

Unit 1

Introduction:

- What is a neural network?

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 the brain in two respects:

1. Knowledge is acquired by the network from its environment through a learning process.
2. Inter neuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.

- The procedure used to perform the learning process is called a “learning algorithm”, the function of which is to modify the synaptic weights of the network in an orderly fashion to attain a desired design objective.

- “Generalization” refers to the neural network producing reasonable o/p’s for i/p’s not encountered during training (learning).

- Benefits of neural network:-

1. Non liberality: - An artificial neuron can be linear (or) non linear. The non linearity is of a special kind in the sense that is distributed throughout the network.
2. Input –output mapping: - A popular paradigm of learning called “learning with a teacher (or) supervised learning” involves modification of the synaptic weights of a neural network by applying a set of labeled “training samples (or) task examples”. Thus the network learns from the examples by constructing an i/p – o/p mapping.
3. Adaptivity: - Neural network are have a built in capability to adapt their synaptic weights to changes in the surrounding environment. A neural network trained to operate in a specific environment can be easily trained to deal with minor changes in the operating environment conditions.
4. Evidential Response: - A neural network can be design to provide information not only about which particular pattern to select but also about the confidence in the decision made.
5. Contextual information: - Knowledge is represented by the very structure & activation state of a neural network. Contextual information is deal with naturally by a neural network.
6. Fault tolerance:- A neural network implemented in hardware from, has the potential to be inherently fault tolerant (or) capable of robust computation, in the sense that its performance degrades gracefully under adverse operating conditions.

7. VLSI implement ability: - The massively parallel nature of a neural network makes it potentially fast for the computation of certain tasks. This feature of neural network is well suited for implementation of lying “very large scale integrated technology (VLSI)” which provides a means of capturing truly complex behavior in a highly hierarchical form.
8. Uniformity of Analysis & Design: - Neural network enjoy universality as information processor. This feature manifests into:-
- * Neurons, in one form (or) another, represent an ingredient common to all neural network
 - * The commonality makes it possible to share theories & learning algorithms in different applications of neural networks.
 - * Modular networks can be built through a seamless integration of modules.
9. Neuron biological analogy: - The design of a neural network is motivated by analogy with the brain, which is a living proof that fault tolerant parallel processing is not only physically possible but also fast & powerful.

➤ Human and Computers:-

| Computers | Human Brain |
|--|---|
| 1. One (or) a few high speed (ns) processors with considerable computing power | 1. Large # (10 ⁹) of low speed processor (ms) with limited computing power. |
| 2. One (or) a few shared high speed buses for communication. | 2. Large # (10 ¹⁵) of low speed connections. |
| 3. Sequential memory access by address. | 3. Content addressable recall (CAM) |
| 4. Problem solving knowledge is separated from the computing component. | 4. Problem solving knowledge resides in the connectivity of neurons. |
| 5. Hard to be adaptive | 5. Adaptation by changing the connectivity. |

➤ Organization of the brain:-

- The human nervous system can be viewed as a three stage system.
- Central to the system is the brain represented by the neural network, which continually receives information, perceives it and makes appropriate decision.

- The receptors convert stimuli from the human body (or) the external environment into electrical impulses that convey information to the neural network.
 - The effectors convert electrical impulses generated by the neural network into discernible responses as system o/p's.
 - The arrows pointing from left to right indicate the "forward transmission" of information bearing signal through the system.
 - The arrows pointing from right to left signify the presence of "feedback" in the system.
-
- The energetic efficiency of the brain is approximately 10^{-16} joules (J) per operation per second, whereas the corresponding value for the best computers in use today is about 10^{-16} Joules per operation per second.
 - Synapses are elementary structural and functional units that mediate the interactions b/w neurons.
 - The most common kind is chemical synapse a transmitter substance that diffuses across the synaptic function b/w neurons & then acts on a post synaptic process.
 - Axons the transmission lines & dendrites, the receptive zones, constitute two types of all filaments that are distinguished on morphological grounds.
 - An axon has a smoother surface, fewer branches & greater length whereas a dendrite by an irregular surface & more branches.
 - Neurons come in a wide variety of shapes sizes in different parts of the brain.
 - The hierarchical model of the brain is shown in the following figure.

Fig: Structural organization of levels in the brain

- Synapses represent the most fundamental level depending on molecules & ions for their action.
- Neural microcircuit refers to an assembly of synapses organized into patterns of connectivity to produce a functional operation of interest.
- The neural microcircuits are grouped to form dendritic subunits within the dendritic trees of individual neurons.
- The whole neuron about 100 μm in size contains several dendritic subunits.
- At the next level of complexity, they have local circuits made up of neurons with similar (or) different properties.
- This is followed by interregional circuits made up of pathways, columns & topographic maps which involve multiple regions located in different parts of the brain.
- At the final level of complexity, the topographic maps & other interregion circuits mediate specific types of behaviors in the central nervous system.

➤ Biological neuron model:-

- The information processing cells of the brain are the neurons.
- The structure of a neuron is shown in the fig.

➤ As in figure, each neuron has a soma (or) cell body which contains the cell's nucleus & other vital components called organelles which perform specialized tasks.

➤ Its main common links are:-

- A set of dendrites which form a tree like structure that spreads out from the cell. The neuron receives its i/p electrical signals along these.
- A signal axon which is a tubular extension from the cell soma that carries an electrical signal away from the soma to another neuron for processing.
- ❖ The dendrites & axon together are sometimes called "processes of the cell".
- ❖ A dendrites tree typically starts out as a narrow extension from the soma & then forms a very dense structure by repeated branching.
- ❖ Membranes that form dendrites are similar to the membranes of the soma & are basically extension of the cell body.
- ❖ Dendrites may also emerge from several different regions of soma.
- ❖ A neuron has only one axon which may repeatedly branch to form an axonal tree.
- ❖ An axon carries the electrical signal called an action potential to other neurons for processing.

- ❖ Axons usually terminate on the dendrites of other cells (or) on muscles.
- ❖ Due to a difference in ion concentrations inside & outside of the cell, there is a difference in electrical potential across the membrane called “resting membrane potential”.
- ❖ This potential difference will tend the ions to diffuse into & out of the cell depending upon the concentration gradient.
- ❖ Hence an electric field is set up which opposes the movement of ions.
- ❖ A stage comes when the electrical force balances the diffusive force, at this point, there is no net movement of the ions across the membrane & we say that the ion is in equilibrium. This potential is called “equilibrium potential of the ion”.
- ❖ Neuron constantly receives i/p’s from other neurons along their dendrites at points of contact called “synapses”.
- ❖ These i/p’s take the form of small electrical disturbances that are called “post synaptic potentials”.
- ❖ The cell soma receives each of these small disturbances which are superimposed upon each other & therefore the soma potential reflects a temporal integration of these potentials.
- ❖ At the point where the axon of the neuron meets the cell body the axon expands into a structure called “the axon hillock”.
- ❖ Specialized electrically gated sodium & potassium ion channels are found at the axon hillock.
- ❖ Synapses are points at which a unidirectional condition of a signal from the pre synaptic to the postsynaptic membrane takes place.
- ❖ The plasma membrane that encloses neurons is a two-layered structure about 90Å^o thick.
- ❖ There are a variety of proteins embedded in the cell membrane.
- ❖ The principle function of these proteins is to regulate the transport of ions through & across the membrane.

- Hodgkin-Huxley neuron model: - The Hodgkin-Huxley model of an axon is based on the electrical equivalent circuit (or) axonal membrane.
The electrical equivalent circuit incorporate a capacitance with three conductance’s as in fig.

The three conductances' are:-

g_L = voltage independent leak conductance.

g_{Na} = voltage dependent sodium conductance.

g_K = voltage independent potassium conductance.

- The current flow across the membrane has two major components:
 - ❖ One that charges the membrane capacitance and
 - ❖ Second that is generated by the movement of specific ions across the membrane.
- The latter ionic current can be subdivided into 3 distinct component age:
 1. A sodium current I_{Na}
 2. A potassium current I_k
 3. A small leakage current I_L primarily carried by chloride ions.
- These currents are assumed to be controlled by batteries E_{Na}, E_k, E_L that correspond to the equilibrium potentials for squid axon are
 $E_{Na} = 50 \text{ mv}, E_k = -77 \text{ mv}, E_L = -543 \text{ mv}$
- The batteries for E_{Na}, E_k are placed in series with variable condition g_{Na}, g_k & battery E_L is placed in series with a passive conductance g_L .
- The maximal value of conductance's are $_{Na} = 120, _k = 36, _L = 0.3 \text{ ms/cm}^2$
- All voltages are measured with respect to external medium which is assumed to be grounded.
- A typical value of membrane capacitance is $1 \text{ } \mu\text{F/cm}^2$
- Writing KCL at inside node,

$$C_m V_m + I_{ion} = I_{ext} \quad \square (1)$$
 Where

$$I_{ion} = I_{Na} + I_k + I_L \quad \square (2)$$

$$\therefore C_m V_m = I_{ext} - I_{Na} - I_k - I_L$$

$$= I_{ext} - g_{Na}(V_m - E_{Na}) - g_k(V_m - E_k) - g_L(V_m - E_L)$$

$$C_m V_m = I_{ext} + g_{Na}(E_{Na} - V_m) - g_k(E_k - V_m) - g_L(E_L - V_m) \quad \square (3)$$

$$\therefore \text{Equation (3) is the Hodgkin-Huxley equation describing the circuit.}$$

- Integrate and fire neuron model: - The Integrate and fire (Infrared (IF)) neuron is a simple & powerful spiking neuron model and is based on the electrical model of the neuron membrane.
- Non-leaky infrared neuron: - In the ideal (or) non-leaky infrared neuron there is a simple capacitor that is responsible for sub-threshold integration.
 - A current injection shown in figure charges the capacitor.
 - The time dependence of the capacitor voltage is generated by the first order differential equation.

$$CV_i = I_i(t) \quad (1)$$

- If $I_i(t) = I$, a constant, then equation has the solution as

$$V_i(t) = \frac{I}{C} t \quad (2)$$
- This means that for a constant “I” the capacitor voltage increases linearly with time.
- When the capacitor voltage equals the threshold V_θ , the neuron is assumed to fire an o/p pulse & this event initiates a reset action that brings the capacitor voltage back to the reset value.
- This reset value is modeled using a switch that merely short circuits the capacitor.
- The voltage waveform of the cell potential $V_i(t)$ for a constant current is shown in figure.

- The firing time of neuron t_i^k is computed from equation (2)
- Assuming that the neuron starts from zero potential, the time to spike T_i is:

$$T_i = \frac{CV_\theta}{I} \quad (3)$$

- This means that the neuron fires at a fixed frequency,

$$f = \frac{1}{T_i} = \frac{I}{CV_\theta} \quad (4)$$

$$\text{And } t_i^{k+1} = t_i^k + T_i \quad (5)$$

- In non-leaky if neuron, any i/p current, however small, will increase the charge of the capacitor.

- Therefore, any arbitrary small i/p current will eventually cause the capacitor voltage to reach the threshold causing the neuron to fire a spike.

- Leaky infrared neuron:- Neuron membranes leaks 2 as a consequence the cell potential has a tendency to decay back towards its resting value.
- To model this aspect, we place a leakage resistance in parallel with the integrating capacitor.
- The resistor row allows the capacitor voltage to discharge with a time constant.

- Writing KCL,

$$CV_i + \frac{V_i}{R} = I_i(t) \quad \square \quad (1) \quad (\text{or})$$

$$T_m \dot{V}_i = -V_i(t) + R \cdot I_i(t) \quad \square \quad (2)$$

- Equation (2) explicitly shows the decay term $-V_i(t)$ and the external current I_i as the forcing function.

- Taking Laplace transformer of equation (2)

$$T_m [s \cdot V_i(s) - V_i(\bar{0})] = -V_i(s) + R \cdot I_i(s) \quad \square \quad (3)$$

- If $I_i(t) = I$, a constant then $I_i(s) = \frac{I}{s}$.

$$\therefore (1 + s \cdot T_m) \cdot V_i(s) = T_m \cdot V_i(\bar{0}) + \frac{RI}{s} \quad \square \quad (4)$$

- $V_i(s) = \frac{V_i(\bar{0})}{s + \frac{1}{T_m}} + \frac{RI}{T_m(s + \frac{1}{T_m})} \quad \square \quad (5)$

$$\therefore V_i(t) = V_i(\bar{0}) \cdot e^{-\frac{t}{T_m}} + I \cdot R (1 - e^{-\frac{t}{T_m}}) \quad \square \quad (6)$$

- The first term is the leak age term in equation (6) and second term is the charging components.
- Once the voltage reaches the threshold, V_θ a spike is fired & the neuron goes into the reset condition where the switch is closed & capacitor voltage is reset to zero.
- An arbitrary small i/p current will no longer cause a spike to be generated.
- There has to be a minimum constant current called "the threshold current" I_θ to generate spikes.

$$I_\theta R = V_\theta \quad (\text{or}) \quad I_\theta = \frac{V_\theta}{R} \quad \square \quad (7)$$

- Spiking neuron model: - The spiking neuron model presents a general mathematical framework of a neuron retaining its biological fidelity.
- Referring to figure, we assume that neurons denoted by “J” that are presynaptic to neuron “I” interact through synapses whose efficacies are described by a weight “ w_{ji} ”.
- An action potential fired along the axon of neuron “J” evokes a post synaptic potential (PSP) in the dendrite of neuron “i” at a point. Where neuron “J” synapses with neuron “i”.
- The basic idea behind the spike response model is to represent each PSP by a kernel function & super position various such functions appropriately depending upon the firing times & physical locations of presynaptic neurons.
- The rest induced action potential is modeled as another kernel function.
- For super position procedure, we need a record of the firing times of a presynaptic neuron “J” which is denoted by “ t_j^k ”, where “k” indexes the times at which the neuron fired.
- Then the set of firing times of neuron “J” is denoted by

$$T_j = \{t_j^k, 1 \leq k \leq n\} \quad (1)$$
 Where t_j^n is the most recent firing time of the neuron
- A neuron fires when its cell potential, which we denoted by $V_j(t)$, equals the threshold V_θ .

$$T_j = \{t_j^k, 1 \leq k \leq n\}$$

$$= \left\{ \frac{t}{V_j}(t) = V_\theta \right\}$$
- Similar expressions can be written for the post synaptic neuron “I”.

Characteristics of ANNS: - The characteristics of Artificial Neural Networks are as follows: -

1. Mapping capabilities
2. Adaptive learning
3. Generalisability (or) Generalization
4. Robustness & fault tolerance
5. Parallel processing
6. Associative recall

The above listed characteristics are explained as follows.

1. Mapping capabilities: - neural networks have the capability of mapping.
2. Adaptive learning: - neural networks have a built in capability to adjust (or) adapt their synaptic weight changes to their surrounding environments.

3. Generalisability (or) Generalization: - Prediction of new outcomes based on the past trends is known as Generalization.
 4. Robustness & fault tolerance: Neural networks are robust and fault to learn in nature
 5. Parallel processing: - Neural networks can process the information at high speed in parallel and distributed fashion.
 6. Associative recall: - Associative recall can be considered as a natural capability of neural networks due to the inter connectivity and reinforcing structure of connected neuron leading to the concept called relative memory.
- Mc Culloch – Pitt’s model: - The first mathematical, artificial model for biological neurons was invented by Mc Culloch – Pitts in 1943.
Mc Culloch – Pitts neuron model use simple binary threshold functions for computation. The model diagram of Mc Culloch – Pitts model is as follows.

$x_i = x_1, x_2, \dots, x_n$ (where $i = 1, 2, \dots, n$) are the i/p that takes the value 0 (or) 1, depending on the presence (or) absence of i/p impulse at an instant k .

- “0” is the o/p signal of neuron. In this model the o/p of neuron is “1”. If induced local field of neuron is non-negative otherwise the value is “0”. This can be considered as all-or-none property of the Mc Culloch – Pitts model.
 $w_i = w_1, w_2, \dots, w_n$ (Where $i = 1, 2, 3, \dots, n$) are the weights of the networks.
- The rule of firing in this model is defined as follows:

$$o^{k+1} = \begin{cases} 1, & \text{if } \sum_{i=1}^n w_i x_i^k > T \\ 0, & \text{if } \sum_{i=1}^n w_i x_i^k < T \end{cases}$$

Where k denotes the discrete time instants ranging from $k=0, 1, 2, 3, \dots, n$.

w_i is the multiplicative weight associated with the i^{th} input.

If $w_i = +1$ then it is for excitatory synapses and

If $w_i = -1$ then it is for inhibitory synapses.

T denotes the threshold value of neuron, which has panoramic range of applications of applications in different domains.

- Potential applications of ANN: - The breadth of the neural networks panoramic range of applications in different domains.

The potential applications of neural networks are as follows:

1. Image processing / pattern recognition, wise recognition
2. Force casting / Risk assessment
3. Process modeling and control system
4. Constraint satisfaction / optimization
5. Portfolio management
6. Medical diagnosis
7. Intelligent searching
8. Quality control
9. Function approximation
10. Fraud detection
11. Target recognition
12. Credit rating
13. Target marketing, signature analysis
14. Machine diagnostics etc.

Unit 2

➤ Artificial neuron model: - A neuron is an information processing unit that is fundamental to the operation of a neuron network. The block diagram below shows the model of a neuron which forms the basis for designing artificial neural networks.

➤ The 3 basic elements of the neuronal model are

1. Synapses (or) connecting links
2. Adder
3. Activation function

1. A set of Synapses (or) connecting links each of which is characterized by a weight (or) strength of its own specifically, a signal x_j at the i/p of synapse J connected to neuron K is multiplied by the synaptic weight w_{KJ} .
2. An adder for summing the i/p signals, weights by the respective synapses of the neuron, the operations described here constitute a linear combiner.
3. An activation function for limiting the amplitude of the o/p of a neuron. The activation function is also referred to as a squashing function in that it squashes the permissible amplitude range of the o/p signal to some finite value.

➤ Typically the normalized amplitude range of the o/p of a neuron is written as the closed unit interval $[0, 1]$ (or) alternatively $[-1, 1]$.

➤ The neuronal model also includes an externally applied bias b_k . The bias b_k has the effect of increasing (or) decreasing the net i/p of the activation function, depending on whether it is positive (or) negative respectively.

➤ In mathematical terms, a neuron "k" may be described by the following pair of equations.

$$U_k = w_{k_1x_1} + w_{k_2x_2} + \dots + w_{k_mx_m} \quad (\text{or}) \quad U_k = \sum_{j=1}^m w_{k_jx_j} \quad \square (1)$$

$$\text{And } y_k = \phi(U_k + b_k) \quad \square (2)$$

Where $x_1, x_2, \dots, x_m =$ i/p signals.

$w_{k_1}, w_{k_2}, \dots, w_{k_m} =$ synaptic weights of neuron k.

U_k = linear combiner o/p due to the i/p signals

b_k = bias, $\phi(\cdot)$ = activation function, y_k = o/p signals of neuron.

- The use of b_k has the effect of applying an affine transformation to the o/p U_k of the linear combiner, as given by

$$V_k = U_k + b_k \quad (3)$$

- Thus the graph of induced local field (or) activation potential V_k versus U_k , no longer passes through the origin.

- The bias b_k is an external parameter of artificial neuron k. we may account for its presence as in equation (2) equivalently in to (3) can be as follows.

$$V_k = \sum_{j=0}^m w_{kj}x_j \quad (4)$$

$$\text{And } y_k = \phi(V_k) \quad (5)$$

- In equation (4), we have added a new synapse its i/p is $x_0 = +1$
And its weight is $w_{k_0} = b_k$.
- We may therefore reformulate the model of neuron k as below

- Types of activation function: - The activation function denoted by $\phi(v)$ defines the o/p of a neuron in terms of the induced local field "v". There are 2 basic types of activation functions.

1. Threshold function
2. Piece wise - linear function
3. Sigmoid function

1. Threshold function: - For this type of activation function described in figure we have.

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

This form of a threshold function is commonly referred to as a Heaviside function correspondingly the o/p of neuron k employing such a threshold function is expressed as

$$y_k = \begin{cases} 1, & \text{if } v_k \geq 0 \\ 0, & \text{if } v_k < 0 \end{cases} \quad (2)$$

Where v_k is the induced local field of the neuron.

$$\text{That is } V_k = \sum_{j=1}^m w_{k,j}x_j + b_k \quad (3)$$

Such a neuron is referred to as the Mc Culloch - Pitts model. In this model, the o/p of a neuron takes on the value of "1". If the induced local field of that neuron is non-negative and "0" otherwise.

2. Piece wise - linear function: -

For the piece wise – linear function described in figure we have,

$$\phi(v) = \begin{cases} 1, & v \geq 1/2 \\ v, & 1/2 > v > -1/2 \\ 0, & v \leq -1/2 \end{cases} \quad (4)$$

Where the amplification factor inside the linear region of operation is assumed to be unity. The following two situations may be viewed as special forms of the piece-wise linear function.

- ❖ A linear combiner arises if the linear region of operation is maintained without running into saturation.
- ❖ The piece-wise linear function reduces to a threshold function. If the amplification factor of the linear region is made infinitely large.

3. Sigmoid function: - The sigmoid function, whose graph is “s” shaped is by far the most common form of activation function used in the construction of artificial neural networks. It is defined as a strictly increasing function that exhibits a graceful balance b/w linear and non-linear behavior. An example of the sigmoid function is the logistic function, defined by

$$\phi(v) = \frac{1}{1 + \exp(-av)} \quad (5)$$

Where “a” is the slope parameter of the sigmoid function. By varying the parameter “a”. We obtain sigmoid functions of different slopes as in figure in fact, the slope at the origin equals $\frac{a}{4}$.

The activation functions defined in equations (1), (4) & (5) range from “0 to +1”. It is sometimes desired to have the activation function range from “-1 to +1”. In that case the function can be defined as

$$\phi(v) = \begin{cases} 1, & \text{if } v > 0 \\ 0, & \text{if } v = 0 \\ -1, & \text{if } v < 0 \end{cases} \quad (6)$$

Which is commonly referred to as the sigmoid function for the corresponding form of a sigmoid function. We may use hyperbolic tangent function defined by

$$\phi(v) = \tanh(v) \quad (7)$$

ANN Architecture: -

The 3 different neural network architecture: -

1. Single layer feed forward network
2. Multi layer feed forward network
3. Recurrent network

1. Single layer feed forward network: - In layered neural network, neurons are organized in the form of layer. This kind of network contains 2 layers (a) Input layer (b) Output layer. The i/p layer nodes collect i/p signals and o/p signals are received by o/p layer nodes.

2. Multi layer feed forward network: - This type of network consists of multiple layer. This architecture distinguishes itself by the presence of one (or) more hidden layers. The computation nodes of hidden layers are called hidden neurons (or) hidden units. A feed forward network of "m" source nodes h_1 neurons in the first hidden layer, h_2 neurons in the second hidden layer and Q o/p neurons in the o/p layer is referred to as "m- h_1 - h_2 -Q" network.
The figure below depicts a multi layer feed forward network.

3. Recurrent networks: - A recurrent neural network distinguishes itself from feed forward neural network in that it has at least one feedback loop.
The figure depicts a recurrent neural network.

- Classification taxonomy of ANN-Connectivity: - The table below illustrates taxonomy of neural networks system in accordance with learning methods and architecture types.

| | Architecture type | | |
|------------------|---------------------------|--------------------------|--------------------------|
| | Single layer feed forward | Multi layer feed forward | Recurrent neural network |
| Gradient Descent | AD Aline | CCN | RNN |
| Hebbian | Hopfield perception | MLFF | RBF |
| Competitive | AM Hopfield | Neocongnitron | BAM, BSB |
| stochastic | LVQ, SOFM | - | Hopfield ART |
| | - | - | Coltzmann machine |
| | | | Couchy machine |

- Neural Dynamic (Activation & Synaptic): - An artificial neuron network structure is useless until the rules governing the changes of activation values and connection weight values are specified. These rules are specified in the equations of activation and synaptic dynamic which governs the behaviors of structure of the network to perform the desired task.

In a neural network, the activation dynamics is related to the fluctuations at the neuronal level where these fluctuations take place in intervals of order of mill seconds where as in synaptic level dynamics the changes in the synaptic weights takes place in intervals of the order of few seconds. Hence activation level dynamics is faster than synaptic level dynamics. Therefore synaptic weights are assumed to be the constant because they not change significantly.

- Learning strategy (Supervised, Unsupervised, Reinforcement): - learning methods in artificial neural networks are classified into 3 fundamental types
1. Supervised learning
 2. Unsupervised learning
 3. Reinforced learning

1. Supervised learning: -

- ❖ In supervised learning, while training the network every i/p pattern is linked with an o/p pattern. Hence o/p pattern is considered as target (or) desired pattern.

- ❖ During learning process, a teacher is needed in order to compare the expected o/p with the actual o/p error determination.
 - ❖ In supervised systems, learning can be carried out in the form of difference equation which is desired to work with global information.
2. Unsupervised learning: -
- ❖ In unsupervised learning, while training the network desired o/p (or) target o/p is not distributed across the network.
 - ❖ During learning process no teacher is required to give the desired patterns. So, the system being to learn itself by recognizing and adjusting to different structures in the i/p patterns.
 - ❖ In unsupervised learning system, learning is carried out in the form of differential equations which are designed to work with the available information in local synapse.
3. Reinforced learning: - Reinforced (or) Reinforcement learning is a behavioral learning problem. In this training is not provided to the learning, instead the learner interacts with the environment continuously to perform the learning of i/p & o/p mapping. The following fig illustrates the diagram of one type of learning system.

- Learning Rules: - Hebbian learning rule: - In Hebbian learning rule, learning signal “r” is equivalent to neurons o/p learning signal “r” is a function of i/p “x” and weight vector “w_i”.

$$r = f(w_i t_x)$$

$$\text{We have } \Delta w_i = cf(w_i t_x) x \text{ and } \Delta w_{ij} = cf(w_i t_x) x_j$$

Where Δw_{ij} represents increment weight vector and laying this increment the individual weight vector Δw_{ij} is adjusted “c” corresponds to the number called “learning constant” that determines the rate of learning single weight adjustment Δw_{ij} can be written as

$$\Delta w_{ij} = c o_i x_j \quad \text{For } j=1, 2, \dots, n$$

The o/p is made stronger for each i/p presented.

- Delta Learning rule: - In delta learning rule the learning signal “r” is called delta which is defined as

$$r = [d_i - f(w_i t_x) f'(w_i t_x)]$$

d_i □ Desired response at o/p unit “i”

$f'(w_i t_x)$ □ It corresponds to the derivative of action function $f(w_i t_x)$.

This rule is applicable for only continuous activation functions and in the supervised training mode.

- Delta rule can be easily derived from the squared error condition. The gradient vector "E" is calculated with respect to w_1 of the square error b/w O_i and d_i . By differentiating the gradient vector "E". We obtain error gradient vector ΔE .

$$E = \frac{1}{2} (d_i - o_i)^2 \quad (1)$$

$$\text{As } o_i = f(w_i t_x)$$

$$\text{Equation (1) can be written as } E = \frac{1}{2} (d_i - o_i)^2 f'(w_i t_x) \quad (2)$$

Gradient vector component for $J=1, 2, \dots, n$ are as follows

$$\frac{\partial E}{\partial w_{ij}} = - (d_i - o_i) f'(w_i t_x) x_j$$

As minimization of error needs weight adjustment in the negative gradient direction. So we get

$$\Delta w_i = -\nabla E \quad (3)$$

From equation (2) and (3) we get

$$\Delta w_i = \nabla E (d_i - o_i) f'(net_i) x_j \quad (4)$$

Equation (4) for the individual weight adjustment for $J=1, 2, \dots, n$ as follows.

$$\Delta w_{ij} = (d_i - o_i) f'(net_i) x_j \quad (5)$$

Based on the minimization of squared error weight adjustment in equation (4) & (5) is calculated. By making use of general learning rule and learning signal defined as,

$$\Delta w_i(t) = C_r (w_i(t), x(t), d_i(t)) x(t)$$

And $r = [d_i - f(w_i t_x)] f'(w_i t_x)$ respectively,

The weight adjustments becomes

$$\Delta w_i = c(d_i - o_i) f'(net_i) x$$

- Window-Hoff learning rule: -

In window-Hoff learning rule the learning signal "r" is defined as

$$r = d_i - w_i^t x$$

This rule is independent of activation function and is valid for supervised training networks.

The weight vector increment Δw_i and single weight adjustment Δw_{ij} as follows.

$$\Delta w_i = c(d_i - w_i)x$$

$$\Delta w_{ij} = c(d_i - w_i^t x)x_j$$

Where “c” is positive constant learning constant.

This rule is a special case of delta learning rule taking $f^I(w_i^t x) = w_i^t x$ in delta learning signal “r”. Shows below in equation (1), we get $f^I(w_i^t x) = 1$ and “r” in equation (1)

$$\text{reduces to } r = d_i - w_i^t x$$

$$r = d_i - f(w_i^t x) f^I(w_i^t) \quad (1)$$

This window Hoff learning rule is occasionally known as “Least means square (LMS) learning” rule.

➤ TYPES OF APPLICATIONS: -

Applications of artificial neural networks: The following are the application domains of artificial neural networks.

1. Pattern reorganization: - Neural networks have been used successfully in large no. of tasks such as follows:
 - a) Recognizing printed.
 - b) Identification of visual image
 - c) Reorganization of speech
2. Constraint satisfaction: - This includes problems which must fulfill the conditions and obtain an optimum solution.
 - a) Manufacturing scheduling
 - b) Finding the shortest path for set of cities.
3. Forecasting and risk assessment: - There are many problems in which future events must be predicted on the basis of past history.
4. Control systems: - By finding applications in control system, neural networks have attained business roots.
5. Vector quantization: - Vector quantization is the process of dividing space into several connected regions.

Unit 3Algorithm

SDPTA: - S-Category discrete perception training algorithm.

Input: - N Training pairs and the augmented i/p vectors.

Output: - Training step and weights.

Parameters: d, y, p, n, k, E, c, w, t, o

k is the training step ; E is the error value; y is the i/p vector; n is the dimension of the space;

c is the correction increment; d is the desired i/p; o is the actual o/p;

w is the weight vector and of order $(n+1) \times 1$; t is the step counter in the training cycle;

p is the pattern vector

The augmented i/p vectors are $y_i \begin{bmatrix} P_j \\ i \end{bmatrix}$

Where $J = 1, 2 \dots N$

The training pairs are $\{P_1, d_1, P_2, d_2 \dots P_N, d_N\}$

Where P_j is $(n \times 1)$ and d_j is (1×1) and $J = 1, 2, 3 \dots N$

1. Choosing "C": - "C" is a constant and a +ve integer
2. Initialization: - "W" is Initialized with small random values; "k" is Initialized to 1; "t" is Initialized to 1; E is Initialized to 0.
3. Computing the output: - i/p is taken and o/p is calculated as $y = y_t, d = d_t, o = \text{sgn}(w^t y)$
4. Updating weight: - $w = w + \frac{1}{2} c (d - o) y$
5. Computing the cycle error: - $E = w + \frac{1}{2} (d - o)^2 + E$
6. Condition Checking: - If $(t < N)$
 - { $t = t + 1$
 - $k = k + 1$
 - go to step 3
 - }
 - else
 - go to step 7

7. Output: - if (

Algorithm: - (SCPTA): -

Input: - N training pairs and the augmented i/p vectors.

Output: - Training step and weights.

Parameters: - η , α , E, y, d, t, o, w, s.

η is the learning coefficient; α is the steepness coefficient; E is the error value; d is the desired output;

o is the actual o/p; t is the step counter in the training cycle; w is the weight vector; y is the augmented i/p vector; s is the signal for exciting the neuron.

The augmented i/p vectors are

$$y_j = \begin{bmatrix} P_j \\ 1 \end{bmatrix} \text{ Where } j = 1, 2 \dots N$$

The training pairs are $\{(P_1, d_1, P_2, d_2 \dots P_N, d_N)\}$

Where P_j is $(n \times 1)$ and d_j is (1×1)

Being

1. Choosing η , α & E_{\max} : - A value for η is chosen such that $\eta > 0$, $\alpha = 1$ and $E_{\max} > 0$ is chosen.
2. Initialization: - w is initialized to small random $k = 1$, $t = 1$, $E = 0$ (zero)
3. Computing the o/p: - the augmented input vector is taken and the actual o/p is evaluated as

$$y = y_t, d = d_t, o = f(w^t y)$$
4. Updating weights: - $w = w + 0.5 \eta (d - o) (1 - o^2)y$ (or) $w = w + \frac{1}{2} \eta (d - o) (1 - o^2)y$
5. Computing the cycle error: - $E = \frac{1}{2} (d - o)^2 + E$
6. Condition checking: - $\begin{cases} \text{if } (t < N) \\ t = t + 1 \end{cases}$ go to step 7
 If $(E < E_{\max})$

```

display weights and k
else
if (E > Emax !! E = Emax)
{
E = 0
t = 1
go to step - 3
}
and

```

- Multi category single layer perception network: - Training (or) correcting the errors in multi category signal layer perception need that the following assumption is made.
- The classes are linearly pair wise separable. The assumption may be restated as
- There are R discriminate function such that $D_j(P) > D_k(P)$
Where $J, k = 1, 2, 3 \dots R$ and $J \neq k$
- The augmented weight vector w_a for training. The R category classifier is given by
 $w_a \geq [w_{a1}, w_{a2}, \dots w_{a,n+1}]^t$ and
 $w_1^1 = w_1$
 $w_2^1 = w_2$ } □ (1)
 $w_R^1 = w_R$

The weights are adjusted only when $w_j^t y$ greater than the remaining R-1 discriminate function similarly.

If $w_j^t y \leq w_s^t y$

For some "s", then the weight vectors are

$$\begin{aligned}
 w_j^1 &= w_j + c_y \\
 w_s^1 &= w_s + c_y \} \square (2) \quad \text{for } e = 1, 2, 3 \dots R \text{ and } e \neq s, J \\
 w_e^1 &= w_e
 \end{aligned}$$

Equation (2) may be written as

$$\begin{aligned}
 w_{jh}^1 &= w_{jh} + c_{yh} \quad h = 1, 2 \dots n+1 \\
 w_{sh}^1 &= w_{sh} + c_{yh} \quad h = 1, 2 \dots n+1 \} \square (3) \\
 w_{eh}^1 &= w_{eh} \quad e = 1, 2 \dots R \\
 &\quad e = J, m, h = 1, 2 \dots n+1
 \end{aligned}$$

- The weights are adjusted depending on whether the weight value is too large (or) too small in equation (3), the equation for w_{jh}^1 & w_{sh}^1 use this rule. The term “ c_{yh} ” is added if the J^{th} o/p is too small similarly, “ c_{yh} ” is subtracted when the J^{th} o/p is too large.

The multi category single layer perceptron network is depicted in figure (1).

- Network is given a fixed value of “+1”. However, n multi category single layer perceptrons, the value is “-1”.
- Any ways, the value for y_{n+1} is irrelevant as during the training process, weights are iteratively chosen.

The equation for “s” in f (3) is

$$S = w_p^t - w_{n+1} \quad \square (4)$$

- The term w_{n+1} is the bias (or) threshold value can be denoted “T” and the neurons o/p in term of “T” may be

$$f(s) - \begin{cases} > 0 \text{ for } w_p^t > T \\ < 0 \text{ for } w_p^t < T \end{cases} \quad \square (5)$$

The activation function with $T > 0$ is depicted in below fig.

If the weighted sum is more than “T” only then the neurons exerted else inhibited.

- R – Category Discrete Preception Training Algorithm (RDPTA): -

Algorithm for RDPTA: -

Input: - N training pairs and the augmented i/p vector

Output: - Training step and weights.

Parameters: - d, y, t, k, o, w, E, R, C

d is the desired o/p

y is the augmented i/p vector
 t is the step counter in the training cycle
 k is the training step
 o is the actual o/p
 w is the weight vector of order $R \times (n+1)$
 E is the error value.
 R is the no. of categories
 C is the correction increment

- The augmented i/vectors are $y_j = \begin{pmatrix} P_j \\ -1 \end{pmatrix}$ where $J = 1, 2 \dots N$.
- The training pairs are $\{P_1, d_1, P_2, d_2 \dots P_N, d_N\}$; where P_j is $(n \times 1)$ and d_j is (1×1) .
 1. Choosing "C": - A position integer value, which is a constant is chosen for "C"
 2. Initialization: - w is initialized with small random values $K = 1, t = 1, E = 0$.
 3. Computing the output: - The augmented i/p vectors are taken and the o/p is calculated as $y = y_z, d = d_z$
 $O_j = s_{gn}(w_j^t \cdot y)$
 Where $J = 1, 2 \dots R$ & w_j indicates the J^{th} row of the weight vector w .
 4. Updating weights: - $w_j = w_j + o \cdot s_c(d_j - o_j)y$ where $J = 1, 2 \dots R$.
 5. Computing the cycle error: - $E = o \cdot s(d_j - o_j)^2 + E$; where $J = 1, 2 \dots R$.
 6. Condition checking: - if $(t < N)$
 - { $t = t + 1$;
 - $k = k + 1$;
 - go to step -3;
 - }
 - else
 - go to step -7
 7. Output: - End of the cycle provided
 - If $(E == 0)$
 - Display weight Q_k
 - else
 - if $(E > 0)$
 - {
 - $E = 0$
 - $t = 1$
 - go to step -3
 - }
 - end.

Preceptron convergence theorem

This theorem states that the preceptor learning law converges to a final set of weight values in a finite no. of steps, if the classes are linearly separable (or) if the given classification problem is representable.

Proof: - The proof of this theorem is as follows: - Let a & w be the augmented i/p and weight vector respectively. Assuming that there exists a section w^* for the classification problem, we have to show that w^* can be approached in a finite no. of steps, starting from some initial random weight values. We know that the solution w^* satisfies the following in equality.

$$w^{*T} a > \alpha > 0$$

For each $a \in A_1$ \square (1) where $\alpha = \min (w^{*T} a)$, $a \in A_1$,

The weight vector is updated it $w^T(m)a \leq 0$, for $a \in A_1$, that is

$$w(m+1) = w(m) + a(m)$$

For $a(m) = a \in A_1$ \square (2)

Where $a(m)$ is used to denote the i/p vector at step m .

If we start with $w(0)=0$, where 0 is all zero column vector, then

$$w^{*T} w(m) = \sum_{i=0}^{m-1} a(i) \square (3)$$

Multiplying both sides of (3) by w^{*T} , we get

$$w^{*T} w(m) = \sum_{i=0}^{m-1} w^{*T} a(i) > m \alpha \square (4)$$

Since $w^{*T} a(i) > \alpha$ according to equation (1) using the Cauchy-Schwartz in equality

$$\|w^{*T}\|^2 \cdot \|w(m)\|^2 \geq [w^{*T} w(m)]^2 \square (5)$$

We get from equation (4)

$$\frac{\|w(m)\|^2 >^2 m^2 \alpha^2}{\|w^{*T}\|^2} \square (6)$$

We also have from equation (2)

$$\begin{aligned} \|w(m+1)\|^2 &= [w(m) + a(m)]^T [w(m) + a(m)] \\ &= \|w(m)\|^2 + 2 w^T(m) a(m) + \|a(m)\|^2 \leq \|w(m)\|^2 + \|a(m)\|^2 \square (7) \end{aligned}$$

Since for learning $w^T(m).a(m) \leq 0$. When $a(m) \in A_1$

Therefore, starting from $w(0)=0$, use we get from equation (7)

$$\|w(m)\|^2 \leq \sum_{i=0}^{m-1} \|a(i)\|^2 \leq \beta m \square (8)$$

Combining equation (6) and (8), we obtain the optimum value of m by solving.

$$\frac{m^2 \alpha^2}{\|w^{*T}\|^2} = \beta m \square (9)$$

Or

$$m = \frac{\beta}{\alpha^2} \|w^{*T}\|^2 = \frac{\beta}{\alpha^2} \|w^*\|^2 \square (10)$$

Since β is positive equation 10 shows that the optimum weight value can be approached in a finite number of steps using the preceptron learning law.

□ Limitations of perceptron learning rule: -

Perceptron cannot handle in particular, tasks which are not linearly separable.

Set of points in two dimensional spaces are linearly separable. If the sets can be separated by a straight line figure below illustrates linearly separable patterns and non-linearly separable patterns.

The perceptron cannot find weights for classification types of problems that are not linearly separable.

An example is the XOR problem.

□ XOR problem: - XOR is a logical operation as described by its truth table presented in table below.

The i/p's as odd parity (or) even parity. Here odd parity means odd number of 1 bit in the o/p.

This is impossible, since as is evident from fig below perceptron is unable to find a line separating even parity i/p patterns from the odd parity i/p patterns.

□ Reinforcement learning: - The reinforcement learning can be viewed as a credit assignment problem depending on the reinforcement signal; the credit (or) blame for the overall outcome is assigned to different units (or) weights of the network. The different types of credit assignment are

1. Structural credit assignment.
2. Temporal credit assignment.
3. Fixed credit assignment.
4. Probabilistic credit assignment.
 - In structural credit assignment, the credit (or) blame is assignment to internal structures of the system whose actions generated the outcome. On the other hand, if the is assigned to outcomes of series of actions based on the reinforcement signal received for the overall outcome, it is called temporal credit assignment.
 - The combined temporal and structural credit assignment problem is also relevant in situations involving temporally extended distributed learning system.
 - The reinforcement signal can also be viewed as a feedback from the environment which provides i/p to the network and observes the o/p of the network.
 - If the reinforcement signal from the environment is the same for a given i/p – o/p pair & if it does not change with time, it is called a fixed credit assignment problem.
 - On the other hand, if the given i/p – o/p pair determines only the probability of positive reinforcement then the network can be viewed as operating in a stochastic environment. In such a case, it is called probabilistic credit assignment. Here the probabilities are assumed stationary.

Unit 4

□ Generalized delta rule: -

3 layers namely i/p layer, hidden layer and o/p layer. Let these layer consist of p, m and n neurons respectively and I, h, j are their respective subscripts.

The following equation (1) is known as generalized delta rule. In according with this rule, weights are updated and corrected incrementally.

$$\Delta w_{ih}^l = \delta_h x_i^l + \alpha \Delta w_{ih}^{l-1} \quad (1)$$

Where “ w_{ih} ” denotes the synaptic weights between i/p & hidden layers & the terms α is a +ve number called “momentum constant”. This term is used to accelerate the process.

In accordance with the same rule, the weights b/w hidden & o/p layer are updated as

$$\Delta w_{hj}^l = \delta_j \delta(z_h)^l + \alpha \Delta w_{hj}^{l-1}$$

The i/p's are applied to the network one by one and the weight updating is done until the total error is computed, from the desired o/p & the network o/p.

□ Summary of back propagation training algorithm: -

Where z_i is (I x 1), d_i is (k x 1) and $i = 1, 2 \dots P$

Note that I^{th} component of each z_i is of value “-1”. Since i/p vectors have been augmented.

Since size

(J - 1) of the hidden layer having o/p's “y” is selected.

Note that J^{th} component of y is of value “-1”. Since hidden layer o/p's have also been augmented, y is (J x 1) and 0 is (K x 1).

Step 1: - $\epsilon > 0$, E_{\max} chosen weights W & V are initialized at small random values; w is $(K \times J)$, V is $(J \times I)$

$$Q \leq 1, P \leq 1, E \leq 0$$

Step 2: -

and the layer o/p's computed using the formula

$$f(\text{net}) \triangleq \frac{2}{1 + e^{-\text{net}}} - 1$$

$$Z \leq z_p, d \leq d_p$$

$$y_j \leq f(u_j^t z) \text{ for } j = 1, 2 \dots J.$$

Where u_j is a column vector is the J^{th} row of V , and

$$\theta_k \leq f(w_k^t y) \text{ for } k = 1, 2 \dots K.$$

Where w_h a column vector is the K^{th} row of W

Step 3: - Error value is computed

$$E \leq \frac{1}{2} (d_k - O_k)^2 + E, \text{ for } k = 1, 2 \dots K$$

Step 4: - Error signal vector δ_o & δ_y of both layers are computed. Vector δ_o is $(K \times 1)$, δ_y is $(J \times 1)$

The error signal terms of the o/p layer in this step are

$$\delta_{ok} = (d_k - O_k)(1 - O_k^2) \text{ for } k = 1, 2 \dots K$$

The error signal terms of the hidden layer in this step are

$$\delta_{yj} = \frac{1}{2} (1 - y_j^2) \sum_{k=1}^K \delta_{ok} w_{kj}, \text{ for } j = 1, 2 \dots J$$

Step 5: - o/p layer weights are adjusted

$$w_{kj} \leq w_{kj} \leq \delta_{ok} y_j \text{ for } j = 1, 2 \dots J \text{ \& } k = 1, 2 \dots K$$

Step 6: - Hidden layer weights are adjusted

$$U_{ij} \leq U_{ij} \leq \delta_{yj} z_i \text{ for } j = 1, 2 \dots J \text{ \& } i = 1, 2 \dots I$$

Step 7: - If $P < P$ then $P \leq P + 1$, $q \leq q + 1$ & go to step 2 otherwise go to step 8

Step 8: - The training cycle is completed for $E < E_{\max}$ terminated the training session.

o/p weights w , v , q & E

If $E > E_{\max}$, then $E \leq 0$, $P \leq 1$ and initiate the new training cycle by going to step 2

Kolmogorov Theorem: - This theorem states that any continuous function is represented by a set of non-linear and continuous increasing functions containing exactly one variable.

$$f(x_1, x_2 \dots x_n) \in \mathbb{R},$$

$$x_i \in [0, 1] \text{ for } i = 1, 2 \dots n$$

It states that any continuous function defined on an n -dimensional cube can be implemented exactly by a three layer feed forward neural network. The network should have "m" elements in the i/p layer $2m+1$ processing elements in the o/p layer the processing element of hidden layer makes useful activation function shown below.

Processing function both & is independent of continuous function f

The constant ϵ is a +ve rational number ($\epsilon > 0$) & x_i corresponds to I^{th} i/p.

The processing elements of the o/p layer make use of the following activation function.

$$y_j = \sum_{p=1}^{2m+1} g_J(h_p)$$

The choice of real continuous function "g" depends of f & ϵ .

This theorem states that there must exist a 3 layer feed forward network but it does not show how to find it.

In back propagation algorithm, the selection of learning coefficient is a tricky task.

The formula to choose the learning coefficient is as follows.

$$= \frac{1.5}{P_1^2 + P_2^2 + \dots P_m^2}$$

Where P_1 = no. of patterns of type 1

M = no. of different pattern types.

Learning coefficient that generates rapid learning depends on the number and type of i/p patterns. To compute patterns type, target o/p is utilized. If each pattern is a different type, then the value of learning coefficient obtained is high.

A large value of learning coefficient greater than 0.5 will lead to rapid training but the weights may oscillate and will overshoot from its ideal weight vector position whereas low values imply slow learning which causes the system to converge slowly but with little oscillation. To permit fast learning, the choice of learning coefficient must be high but with no oscillations.

Unit 6

Classical set theory: - A set is a well defined collection of objects. Here, well defined means the object either belongs to (or) does not belong to the set.

To indicate that an individual object "x" is a member (or) element of a set A, we write.

$$x \in A.$$

Whenever "x" is not an element of set A, we write $x \notin A$.

To represent sets, we have basic methods, as follows.

1. List method $A = \{a, b, c\}$
2. Rule method $C = \{x / P(x)\}$

The set "c" is composed of elements x. every x has prototype.

Relations on Sets: - If A & B are sets then \exists following relations between sets.

1. A C B: - $x \in A \Rightarrow x \in B$ i.e.

If every element of set A is also a member of set B, then A is called subset of B.

2. A = B: - If ACB & BCA then A & B contain same numbers. They are called equal sets.
3. A C B: - If ACB & $A \neq B$ then B contains at least one element that is not member of A. A is called "Proper subset" of B

All the possible subsets of a given set x is called power set of x, denoted by

$$P(x) = \{A / A \subset X\}$$

$$|P(x)| = 2^n \text{ where } n = |X|.$$

4. Null set: - A set consisting of no elements is called null set. It is denoted by \emptyset .

The basic operations on crisp sets are

- a. Union (U)
- b. Intersection (\cap)
- c. Complement (C)
- d. Difference (-)

- a. Union (U): - Let P & Q be 2 sets. The union b/w 2 sets, denoted $P \cup Q$, Represents all the elements that reside in set P, the set Q (or) both sets P & Q.

It is defined as

$$P \cup Q = \{x / x \in P \text{ (or) } x \in Q\}$$

$$\text{Example: } P = \{1, 2\}, Q = \{a, b\}, P \cup Q = \{1, 2, a, b\}$$

Unit 7

Fuzzification: - Fuzzification is the process of transforming the crisp (or) classical sets in to fuzzy sets. These sets are required to be converted in to fuzzy sets, as they no more carry the tag of “crisp sets”

The voltage readings are not accurate rather approximate. The membership function representing such imprecision's in the voltage reading is depicted in the below figure:

Fuzzification is not a compulsory step for using crisp data in fuzzy systems. However, it is recommended to fuzzily the crisp data. The difference in crisp and fuzzy readings of the voltmeter is depicted in the below figure:

If figure 2, the intersection at 0.3 indicates that both the crisp as well as fuzzy readings have agreed at a membership value of 0.3, the same agreement exists in figure3.

Membership value assignment: -

Methods to generate membership functions:- Many procedures are available to generate membership function of these, six procedures are the straight forward ones.

1. **Intuition:** - Intuition is the in both tendency to behave in a situation. It is enhanced with experience. Linguistic truth values, semantic and contextual knowledge of any issues are the basic elements of intuition.

For various