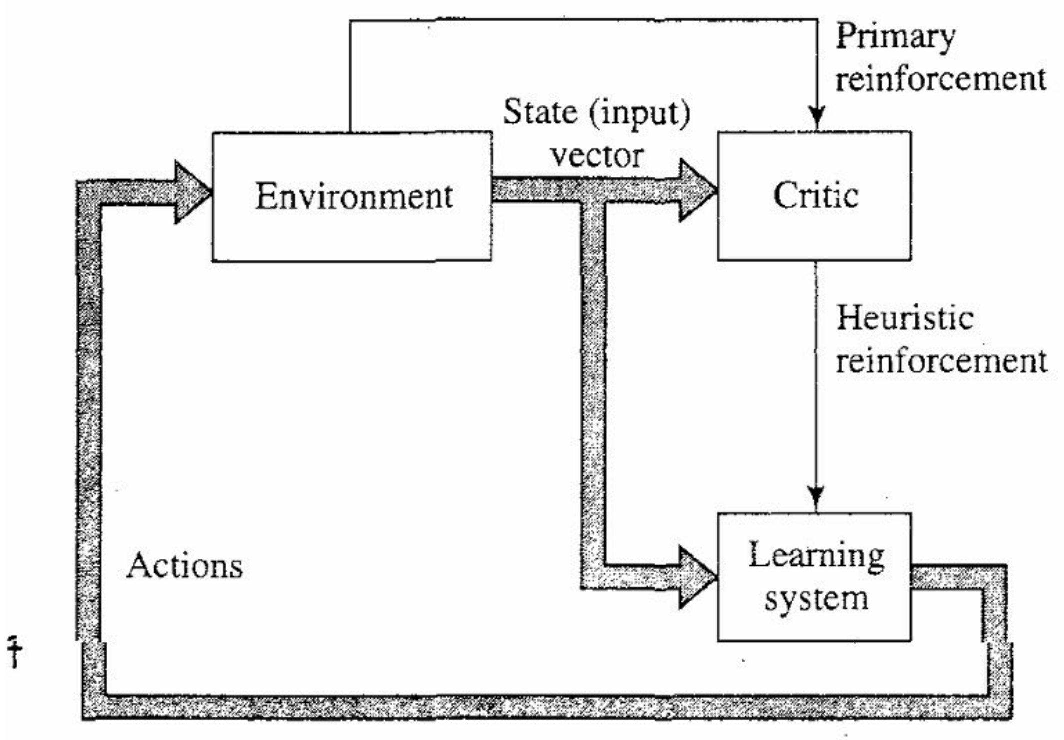


Unit -1

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

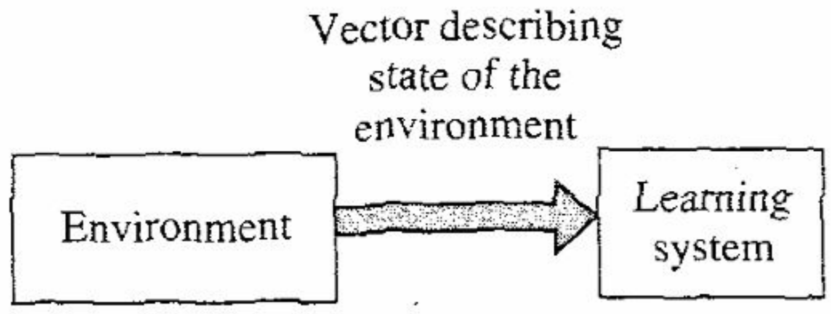
Delayed-reinforcement learning is difficult to perform for two basic reasons:

- 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



Unit -1

Types of Learning

Learning Tasks

- Pattern Association
- Patter Recognition
- Function Approximation
- Control
- Filtering
- Beamforming

Unit -1

Learning Tasks

Pattern Association

Associative memory is a brain type distributed memory that learns from associations

Two forms : Auto association , Hetroassociation

Auto association – NN is required to store a set of patterns(vectors) by repeatedly presenting them to the network

the network is presented with a partial description or distorted noisy version and the task is to retrieve (recall) the correct pattern.

unsupervised learning

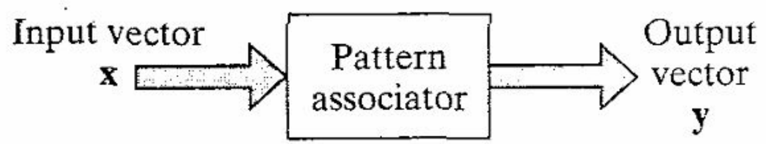
Hetroassociation – the arbitrary inputs are paired with another arbitrary output pattern

supervised learning

Unit -1

Learning Tasks

Pattern Association



Let $\mathbf{x}_k$ denote a *key pattern* (vector) applied to an associative memory and $\mathbf{y}_k$ denote a *memorized pattern* (vector). The pattern association performed by the network is described by

$$\mathbf{x}_k \rightarrow \mathbf{y}_k, \qquad k = 1, 2, \ldots, q \tag{2.18}$$

where q is the number of patterns stored in the network. The key pattern $\mathbf{x}_k$ acts as a stimulus that not only determines the storage location of memorized pattern $\mathbf{y}_k$, but also holds the key for its retrieval.

In an autoassociative memory, $\mathbf{y}_k = \mathbf{x}_k$, so the input and output (data) spaces of the network have the same dimensionality.

In a heteroassociative memory, $\mathbf{y}_k \neq \mathbf{x}_k$;

the dimensionality of the output space in this second case may or may not equal the dimensionality of the input space.

Unit -1

Learning Tasks

Pattern Association

Two Phases of Operation of Associative Memory

- *Storage phase*, which refers to the training of the network in accordance with Eq. (2.18).
- *Recall phase*, which involves the retrieval of a memorized pattern in response to the presentation of a noisy or distorted version of a key pattern to the network.

Pattern Recognition

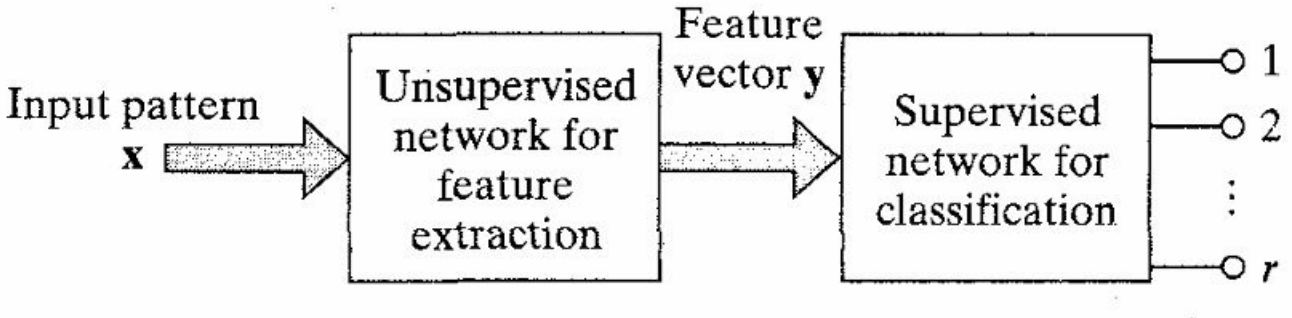
Pattern recognition is formally defined as *the process whereby a received pattern/signal is assigned to one of a prescribed number of classes (categories).*

A NN undergoes a training session, where the patterns are presented and then new pattern is presented to identify which pattern it belongs to.

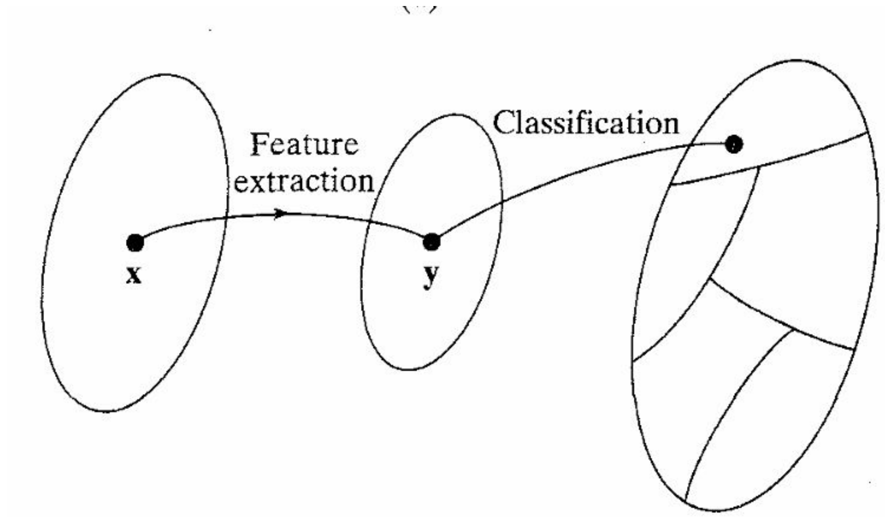
Unit -1

Learning Tasks

Pattern Recognition – Two Forms



(a)



(b)

Unit -1

Learning Tasks

Function Approximation

Consider the non linear input – output mapping of the relationship $\mathbf{d} = \mathbf{f}(\mathbf{x})$

where vector x is input and vector d is output. The function f(.) is assumed as unknown. We are given with the labelled examples to make up the lack of knowledge $\mathcal{T} = \{(\mathbf{x}_i, \mathbf{d}_i)\}_{i=1}^{N}$

The requirement is to design a neural network that approximates the unknown function $\mathbf{f}(\cdot)$ such that the function $\mathbf{F}(\cdot)$ describing the input–output mapping actually realized by the network is close enough to $\mathbf{f}(\cdot)$ in a Euclidean sense over all inputs, as shown by

$$\|\mathbf{F}(\mathbf{x}) - \mathbf{f}(\mathbf{x})\| < \epsilon \qquad \text{for all } \mathbf{x}$$

where ϵ is a small positive number. Provided that the size N of the training set is large enough and the network is equipped with an adequate number of free parameters, then the approximation error ϵ can be made small enough for the task.

Unit -1

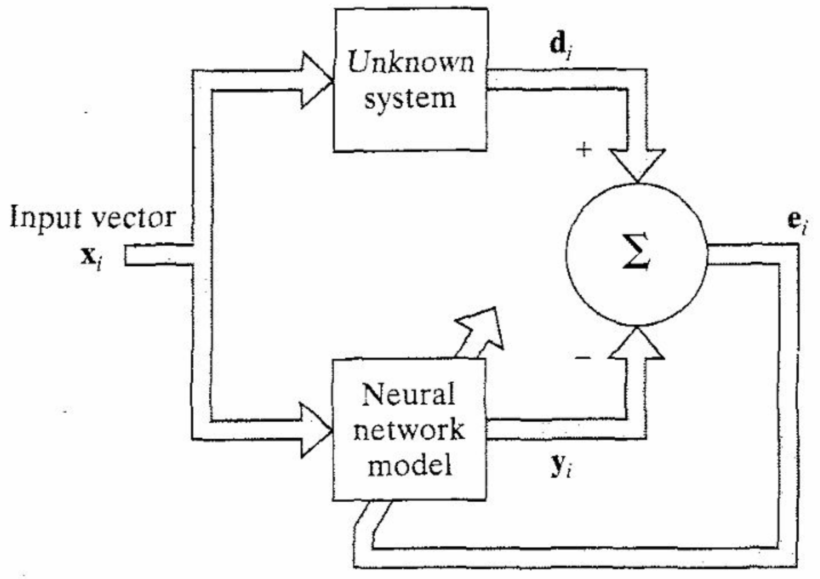
Learning Tasks

Function Approximation

To approximate an unknown input – output mapping two methods are used.

1. System Identification

$\mathbf{d} = \mathbf{f}(\mathbf{x})$ is a memoryless multiple input – output system (MIMO) ie time variant



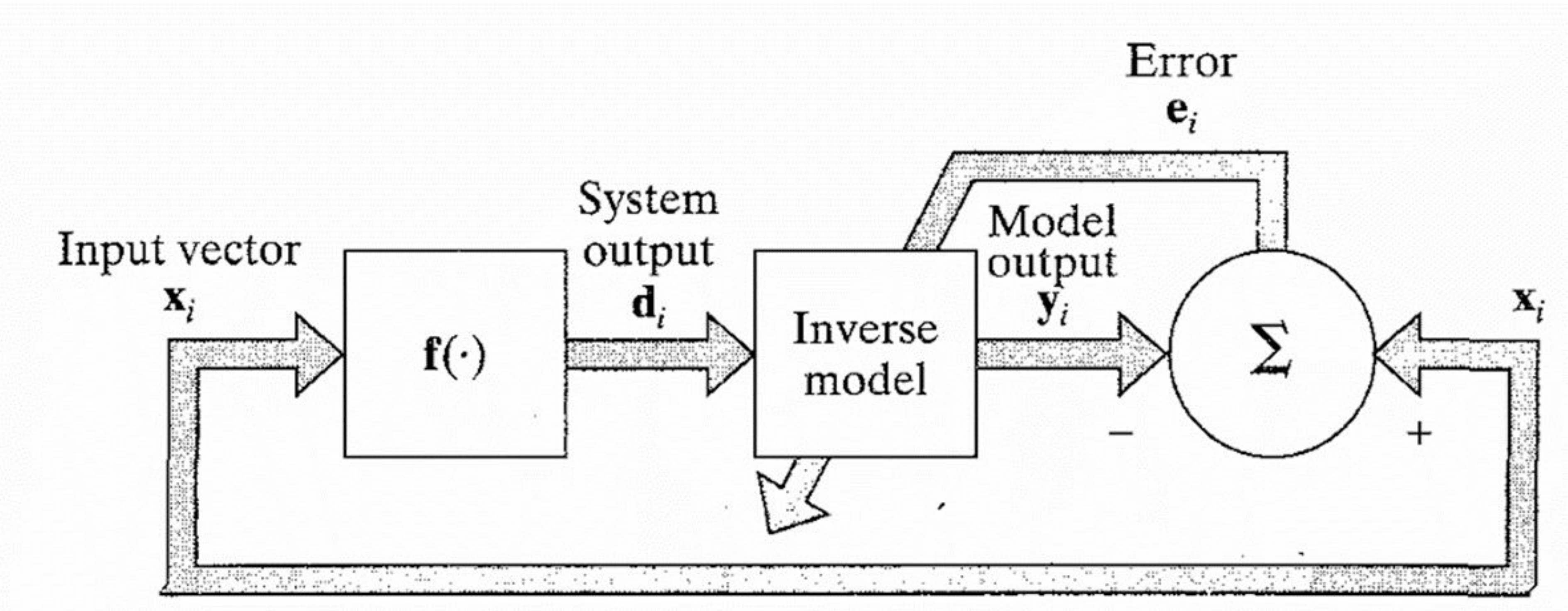
Unit -1

Learning Tasks

Function Approximation

Inverse System : The inverse system may thus be described by

$$\mathbf{x} = \mathbf{f}^{-1}(\mathbf{d})$$



Unit -1

Learning Tasks

Control

- The control of a plant is a learning task that can be done by a nn
- “Plant” is a process or critical part of a system that is to be maintained in a controlled condition
- The objective is to supply appropriate inputs to the plant to make its output y track the reference signal d.
- The controller has to invert the plant’s input-output behavior
- The objective is to supply appropriate inputs to the plant to make its output y track the reference signal d.
- The controller has to invert the plant’s input-output behavior

Unit -1

Learning Tasks

Control - Example

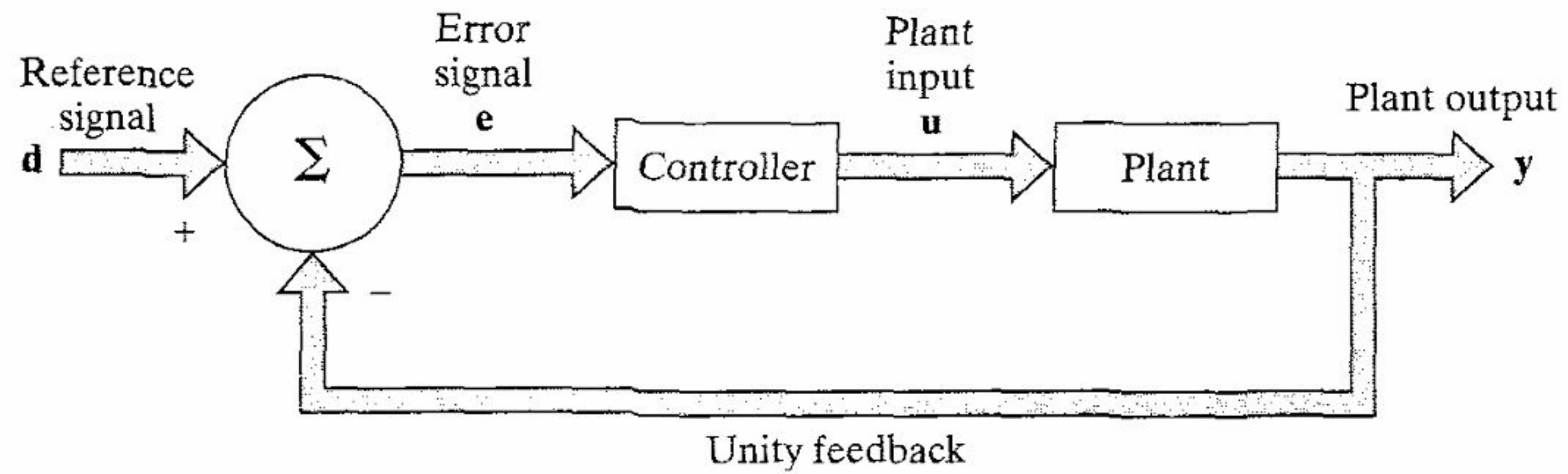


FIGURE 2.13 Block diagram of feedback control system.

- The system involves the use of unity feedback around the plant to be controlled.
- The plant output y is subtracted from a reference signal d supplied from an external source
- The error signal e so produced is applied to a neural controller for the purpose of adjusting free parameters.

Unit -1

Learning Tasks

Control

The signal e has to propagate through the neural controller before reaching the plant.

The adjustments on the free parameters of the plant is given by Jacobian matrix

$$\mathbf{J} = \left\{ \frac{\partial y_k}{\partial u_j} \right\}$$

where y_k is an element of the plant output $\mathbf{y}$ and u_j is an element of the plant input $\mathbf{u}$.

Unfortunately, the partial derivatives $\partial y_k / \partial u_j$ for various k and j depend on the operating point of the plant and are therefore not known. We may use one of two approaches to account for them:

1. Indirect Learning
2. Direct Learning

Unit -1

Learning Tasks

Control

Indirect Learning

- Using the actual input-output measurement on the plant , a nn model is constructed to produce a copy of it
- It is used to provide estimate of the Jacobian matrix J
- The partial derivates are used in error correction learning algorithm for computing the adjustments to the free parameters of the neural controller

Direct Learning

The signs of the partial derivatives $\partial y_k/\partial u_j$ are generally known and usually remain constant over the dynamic range of the plant.

we may approximate these partial derivatives by their individual signs.

absolute values are given a distributed representation in the free parameters of the neural controller

Unit -1

Learning Tasks

Filtering

- Filter refers to a device or algorithm used to extract information about a prescribed quantity of interest from a set of noisy data.
- The noise may arise from a variety of sources
- Filter can be used to perform

Filtering - refers to the extraction of information about a quantity of interest at discrete time n by using data measured up to and including time n.

Smoothing - information about the quantity of interest need not be available at time n, and data measured later than time n can be used in obtaining information

- In smoothing there is a delay in producing result of interest

Unit -1

Learning Tasks

Filtering

Prediction - task is to forecast the information processing.

Derive information about what the quantity of interest will be like at some time $n + n_0$ in the future, for some $n_0 > 0$, by using data measured up to and including time n.

Beamforming

- Spatial form of filtering and is used to distinguish between the spatial properties of a target signal and background noise
- The devise used is called as beamformer

Beamforming is commonly used in radar and sonar systems where the primary task is to detect and track a target of interest in the combined presence of receiver noise and interfering signals (e.g., jammers). This task is complicated by two factors.

- The target signal originates from an unknown direction.
- There is no *a priori* information available on the interfering signals.