Unit-3

Hebb's Rule

If neuron 'i' is near enough to excite 'j' & repeatedly participati in its activ^n, the synaptic connec^n b/w that neurons is strengthened & neuron 'j' becomes more sensitive to stimuli from neuron 'i'.

It follows 2 rules:-

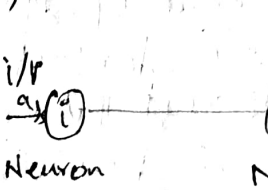
(i) If 2 neurons on either side of connec^n are activated synchronously, then the wt. of that connec^n is ↑sed.

(ii) _____ asynchronously _____ ↓sed.

→ If 2 neurons have no relaⁿship, then the wt. will not change.

→ If i/p of both nodes are ~~either~~ either both +ve or both -ve, then a +ve wt. exists b/w the nodes.

→ If i/p of a node is either +ve or -ve for other, a +ve wt. exists b/w the nodes.



x_i^k → i/p value to neuron 'i' at time 'k'

x_j^k → o/p value of neuron 'j' at time 'k'

w_{ji}^k → wt of connec^n at time 'k'

w_{ji}^{k+1} → _____ k+1

Δw_{ji}^k → ↑ment by which wt. of connec^n is strengthened.

α → +ve const. coeff. which determines learning rate.

Hebb's formula:

$$w_{ji}^{k+1} = w_{ji}^k + \Delta w_{ji}^k$$

where $\Delta w_{ji}^k = \alpha x_i^k x_j^k$.

Kohonen

Self Organising Maps (SOM)

→ SOM are named as 'self organising' bcoz no supervision is reqd.

→ SOM learns on their own through unsupervised competitive learning.

Defn :- SOM are neural n/ws that uses USL approach & trained its n/w through a competitive learning algo, to map multidimensional data into lower dimensional which allows easy interpretation of complex probs.

→ SOM has 2 layers : ↳ I/P Layer
↳ O/P Layer.

SOM Training Algo :

i) Initialisation : choose random values for the initial wts- w_{ij}.

ii) Sampling :- take a sample training i/p vector $x\{x_1, x_2, \ldots x_n\}$ from the i/p layer.

iii) Matching :- Find the winning neuron from the O/P layer that has wt vector closest to the i/p vector.

→ It can be calculated by taking the sq. of Euclidean distance for each output unit & finds that o/p unit that has min. Euclidean distance from the i/p vector.

for each $j = 1$ to m

$$D(j) = \sum_{i=1}^{n} \sum_{j=1}^{m} (x_i - w_{ij})^2$$

→ Phonetic Typewriter :- one of the earliest & well-known ~~technology~~ app^n of SOM which is set in the field of speech recognition.

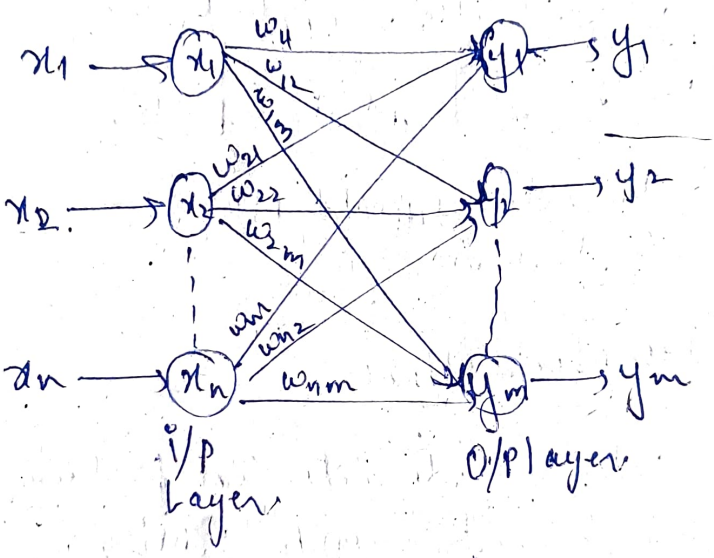
iii) New Wt- Calculation :- Find the new wt. b/w i/p vector sample & winning o/p unit.

$$w_{ij}(new) = w_{ij}(old) + \alpha (x_i - w_{ij}(old))$$

α → learning rate.

iv) Continuation :- Repeat step 2 to 4 until wt. updation is negligible.

Architecture of SOM :-

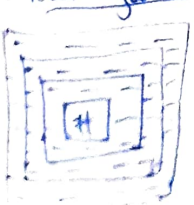


App^n of SOM :

→ Clustering of animals due to their features.
→ Analysis of financial stability
→ Fault diagnosis of plants
→ Atmospheric sciences.
→ Creation of well-composed heterogeneous teams.
→ Phonetic Typewriter.

Neighbor Topology in SOM : Winning unit is indicated by #.

① Rectangular Grid



Distance 0 grid = 8 nodes
1 = 16
2 = 24

② Hexagonal Grid

Distance 0 grid = 6 nodes
1 = 12
2 = 18

Hopfield Network:-

→ It consists of a set of neurons with o/p of each neuron is feedback to all other neurons.

→ Hopfield N/ws have been used in many app's of associative memory & many optimisation probs (Traveling salesman)

→ It is an auto associative, fully interconnected, single layer feedback n/w.

→ It is a symmetrically weighted n/w

$$w_{ij} = w_{ji} \quad , \quad w_{ii} = 0$$

→ 2 types:- Discrete Hopfield N/W
Continuous "

(i) Discrete Hopfield N/w:-

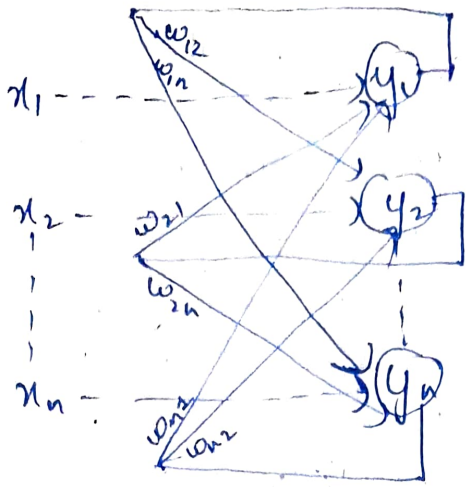
→ It operates in discrete line fashion.

→ This n/w has 2 types of values i/ps: ⓐ Binary (0,1)
ⓑ Bipolar (-1, +1)

→ It has symmetrical wts with no self connects. $w_{ij} = w_{ji}$, $w_{ii} = 0$

→ The architure ~~has~~ consist of 2 o/ps; one inverting & the other non-inverting.

→ The o/p from each processing element are feedback to the i/p of other processing element but not to itself.



$S(p) = \{ S_1\ S_2\ S_3\ S_4\ S_5 \}$
$\quad\quad [\ 0\ 1\ 1\ 0\ 1\ ]$

$p \Rightarrow$ no. of i/ps.

$S(p) \Rightarrow$ input vector

Hopfield n/w

→ for storing a set of binary pattern $S(p)$, where $p = 1$ to P, the wt. matrix is :

$$\omega_{ij} = \sum_{p=1}^{P} [2S_i(p) - 1][2S_j(p) - 1], \text{ where } i \neq j$$

→ for bipolar pattern $S(p)$,

$$\omega'_{ij} = \sum_{p=1}^{P} S_i(p) \cdot S_j(p), \text{ where } i \neq j$$

for self connec^n, $\omega_{ij} = 0$

(ii) Continuous Hopfield N/W

→ A discrete HN can be modified to continuous HN.

→ The nodes of this n/w have continuous o/p, rather than a 2 state o/p (b/w 0 & 1)

→ The continuous HN can be realised as an electric circuit, which uses non-linear amplifier & resistors.

Adaptive Resonance Theory (ART)

→ ART is open to new-learning without discarding the old info.
→ It is based on competition and uses USL model.
→ It is self organising.
→ ART N/w solves the 'stability - plasticity dilemma'
→ It implements a clustering algo.
I/P is presented to the n/w, algo checks whether it fits into any already stored cluster.
Yes → I/P is added
NO → new cluster is formed.

Types of ART architecture :-

1) ART 1 → To cluster binary i/p values.
2) ART 2 → ——— continuous ———.
3) Fuzzy ART → Augmentation of fuzzy logic & ART.
4) ART MAP → Supervised form of ART learning.

Advs :-

→ Exhibits stability (undisturbed by i/ps).
→ Ability to add new clusters.
→ Can be integrated with other technologies.
→ Has benefits over competitive learning.

Appn :-

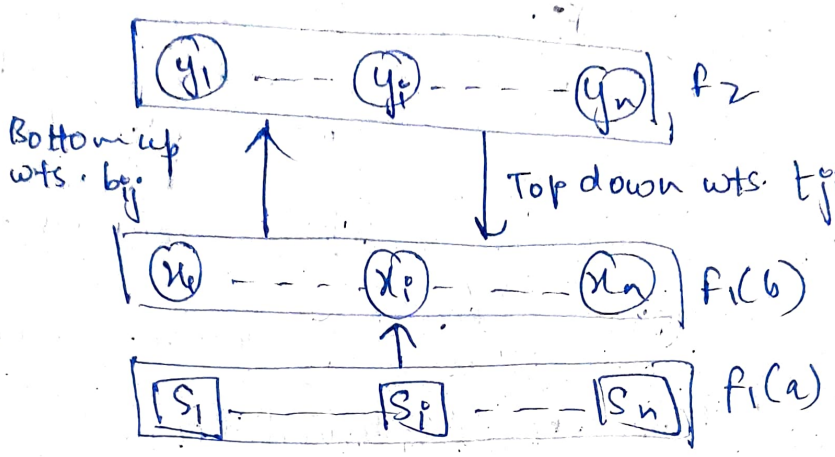
→ Target Recogn → Sign verifican
→ Medical diagnosis → Mobile Control Robot

Disadv.:- depends on learning rate & order of training data.

Architecture /

Phases: f_1: Comparison : short term memory layer.

f_2: Recognition : long term memory layer.



Boltzman Machines

⇒ It is a n/w of symmetrically connected, neurons like units that make stochastic (Random Probability Distribu^n) decision about whether to be ON or OFF.

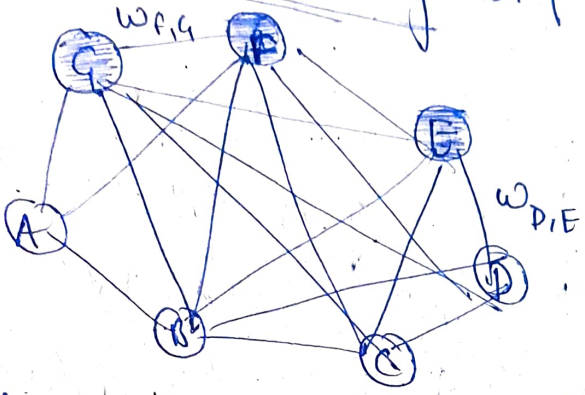
→ It is a part of USL.

→ We provide some i/p to the model, we let the model decide the relat^{n}ship b/w the features in the data.

→ Here, we don't provide the model with any o/p.

→ Undirected Model - Connec^n goes both the ways.
→ Non Deterministic Model → no outputs given
→ Purpose → to optimise the sol^n (Travelling salesman prob)

Graphical Representa^n of BTM



- A few wts labelled
- Each undirected edge represents dependency & is weighted with wt.

Here, there are 3 hidden units (colored) & 4 visible units (uncolored). ← neurons are not visible

i/p layers.

- Every node is connected to every other node.
- No o/p layer.
- The machine tries to find out the relat^ship b/w the i/ps using the features.

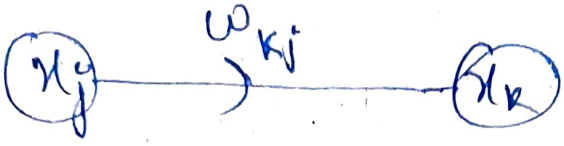
→ ~~Disadv~~: - learning algo. is very slow in n/ws with many layer of feature detectors, but it can be made much faster by learning one layer of feature detectors at a time.

Energy fn

→ BM is characterised by an energy fn.

$$E = -\frac{1}{2} \sum_j \sum_k w_{kj} x_k x_j$$

where $j \neq k$; x_k is state of neuron, k



→ Probability of state chang

$$P = \frac{1}{1 + e^{(-\Delta E_k / T)}}$$

Network Topology

→ It is building block of ANN.

→ Topology refers to the way neurons are linked.

→ It is a significant factor in n/w functioning & learning

Then

Soft Computing

An intelligent machine is technologically advanced machine that responds to world. It can learn from its experience. It includes AI based s/w system such as chatbots.

Chars
1) Real time stimulan
2) learning is based on ML
3) Action based on sensing data.

Soft Computing

→ use of approx. calculan to provide usable result to complex computational prob.
→ It can solve prob which are too time consuming or hard to solve with hardware.
→ Also kla Computational Intelligence.

Appⁿ → Image Processing
→ Robotics
→ Gaming Products
→ Handwriting detecn.

Perceptron ~~Architecture~~ Model

→ Perceptron is categorized as the simplest form of a neural net (single layer NN).
→ It consists of a single neuron with adjustable synaptic wts & bias.
→ A perceptron has one or more i/ps, a process & only one o/p.
→ It is used as algo. to facilitate SL of binary classifier.
→ This algo. enable neurons to learn & process elements in the training set one at a time.

Perceptron Arch. consist of 4 parts:

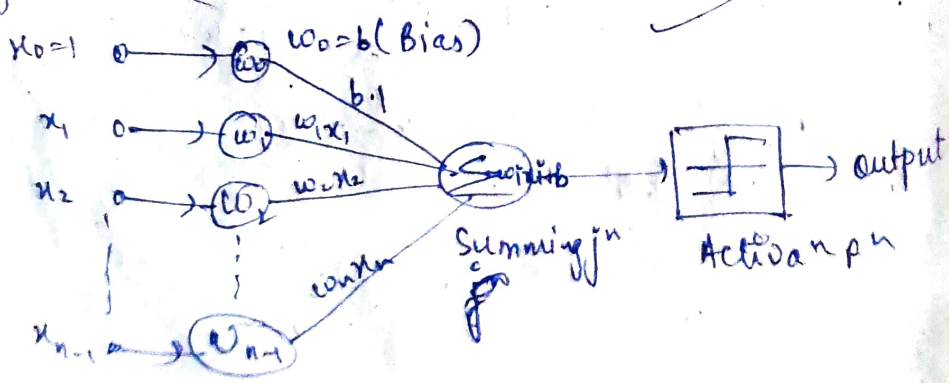
I/Ps: Features are taken as i/ps inside perceptron algo. I/Ps are denoted by $x_1, x_2, \ldots x_n$.

Weights: Determines how much influence ip will have on o/p. Represents the strength of the connecn b/w units.

Bias: Adjusts the o/p along with the weighted sum of i/ps to the neurons.

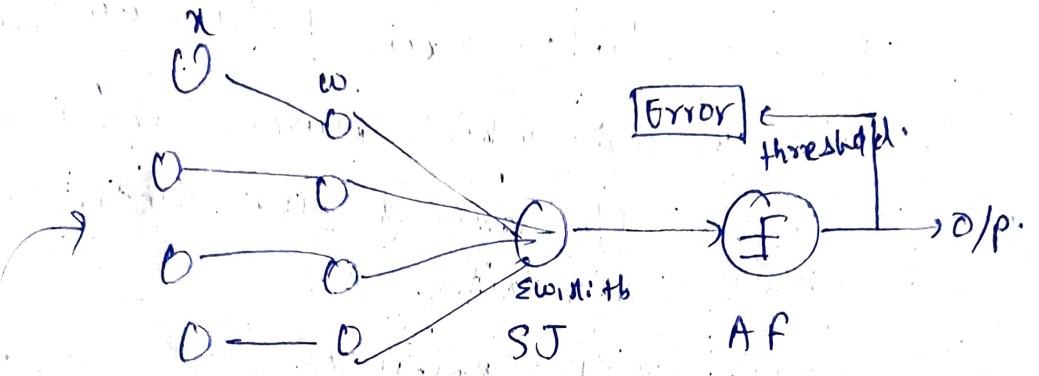
Activan fn: Used to change the value of Neural N/w to 0 or 1. The conversion of value is done to make a data set easy to classify.

Summing fn: $(\sum w_i x_i)$



Perceptron Learning Rule

It states that the algo. would automatically learn the optimal wt. coefficients. The i/p features are multiplied with these wts. to determine if the neuron fires or not.



Rule:- Perceptron receives multiple i/p signals, & if the sum of the i/p signals exceeds a certain threshold, it either outputs a signal 'f(x)' or doesnot return an o/p. It is used to predict the class of a sample.

$$f(x) = \begin{cases} 1 & ; \text{if } \Sigma w_i x + b \geq 0 \\ 0 & ; \text{if } \Sigma w_i x + b \leq 0 \end{cases}$$

w → wts.
b → bias
x → i/ps.
n → no. of i/ps.

Output can be: $f(x) = 0$ or 1 } or $f(x) = -1$ or 1 }

→ o/p can be '0' or '1'. It can also be '-1' & '1' depending on the activation fn.

Biological Neural N/w

A typical biological neuron consists of :-

Dendrites: Tree like branches responsible for processing info. received from other neurons.

Soma: Cell body of neuron responsible for processing info. received from dendrites.

Axon: Cable through which neurons send info.

Synapses: Connecⁿ b/w axon & other ~~dendrites~~ neuron's dendrites.

ANN	BNN
Node	Soma
I/p	Dendrites
Weights	Synapse
O/p	Axon

	ANN	vs	BNN
Processing	Fast, Inferior		Slow, Superior
Size	$10^2 - 10^4$ nodes		$10^{11} - 10^{15}$ neurons.
Storage	Continuous memory locaⁿs		Synapse
Fault Tolerance	Robust, fault tolerant		performance degrades with damage.

Limitaⁿs of perceptron :-

→ Works only with linearly separable classes.
→ Unable to learn logic fⁿs like XOR.
→ O/p is only a binary no. (0, 1) due to hard limit transfer fⁿ.

Multilayer Perceptron

→ It has the same model str. like single layer perceptron, but dif. no. of layers.

→ This model is kla the Back Propogation Learning Algo., which executes in 2 stages :-

(i) Forward Stage :- Activation fn start from i/p layer in the forward stage & terminate on the o/p layer.

(ii) Backward Stage :- Weight & Bias values are modified as per the model's requirement.

→ MLP uses sigmoidal activation fn.

→ It has multiple interconnected perceptron organized in dif. sequential layers; an i/p layer, one or more hidden layers & an o/p layer.

→ Every unit in a layer is connected to every unit in the ~~other~~ next layer.

→ O/P is passed as an i/p to next layer.

Advs of MLP :-

→ Works well with both small & large i/p data.
→ can be used to solve complex non-linear probs.
→ Helps to obtain quick predict after learning.
→ Works with logical fns like XOR.

Limitations of MLP

- Computations are difficult & time taking.
- Depends on the quality of training.
- Its difficult to predict, how much the dependent var. affects the independent var.
- Sensitive to feature scaling.
- Requires the tuning of a no. of hyperparameters, eg. no of hidden neurons.

Multilayer Feed forward Neural Network

- In a multilayer n/w, o/p of one layer is passed as an i/p to the next layer which is propogated further until the last layer. Hence, the name, feed forward n/w is given.
- Signal follows one approach only (i/p to o/p).
- There is no loop (feedback).
- O/p of some layer does not influence it.
- Such n/w can be used in pattern recognition.
- Backpropogan learning algo. performs learning in FFNN.

Backpropogan Algorithm

It is a fn of NN that calculates the gradient of error fn.
It has set of methods to efficiently train FNN fo

Back Propoga^n

→ It is a std. method of training ANN.

→ It is a method of repeatedly adjusts the wts. of the connects in the n/w so as to minimize the dif. b/w actual & desired o/p.

→ This method looks for a min value of the error fn in weight space using either Delta rule / Gradient Descent alga.

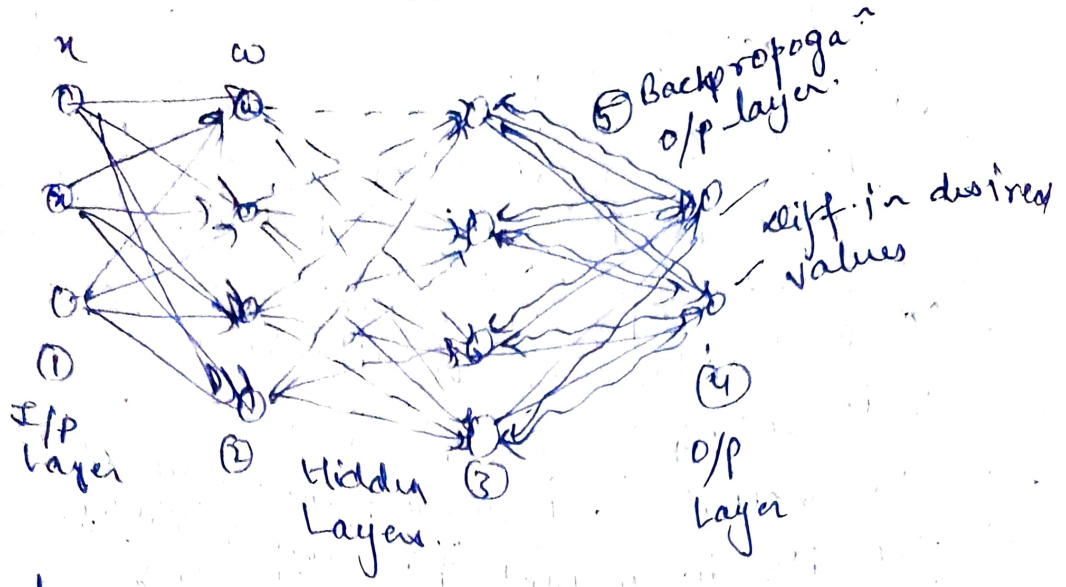
Steps:

1) Input 'x' arrives through the pre-connected path.
2) Inputs are modelled using randomly assigned wt- 'w'.
3) Calculate the o/p for each neuron from the i/p layer to the hidden layer, to the o/p layer.
4) Calculate the error in the o/p.

$$\text{Error}_B = \text{Actual O/P} - \text{Desired O/P}$$

5) Travel back from the o/p layer to the hidden layer to adjust the wts such that error is decreased.

6) Repeat ~~this~~ the process until the desired o/p is achieved.



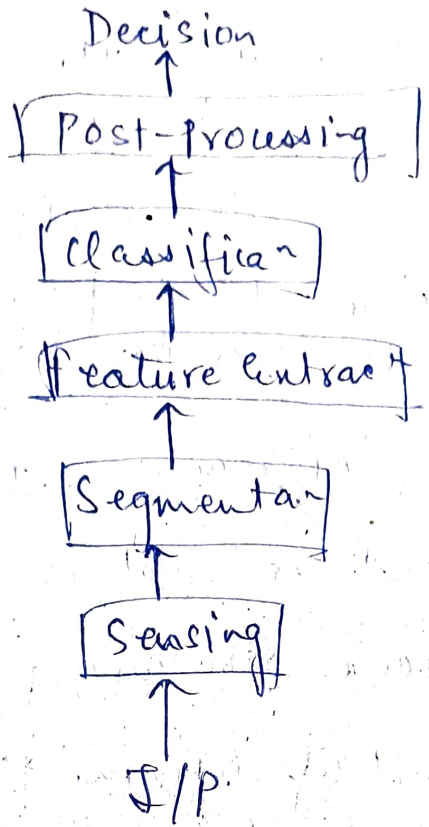
Advantages of Backpropoga^n

→ It is fast, simple, easy to program.
→ It is flexible method as it doesn't require the prior knowledge about the n/w.
→ It is mainly useful for Deep NNs working on error prone projs. such as image or speech recogn^n.

Pattern Recognition

Goal :-
→ It classifies the obj. into a no. of categories or classes.
→ Def^n :- It can be defined as the act of taking in raw data & taking an action based on the category of the data.

Components of Pattern Recognition :-



Appⁿs :-

- Machine Vision
- Computer Aided Diagnosis (CAD)
- Speech Recogniⁿ
- Character Recogniⁿ
- Manufacturing 3D-images.
- Fingerprint Identificaⁿ.
- Industria Automaⁿ → to identify if the product is defective or not

Design Principls :- The design of pattern recogniⁿ system usually follows the repetⁿ of a no. of diff. activities :-

1. Data collecⁿ
2. Feature choice
3. Model choice
4. Training of classifier.
5. Evaluaⁿ of classifier.

Approaches of PR System:-

(1) Statistical Approach:- The decision theoretic approach is based on the use of decision f^n to classify objects.

→ A decision f^n maps pattern vector 'x' into decision regions of 'D'.

(2) Syntactic Approach - The structured approach is based on the uniqueness of syntactic str. among the obj. classes.

→ Instead of defining grammar in terms of an alphabet or chars. or terminal words, the vocabulary is based on 'shape primitives'.

Phonetic Typewriter :

→ One of the earliest & wellknown appⁿ of ANN.
→ It is set in the field of speech recognition & prob. is to classify phonemes.

Unit 5

Evolutionary Computing :

→ In CS, EC is a family of algos. for global optimisaⁿ inspired by biological evoluⁿ, and the subfield of AI & soft computing algos.

→ In technical terms, they are a family of populaⁿ-based trial & error prob. solvers with a metaheuristic or stochastic optimisaⁿ character.

→ In EC, An initial set of candidate sol'ns is ~~generate~~ generated & iteratively updated.

→ Each new generaⁿ is produced by stochastically removing less desired sol'ns, & introducing small random changes.

→ In biological terminology, a populaⁿ of soluⁿ is is subjected to natural seleeⁿ & mutaⁿ. ~~As a result~~ As a result, the populaⁿ will gradually evolve to ↑se in fitness, in this case the chosen fitness fⁿ of the algo.

→ EC techniques can produce highly optimized sol'ns in a wide range of prob. setting.

Genetic Algorithm.

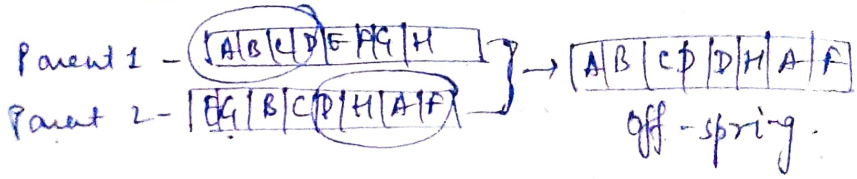
- based on the principle of survival of fittest.
- from a initial random popula^n of soln, popula^n is advanced through selection, mutation & crossover operators.
- Popula^n goes through an iterative procedure in which it reaches various states called Generation.
- The next popula^n is selected randomly.
- The best pt. in the popula^n approaches an optimal soln.
- The soln is coded as string of bits or real nos.

Operators in GA :-

1) Encoding :- First step.

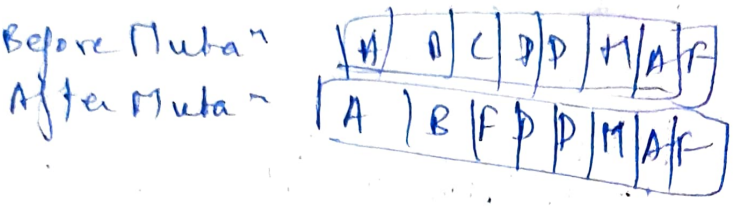
2) Selection :- primary objective is to emphasize the good soln & eliminate the bad solns in a popula^n, while keeping the popula^n size const.
   - → Diff for choosing the best soln, there exist fitness fn, tournament select etc.

3) Crossover :- process of taking more than one parent soln & producing a child soln from them.
   - → By recombining portion of good solns, the GA is more likely to create a better soln.



4) Mutation → It encourages genetic diversity amongst solns & attempts to prevent the GA to give solns that are very close to one another

→ In this, a given sol^{n} may change entirely from the previous sol^{n} to give a much better sol^{n}.



Types of GA :-

1) Simple GA :- It starts with the creation of an initial population of size 'N'.

→ Then we evaluate the fitness of each soln

→ We are biased towards highly fit solns

2) Parallel GA :- It uses multiple GAs to solve a single task.

→ After the algos have completed their job, the best soln of every algo is selected as the soln.

→ It is also k/a Island Model.

3) Distributed GA → Independent Algos run on seperate machines.

→ Here each algo., in turn, may be a parallel GA.

4) Steady State GA → It means there are no generations.

→ Used for small population size.

Fitness function :-

→ It is simply described as a f^n that takes as a i/p, potential sol^n to a prob., & gives as o/p, how fit or how excellent an ans. is in relation to the prob. under discussion.

→ It is used to evaluate the quality of single sol^n in a popⁿ.

→ Eg. Consider Three vars. x, y, z.
Goal: to determine the optimal combina^n of nos. for x, y & z, such that their sum equals a given value 't'.

~~For best sol^n,~~

- The sum x+y+z must be reduced to prevent it from straying from the value t i.e. |x+y+z − t| should be equal to zero.
- As a result, the fitness f^n may be thought as the inverse of the f^n |x+y+z−t|.