UNIT-4 (DIVYA BHARDWAJ)

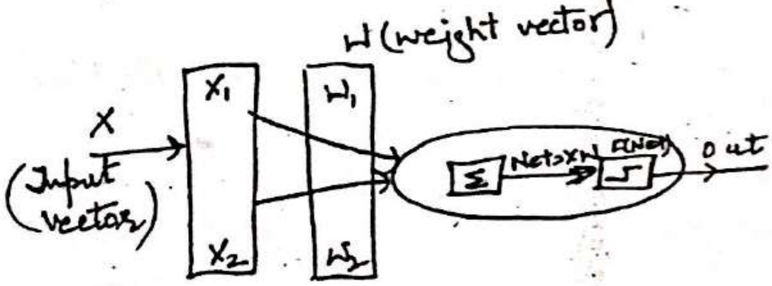
(NEURAL NETWORKS & FUZZY LOGIC)

28.

Syllabus :- Optical Neural Networks: Vector Matrix Multiplier, Hop field net using electro optical matrix multipliers, Holographic correlator, Optical hopfield net using Volume Holograms.

The Cognitrons and Neocognitrons: Their structure & training.

Genetic Algorithms: Elements, a simple genetic algorithm, working of genetic algorithms, evolving neural networks



Ⓐ VECTOR MATRIX MULTIPLIERS :

The operation of most ANNs can be described mathematically as a series of vector matrix multiplications, one for each layer.

→ To calculate the output of a layer, an input vector is applied and then multiplied by the weight matrix to produce the NET vector.

→ This vector is then operated on by the activation function F to produce the output vector for that layer.

→ In biological neural networks, this operation is performed by large no of neurons simultaneously, hence the system responds rapidly despite the slowness of each neuron.

→ But when ANNs are simulated on general purpose computers, the inherently parallel nature of the computation is lost, each operation must be performed sequentially.

→ Despite the rapidity of individual computations, the no of operations required for the matrix multiplication is proportional to the square of the size of the input vector and computation time can become intolerably long.

→ $\boxed{\text{limitations of Vector Matrix Multiplier}}$.

Example of Vector Matrix Multiplier:

→ Consider the hopfield network taking input vector as (1 0 1 1).

is discre

Step

this example demonstrate the working of discrete hopfield net. (2)

Step-I: We have the given input vector as $x = (1\ 0\ 1\ 1)$

Step-II: Weight matrix will be computed as.

$$W = x'x.$$

$$\begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix}_{4\times1} \cdot \begin{bmatrix} 1 & -1 & 1 & 1 \end{bmatrix}_{1\times4}$$

$$W = \begin{bmatrix} 1 & -1 & 1 & 1 \\ -1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & 1 \end{bmatrix}_{4\times4}$$

Step-III: Now the output vector will be obtained by using activation fn of Hopfield net as.

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

Step-IV: Input and weight matrix will produce net input and then activation fn will be applied on net input to obtain ~~o~~ output.

Now so as to overcome the limitations of vector matrix multiplier i.e sequential nature and large time taken for computation, we will be switching to next topic i.e Optical Neural Networks.

There

(B) Optical Neural Networks :

→ Optical neural networks interconnect neurons with light beams.

→ A optical neural network is a physical implementation of an ANN with optical components.

→ Biological neural networks function on an electrochemical basis, while optical neural networks use electromagnetic waves.

→ All the signal paths operate simultaneously which results in true data rate.

→ The strengths of weights are stored in holograms [3-D image formed by interference of light beams from a coherent light source] with high density. These weights can be modified during operation to produce a fully adaptive system.

There are two categories of optical neural network. (3)

Electro - Optical Matrix Multiplier

Holographic correlators

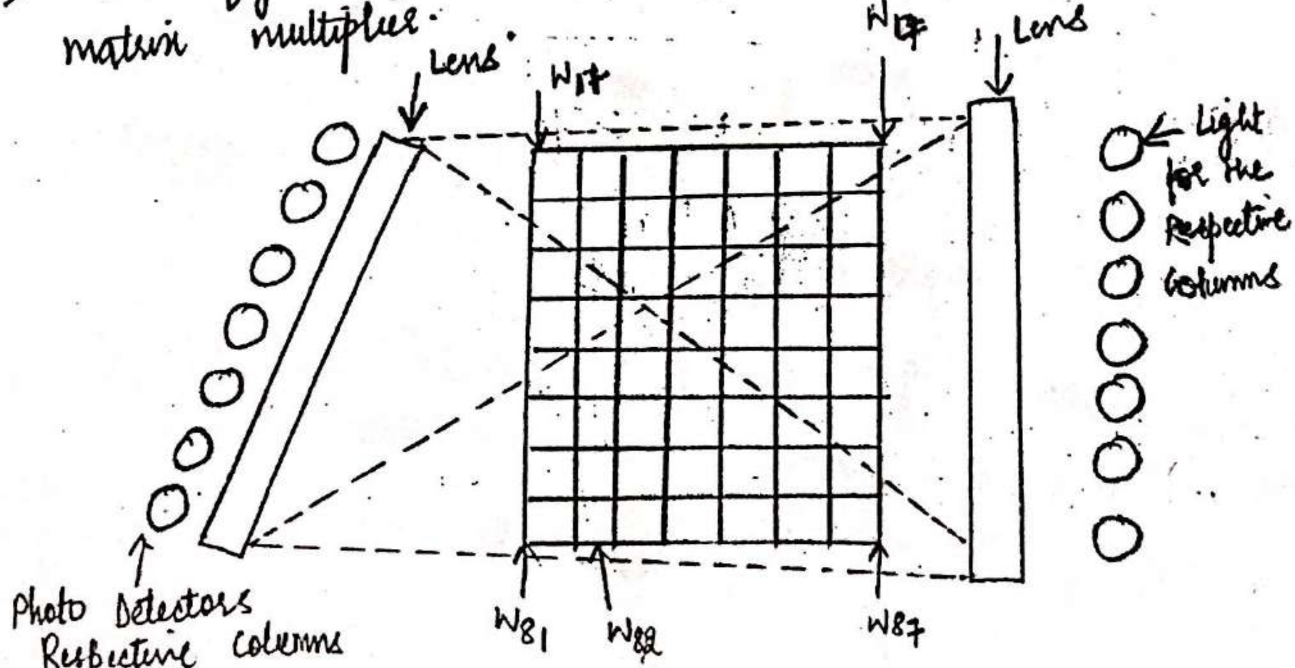
Electro - Optical Matrix Multiplier :-

→ Electro - optical matrix multiplier provides a mean for performing matrix multiplication in parallel.

→ The network speed is limited only by the available electro - optical components.

→ The computational time is potentially in the nanosecond range.

→ The figure shows the electro - optical vector matrix multiplier.



→ The above system is capable of multiplying an 8 element input vector by 8×7 matrix, which produces seven-element NET vector.

→ There is a column of light sources that passes its rays through a lens, such that each light - illuminates a single row of weight shield.

→ The weight shield is a photographic film in which the transmittance of each square is proportional to the weight.

→ There is an another lens, which focuses the light from each column of the shield to a corresponding photo detector.

→ The NET is calculated by,

$$\boxed{NET_k = \sum_i w_{ik} x_i}$$

where NET_k = NET output of neuron k.
w_{ik} = weight from neuron i to neuron K.
x_i = input vector component i.

→ The output of each photo detector will represent the dot product between the input vector and a column of the weight matrix.

Scanned with CamScanner

→ The set of outputs is a vector equal to the product of the input vector with weight matrix. (4)

→ Hence the matrix multiplication is done in parallel.

→ This electro-optical matrix multiplier can be used in Hopfield net and Bidirectional associative memory.

→ Hopfield Net using electro-optical Matrix Multiplier :

→ If the photo detector outputs of the networks are feedback to drive the corresponding light inputs, an electro-optical hopfield net is produced.

→ To do so, an threshold activation function must be provided.

→ Bidirectional Associative Memories using electro-optical Matrix Multiplier.

- If the two systems (above figure) are cascaded, an electro-optical BAM is produced.
- To ensure stability, the second weight mask must be the transpose of the first.

[For more information, Refer BAM from UNIT-3 notes and Hopfield net from Unit-2 notes]

① Holographic Correlators :

→ In this case, the reference images are stored in a thin hologram and retrieve them in a coherently, illuminated feedback loop.

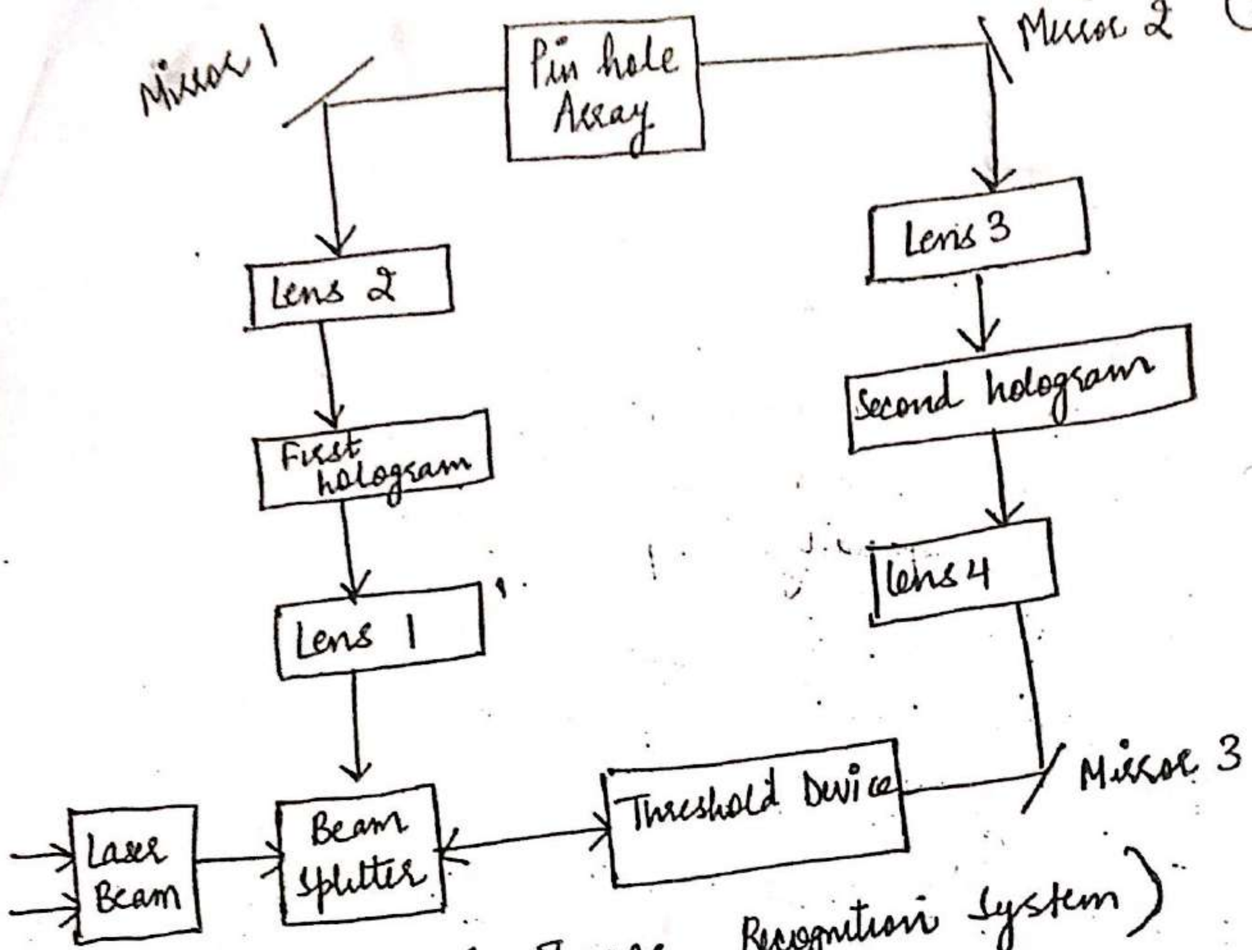
→ A noisy or incomplete input image is applied to the system and can simultaneously be correlated optically with all of the stored reference images.

→ These correlations can be threshold and are feedback to the input, where the strongest correlation reinforces the input image.

→ The image which is enhanced passes around the loop repeatedly, which approaches the stored image more closely on each pass, up to the system getting stabilized on desired image.

→ This optical correlator can be used for image recognition.

→ A generalized optical image recognition system is shown in figure.



(Optical Image Recognition System)

<u>Working</u>:

→ The input to the system is an image from a laser beam.

→ This passes through a beam splitter, which passes it to the threshold device.

→ The image is reflected then gets reflects from the threshold device, passes back to the beam splitter, then goes to lens 1, which makes it falls on the first hologram.

→ The first hologram contains several stored images. The image then gets correlated with each of them, that produces patterns of light.

→ The brightness of the patterns varies with the degree of correlation.

→ The projected image from lens 2 and mirror 1 passes through pin hole array, at which they are spatially seperated.

→ From this array, the light patterns goes to mirror 2, through lens 3 and then applied to second hologram.

→ Lens 4 and mirror 3, then produce, superposition of the multiple correlated images onto the back side of threshold device.

→ Operation of threshold device:
Its front surface reflects most strongly that pattern which is brightest on its rear surface.

→ optical neural networks is advantageous in terms of speed and interconnect density.

⇒ They can virtually construct any network architecture.

E) VOLUME HOLOGRAMS :-

→ volume holograms are holograms where the thickness of the recording material is much larger than the light wavelength used for recording.

→ Volume holograms are also called thick holograms or Bragg holograms.

→ A volume hologram is usually made by exposing a photo-thermo refractive glass to an interference pattern from an ultraviolet laser.

→ For volume phase holograms it is possible to diffract 100 % of the incoming reference light into the signal wave. ie full distraction of light can be achieved.

→ Applications of volume holograms :-

i) Distributed feedback lasers :

ii) Holographic memory devices for holographic data storage.

iii) Fiber Bragg gratings that employ volume holographic gratings encrypted into an optical fiber.

iv) Imaging spectroscopy :- can be achieved by selecting a single wavelength for each pixel in a full camera field.

→ An optical hopfield net using volume holograms

→ It operates as an implementation of the hopfield net, seeking a minimum on an optically generated energy surface.

→ when a noisy or incomplete input pattern is applied, the system converges to the stored image ~~i.e most~~ similar, thereby functioning as an optical associative memory.

(F) COGNITRONS :-

→ The cognitron was developed by Fukushima in 1975.

→ It is a hypothetical model of the human perception system.

→ Architecture :

- The cognitron is made up of layers of neurons which are connected by synapses.
- There exist post-synaptic neurons and pre-synaptic neurons.

→ The pre- synaptic neuron in one layer feeds the post- synaptic neuron in the next layer.

→ Also, there exist two types of cells,

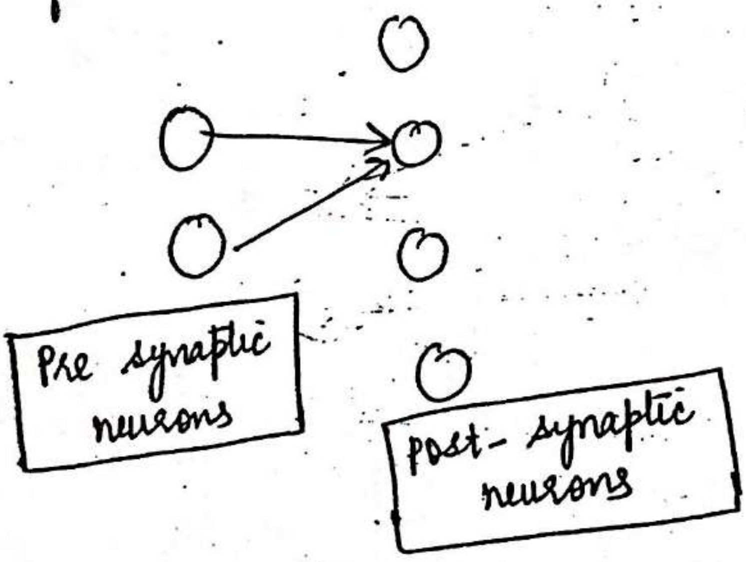
<u>Excitatory cells</u>

The excitatory cells make the post- synaptic cell to fire.

<u>Inhibitory cells</u>

The purpose of inhibitory cells is to reduce the firing of post- synaptic cell.

→ The firing of neuron depends on the weighted sums of its excitatory and inhibitory inputs.



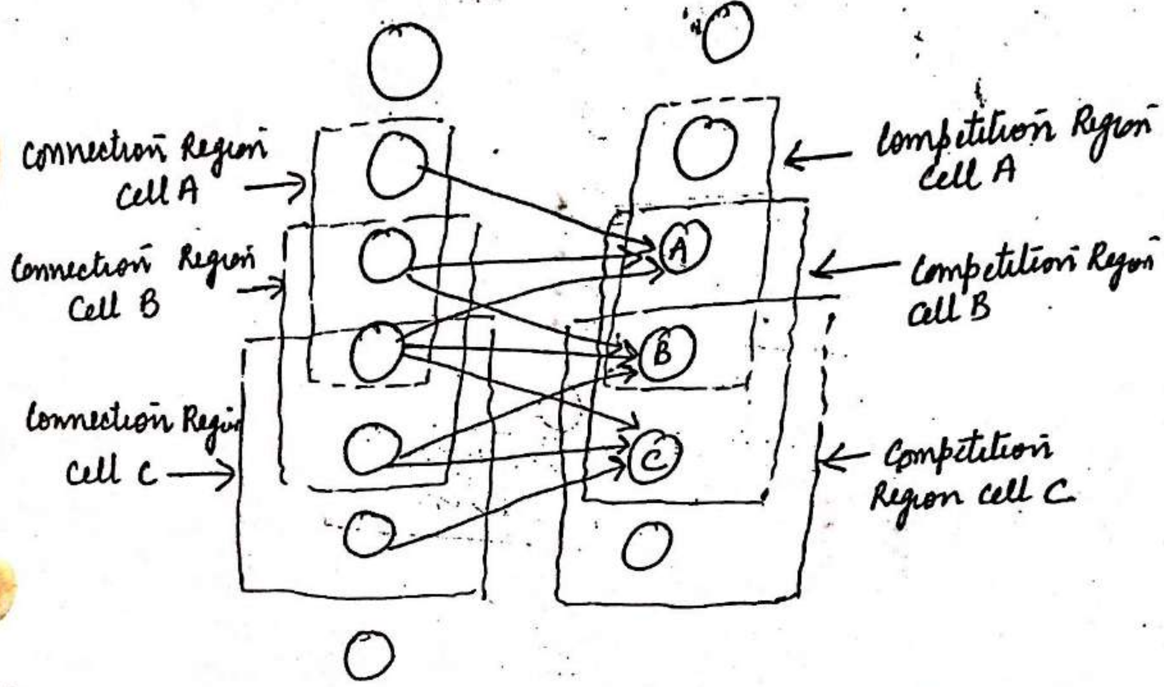
• <u>Training of cognitrons</u> :-

→ The training employed in cognitron is <u>unsupervised learning</u>.

→ For a given training set of input patterns, the network self-organizes by adjusting its synaptic strengths.

→ In a given region of a layer, only the neuron which most vigorously fires is trained.

→ But the others which are already well trained, are shown by the strength of their firing, which has their synaptic weights increased to further enhance their firing.. Hence, competition among nearby cells is adopted.



[Connection and Competitive Regions].

- Excitatory Neuron

→ The output of excitatory cognition neuron is determined by the ratio of excitatory inputs to inhibitory inputs.

→ The total excitatory input to a neuron E is simply the weighted sum of the inputs from the excitatory neurons in previous layer.

→ Also, the total inhibitory input to a neuron I, is the weighted sum of the inputs from the inhibitory neurons.

→ It can be given as,

$$\boxed{E = \Sigma w_i x_i}$$

$$\boxed{I = \Sigma v_j y_j}$$

where,

w_i = The weight of the ith excitatory synapse
x_i = The ~~output~~ in of the " " neuron.
v_j = The weight of the jth inhibitory synapse.
y_j = The ~~output~~ in of " " " neuron.

→ Here the weights take only positive values.

→ The output of the neuron is then calculated by,

$$\boxed{Net = \left[\frac{1+E}{1+I}\right] - 1}$$

$$\boxed{output = \begin{cases} net & for \; net \geq 0 \\ 0 & for \; net < 0 \end{cases}}$$

• Inhibitory Neuron :-

→ We know that in cognitron a layer consist of both excitatory and inhibitory cells.

→ The weights coming into inhibitory cells are not modified during training, their weights into any inhibitory neuron is equal to one.

→ Hence the output of the inhibitory cell is simply the weighted sum of its inputs; which in this case is the arithmetic mean of excitatory output to which it connects.

$$\boxed{Inhibit = \sum_i u_i \; Output_i}$$

$$\sum_i u_i = 1$$

u_i = inhibitory weight i

$output_i$ = excitatory neuron output.

Training Problems in Cognitron :-

The weights associated with an excitatory neuron are adjusted only when it is firing more strongly than any of the neighboring cells in competitive region.

- The change in one of its weights in the case is calculated as,

$$\boxed{\delta w_i = b u_j x_j}$$

where,

u_j – The inhibitory weight coming from neuron j in pre-synaptic layer to the inhibitory neuron i

x_j – The output of neuron j in pre-synaptic layer.

w_i – Excitatory weight i.

b – learning rate coefficient.

- The change in the inhibitory weight into neuron i in post-synaptic layer is proportional to the ratio of the weighted sum of excitatory input to twice the inhibitory input.
- This is calculated using formula:

$$\boxed{\delta v_i = b \left(\sum_j w_j x_j\right) / (2 \times \text{Inhibit}_i)}$$

- When no neuron fires in the competition region, other weight adjustment formula is used.

- In cases, when there is no winner in a neuron's competition region, its weight changes are calculated as follows:

$$\boxed{\begin{aligned} \delta w_i &= b u_j x_j \\ \delta v_i &= b \text{ Inhibit}_i \end{aligned}}$$

where b is a positive training co-efficient more than 1.

- Cognitron being a self-organizing network failed to recognize position or rotation distorted characters. This failure was overcome in the neo-cognitron network.

(G) NEO COGNITRONS :-

→ The neocognitrons is powerful in its ability to recognize patterns despite translation, rotation, distortion and changes in scale.

→ It is modeled towards human visual system.

It c...
image...
su...

It can accept 2-D patterns like those imaged onto the retina and processes them in successive layers as that of human visual cortex.

→ It is a hierarchical network in which there are many layers with sparse and localized pattern of connectivity between the layers.

→ The neocognitron is based on supervised learning.

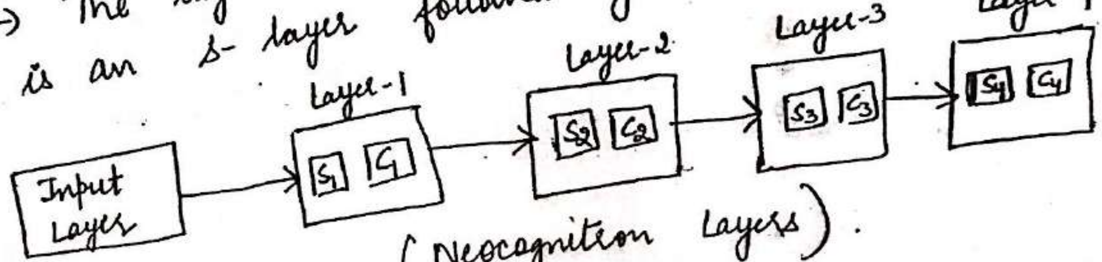
→ This net is designed so that it is insensitive to the variations in the positions, style etc of the patterns to be considered.

• Architecture of Neocognitrons:

→ The cognitron consists of several layers.

→ Each layer has units arranged in no of square arrays.

→ The layers are arranged in pairs, i.e there is an S-layer followed by C-layer.



(Neocognitron Layers).

• S- Cells : Simple Cells :

→ The S arrays are trained to respond to a particular pattern or group of patterns.

→ All cells in a simple - cell plane respond to the same pattern.

→ Each simple cell is sensitive to a restricted area by the input pattern, which is called its receptive range.

• C- Cells : Complex Cells :

→ The C- cells combines the outputs from S- cells and simultaneously thin out the no of units in each array.

→ Complex cells serve to make the system less sensitive to the position of patterns in the input field. [overcomes the problem of recognizing position or rotation distorted characters].

→ Algorithm Calculations :-

The S-type cell possess excitatory signals received from the units in the previous layer and possess inhibitory signals obtained within same layer.

$$v = \sqrt{\sum\sum t_i c_i^2}$$

where v = a layer that consist of only one cell plane.

t_i = Fixed weight.

c_i = output from C unit.

→ The S-unit has its scaled input as,

$$x = \frac{1+e}{1+vw_0} - 1$$

where $e = \sum_i c_i w_i$

w_i = weights adjustable from C to S

w_0 = " " between v and S

e = excitatory input from C units.

→ The activation of output signal is

$$S = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

→ The net input of C- layer is

$$C_{in} = \sum_i S_i \cdot x_i$$

where

S_i - Output from S unit.

x_i - fixed weight S-unit to C-unit.

The output is

$$C = \begin{cases} \dfrac{C_{in}}{a + C_{in}} & \text{if } C_{in} > 0 \\ 0 & \text{, otherwise} \end{cases}$$

'a' is a parameter that depends on the level of network performance.

- <u>Training of Neocognitrons</u> :-

→ The neocognitron network is trained layer by layer.

→ The simple cells have adjustable weights.

→ These weights connect to complex cells in the previous layer by modifiable synaptic weights, which are adjusted during the training process.

→ few weights are inhibitory, which reduces the output, few are excitatory which increases the output.

→ The simplex cells responds to a set of complex cells within its receptive range.

→ There are inhibitory cells that responds to exactly the, same complex cells.

→ The weights of the inhibitory cell are not trained.

→ The desired response of each layer may be chosen. The weights were then adjusted using conventional two - layer training methods to produce desired response.

(H) GENETIC ALGORITHMS :- (GA)

→ GA is a search- based optimization technique based on the principles of genetics and natural selection.

→ It is frequently used to find optimal or near optimal solutions to difficult problems which otherwise would take lifetime to solve.

Optimization : process of making something better.

→ GA was developed by John Holland.

→ GA Terminologies :- → i) Population : subset of all possible solutions to given problem.

ii) Genotype :- It is the population in computation space. In the computation space, the solutions are represented in a way which can be easily understood and manipulated using a computing system.

iii) Phenotype :- It is the population in actual real-world solution space in which solutions are represented in a way they are represented in real world situations.

iii) Decoding : It is a process of transforming a solution from the genotype to phenotype space.

iv) Encoding : It is a process of transforming from the phenotype to genotype space.

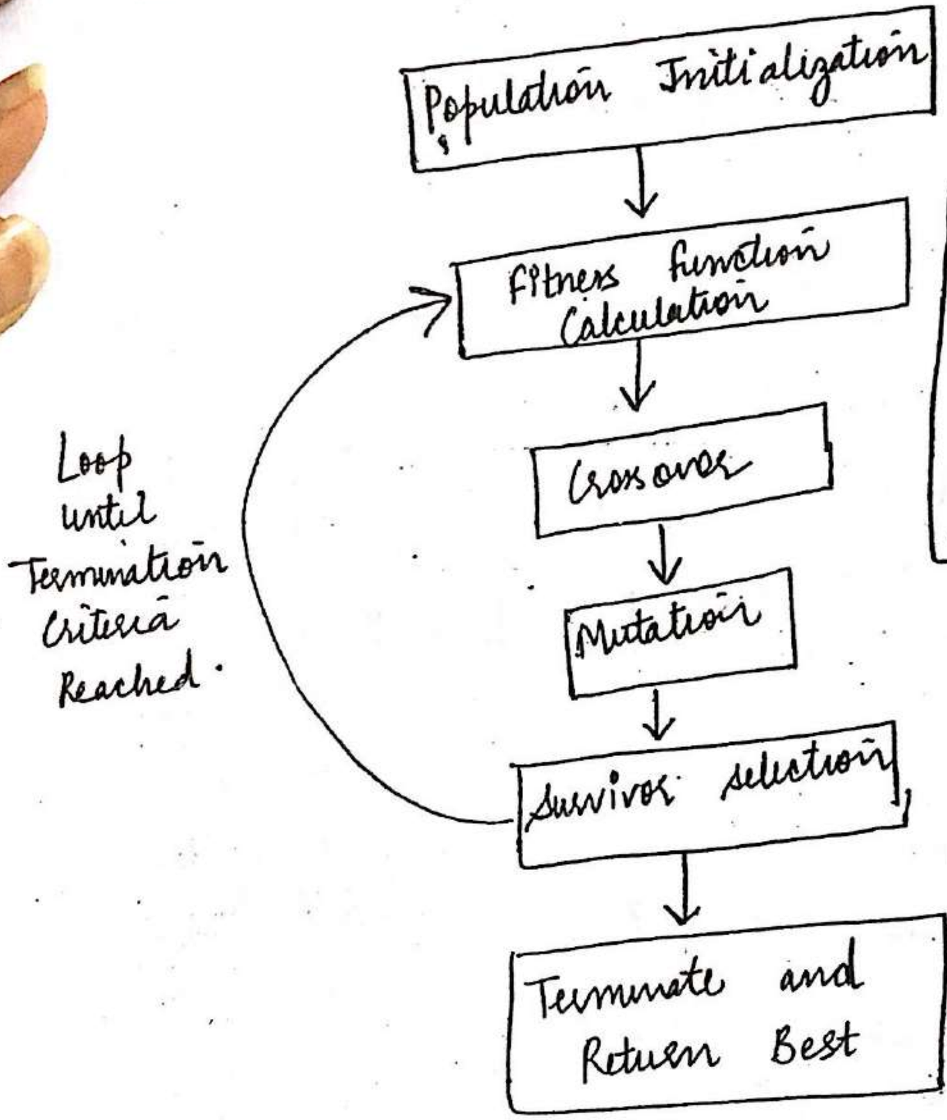
vi) Fitness function : It is a function which takes the solution as input and produces the suitability of the solution as the output.

vii) Chromosome : A chromosome is one such solution to given problem.

Genetic Operators: These alter the genetic composition of the offspring. These include cross over, mutation, selection etc.

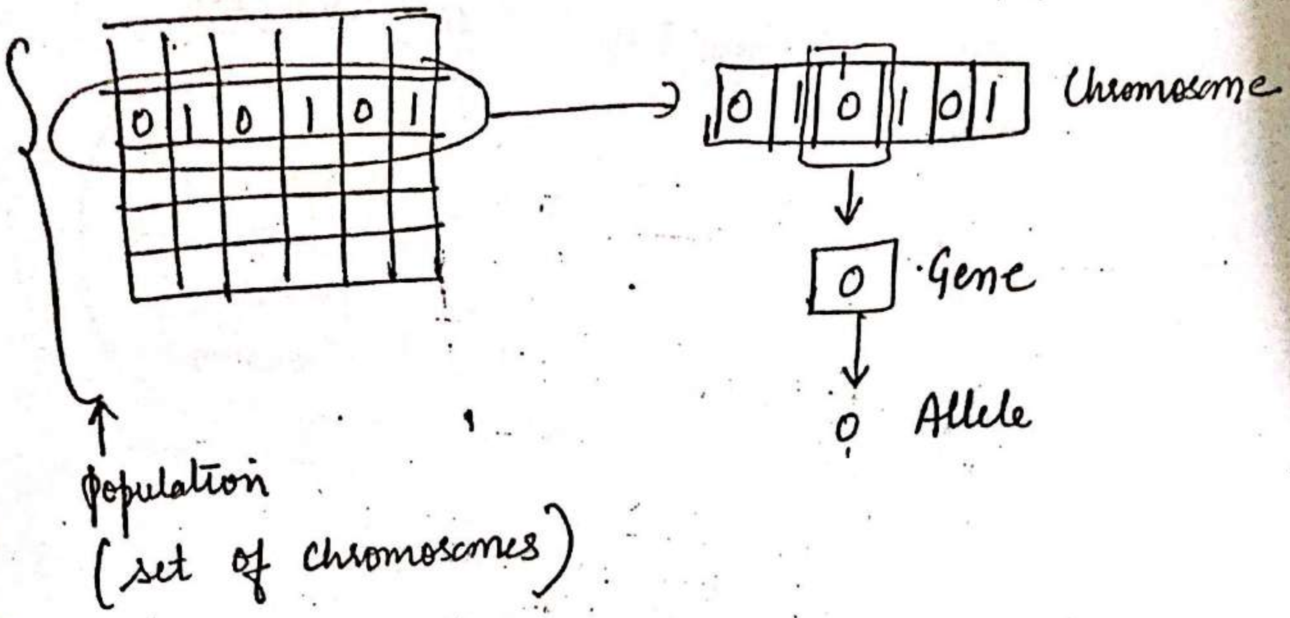
ix) Gene.:
A gene is one element position of chromosome.

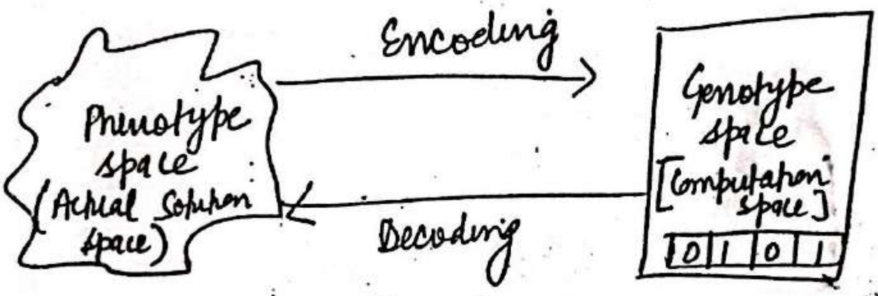
→ Basic structure of GA :-



x) Allele:
It is the value a gene takes for a particular chromosome

Representation of GA Terminologies :





Working of GA :-

i) firstly the populations are selected
ii) Every chromosome is assigned some fitness value.
iii) Crossover (Reproduction of offspring) is done between diff chromosome.

During the creation of offspring, recombination occurs (due to cross over) and in that process genes from parents form a whole new chromosome in some way.

v) The new created offspring can then be mutated.

vi) Mutation means element of DNA is modified.

vii) The fitness of an organism is measured by means of success of organism in life.

→ Representation of GA :

i) Binary Representation : Eg: 0/1 knapsack problem

ii) Real valued Representation : when we want to define the genes using continuous rather than discrete variables.

0.5	0.2	0.6	0.8	0.7

iii) Integer Representation :

Eg: If you want to encode 4 distances – North, South, East, West we can encode them as {0, 1, 2, 3}.

For such cases integer representation is desirable

1	2	3	4	3	2	1

iv) Permutation Combination :-

The solution is represented by order of elements.

Eg: Travelling Salesman Problem : visiting each city exactly once + coming back to starting city.

1	5	4	2	3

v) Hexa decimal Representation (0123456789ABCDEF)

vi) Octal Representation.

→ The various methods for selection of chromosome for parents to crossover are :

i) Roulette-wheel selection
ii) Boltzmann Selection
iii) Tournament Selection
iv) Rank selection
v) Steady-state selection.

← No need for going in details