Vector - Matrix Multipliers :- The operation of most artificial neural n/ws can be described mathematically as a series of vector matrix multipliers. One for each layer. To calculate the o/p 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 fn F to produce the OUT vector for that layer. In symbols

$$NET = XW$$

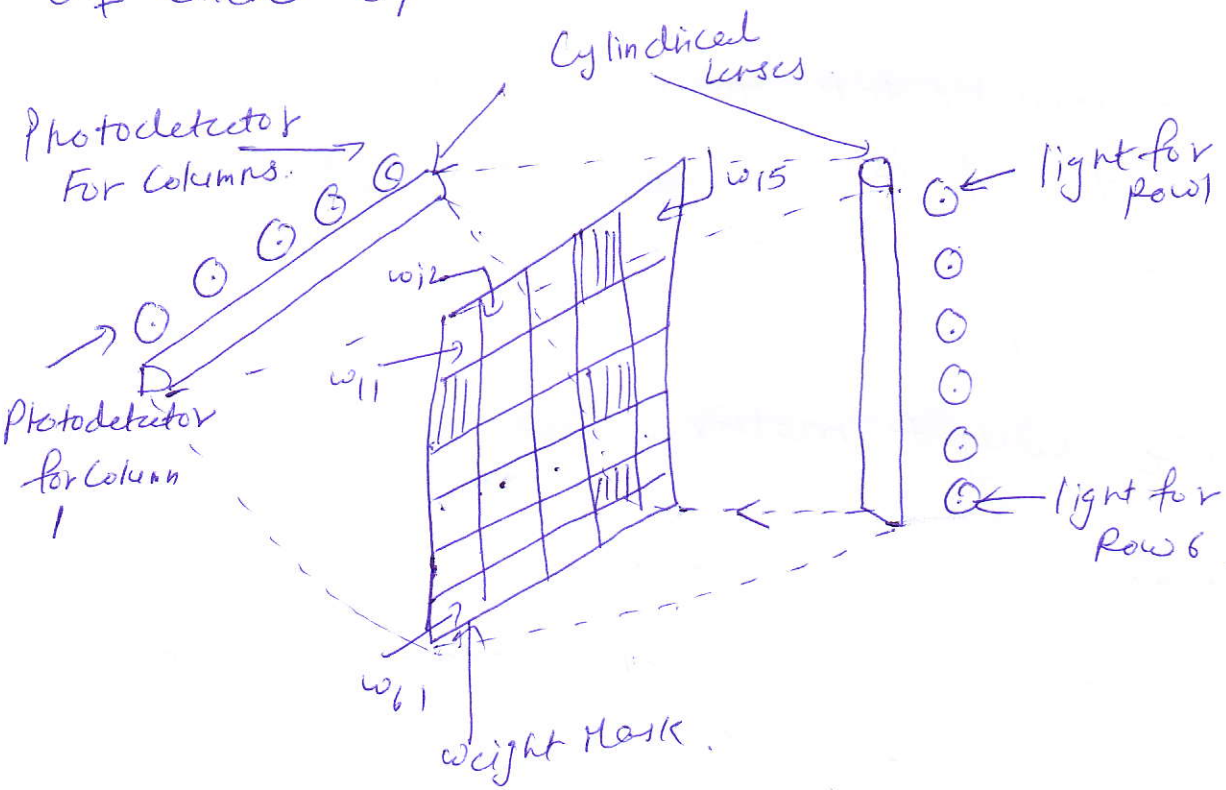
$$OUT = F(NET)$$

where
NET = the row vector formed from weighted sum of inputs
OUT = the o/p row vector
X = the input row vector
W = the weight matrix.

When ANNs are simulated on general-purpose computers, the inherently //el nature of the computation is lost; each operation must be performed sequentially. Despite the rapidity of individual computations; the no. of op^ns on required for matrix multiplication is proportional to the square of size of input

- Vector.

Electro-optical Matrix Multiplier:- Electro-optical neural nets provide a means for performing matrix multiplication in llel. Fig. shows a system capable of multiplying a six-element input vector by a six-by-five matrix, producing a five element NET vector. On the right a column of light sources passes its rays through a cylindrical lens; each light uniformly illuminates w_{11}, w_{12}, w_{15}. The weight mask may be a photographic film in which the transmitance of each square is proportional to the weight.



Electro-optical - Vector Matrix Multiplier

On the left side is a second cylindrical lens that focuses the light from each column of the mask on to a corresponding photodetector.

Thus, the light impinging on photodetector 1 is the sum of the products of the light intensities multiplied by the transmittances for column 1. Symbolically

$$NET_j = \sum_i w_{ij} x_i$$

NET_j = The NET output of neuron j

w_{ij} = the weight from neuron i to neuron j

x_i = the input vector component i

Each photodetector O/P represent the dot product b/w the input vector and a column of the weight matrix.

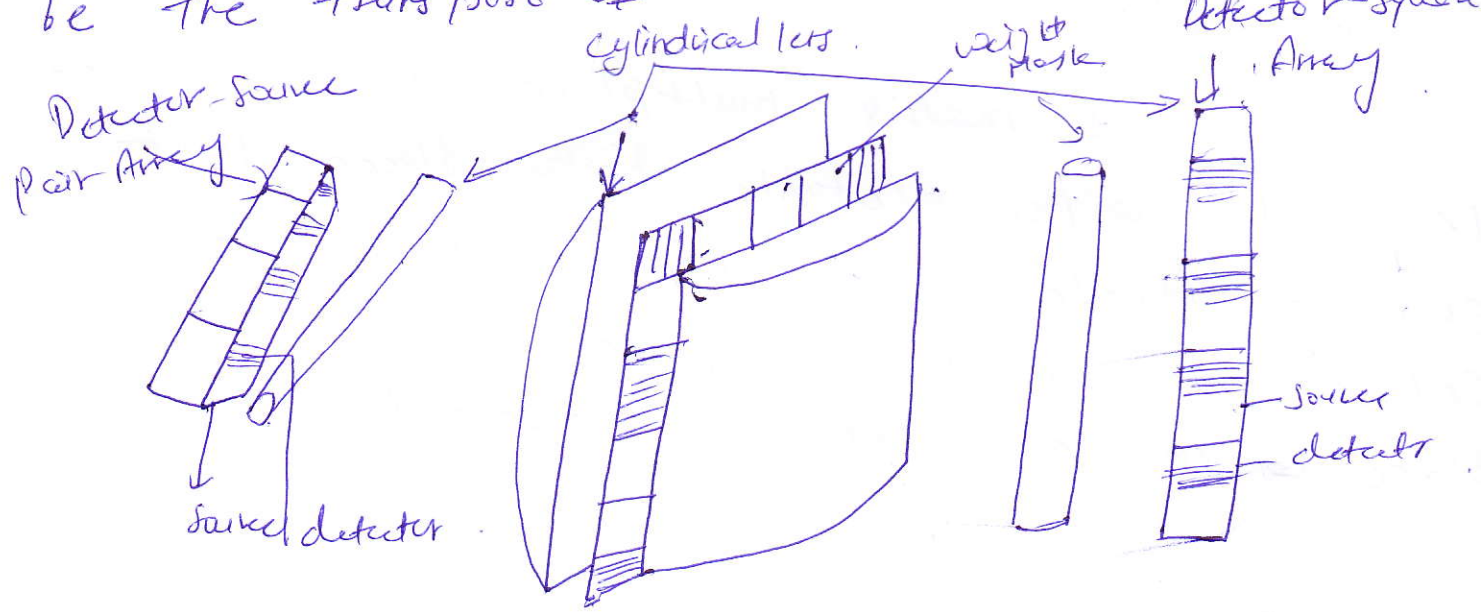
This matrix multiplication is performed in llcl. with suitable high-speed light emitting diodes and PIN detectors, the entire vector matrix multiplication can take place in less than a nano second.

Hopfield Net using using electro-optical Matrix multipl

If the photodetector o/p of the n/w are fed back to drive the corresponding light inputs, an electro-optical hopfield net is produced. To do so, a threshold activation f^n must be provided. Today, this is best done in electronics circuitry following each photodetector.

To satisfy the stability requirements the weight array must be symmetrical with transmittance set to zero for squares of the main diagonal (w_{11}, w_{12}, ..., w_{mn})

Electro-optical BAM :- If two of the system in following fig-1 are cascaded, an, electro optical BAM is produced. To ensure stability, second weight mask must be the transpose of the first.



Electro optical BAMs

Kosko has described a Compact System in which only a single mask and optical system is required. Here each photodetector and light source is replaced by a photodetector light-source pair. The operation is similar to that described for the simple photo-optical multiplier, except that the o/p from each photodetector drives its adjacent light source.

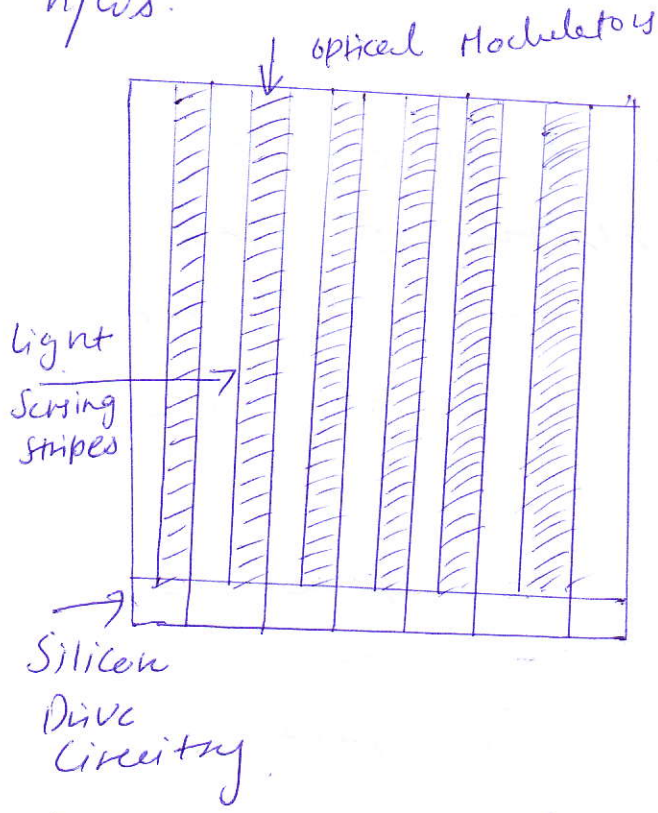
In operation, light from each light source on the right passes through the cylindrical lens, illuminating the corresponding row of weight mask. These lenses act to spread the light in a horizontal direction, while leaving it collimated in the vertical direction.

On the left each photodetector receives all of the light from a column of the weight mask, and its electronics provides the threshold f^n to produce the NET o/p. The o/p of the electronics then drives the adjacent left side light source, the light from which passes through the optics to illuminate the same column. It may be seen that the same space in the optics is occupied by light patterns passing from left to right and from right to left.

On the right side, each photodetector responds to the light from an entire row and its electronics perform the threshold f^n and drives the adjacent light source. In this way, a feedback loop is closed, coupling light sources, photodetectors and the optical system

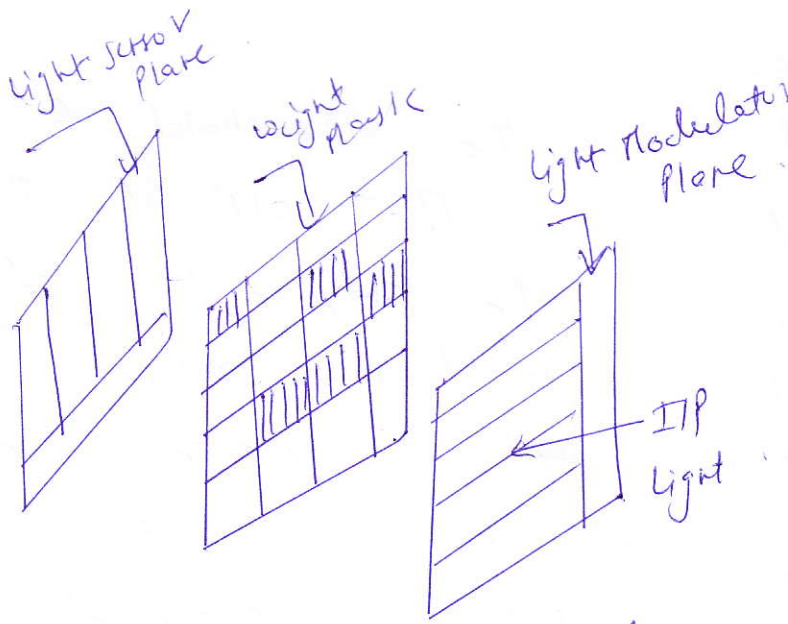
Note:- BAM stability is ensured even if the matrix is not symmetrical; also the main diagonal need not to be zero.

Linear Modular Array:- The linear modulator, a device currently under development promises to simplify substantially the structure of electro-optical n/ws.



Linear Spatial Modulator Array.

Fig 1



Linear Modulator used as an optical Matrix Multiplier

Fig 2

As shown in figure 1 ~~a~~, it consist of a thin plate with alternate stripes of light-sensing material and optical modulators. The transmisivity of each stripe of the original modulator region can be varied electronically.

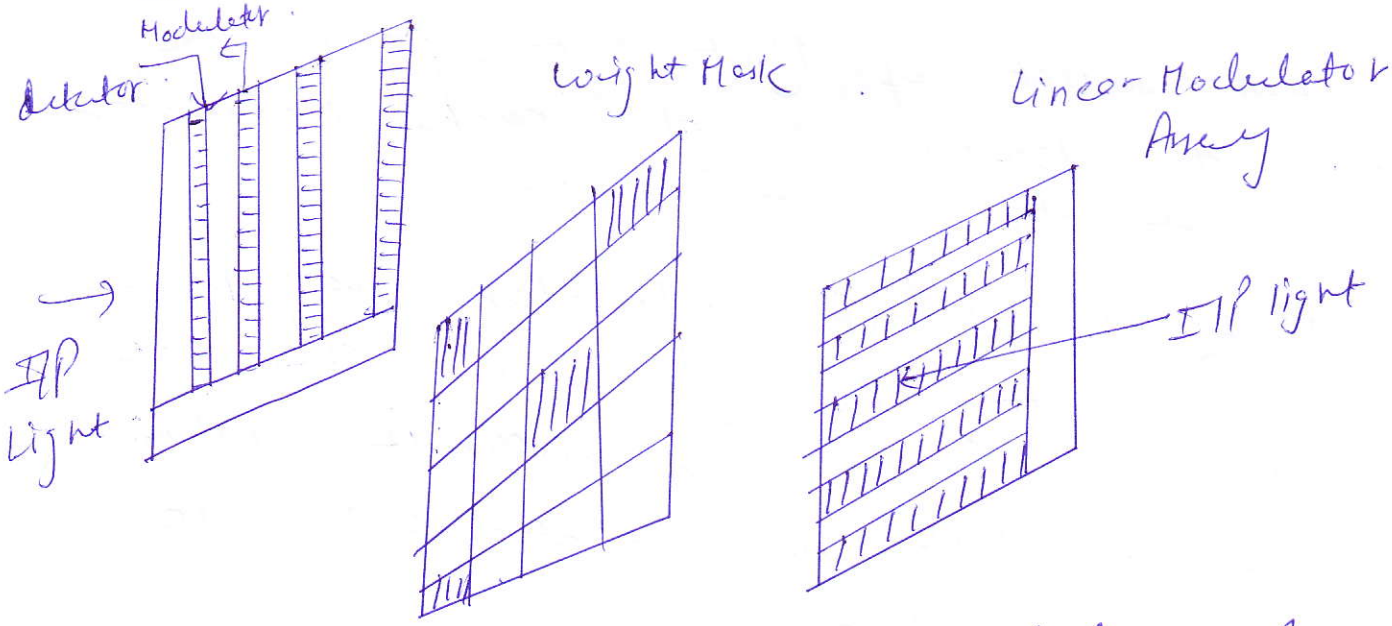
Fig 2 shows a simplified linear modulator assembly used as an optical matrix multiplier. The horizontal optical modulator stripes matrix are electronically controlled. The transmissivity of each strip corresponds to the magnitude of a component of the X input vector, thereby controlling the amount of light impinging upone the corresponding row of the weight mask.

In this system; there are no separate lights for each row; one controlled collimated source of light enters from the right and passes through each modulator strip on to the weight mask.

Becoz the linear modulator array passes collimated light, no cylindrical lenses are required. This solves the difficult problem of geometric distortion associated with the optics of earlier designs.

BAM implementation using linear Modulator arrays:-

It is similar to multiplier described above except that each column light detector strips on the left drives a threshold circuit, which in turn controls the transmissivity of its adjacent vertical strip. In this way a second collimated light source from left is modulated. and the corresponding column of the weight mask receives a controlled illumination level. This produces the necessary feedback signal to the horizontal rows of light detectors on the right, their o/p signals are thresholded and control the strips transmissivity of their corresponding horizontal light-modulator strips, thereby closing the BAM feedback loop.

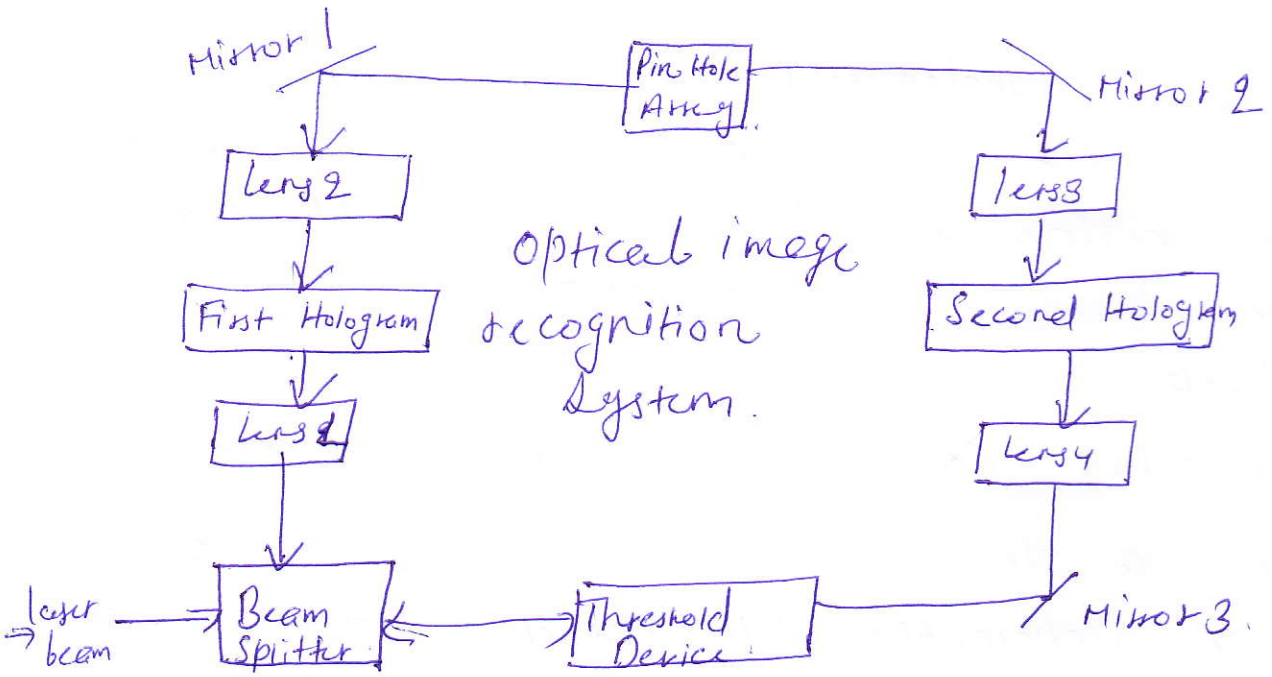


Optical BAM using Linear Modulator Arrays

Holographic Correlators:- Although there are many variations of the holographic, Correlator, their fundamental operating principles are quite similar. They all stored reference images in either a thin or volume hologram and retrive them in a coherently illuminated feedback loop.

In this case reference images are stored in a thin hologram and retrive 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 I/P, 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 stablized on the desired image. This optical correlator can be used for image recognition.

A generalized optical image recognition is shown as.



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.

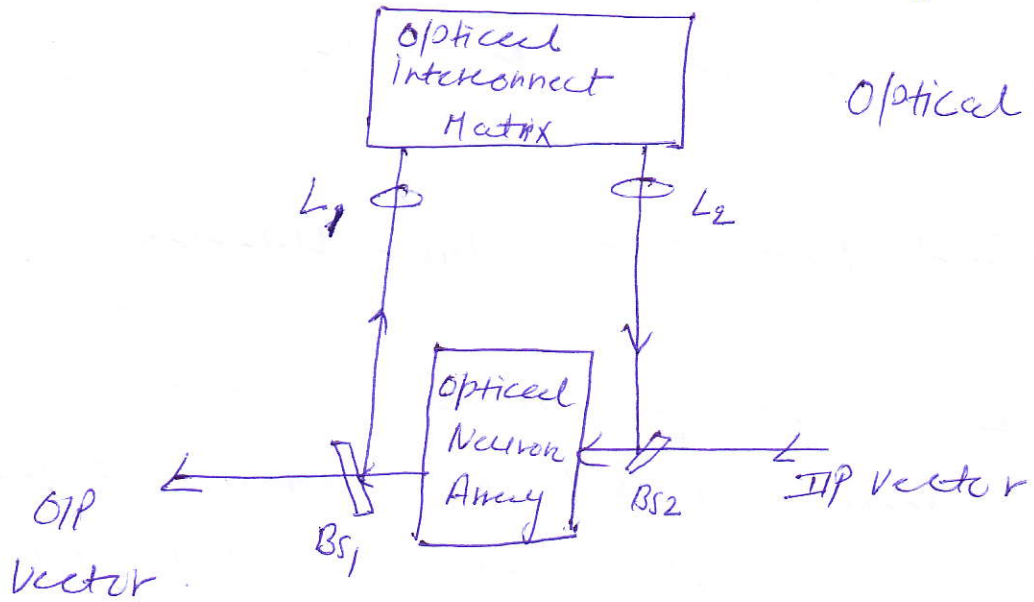
The first hologram contains several stored images. The image then gets correlated with each of them, that produces pattern of light. The brightness of the patterns varies with the degree of correlation. The projected image from lens 2 and mirror 1 passes through pin hole array, at which they are spatially seperated.

From this array, 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 the threshold device.

An optical Hopfield Net Using Volume Holograms:- An all optical occurrent neural n/w using volume holograms has been reported by stroll and Lee. It operates as an implementation of 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 that is most similar, thereby functioning as an optical associative memory.

Fig 1 shows a system with resonant l/p encloses the optical neuron array, the optical interconnect pass around this feedback l/p in the direction indicated by the arrows, being amplified in the process. There is a close analogy here to the operation of the Hopfield n/w. The optical neuron array sums the input and the feedback signals, and then applies the sigmoidal f^n; the optical interconnect matrix performs the vector matrix

when an input vector is applied at the right, it passes through beam splitter BS_2 to the optical neuron array.



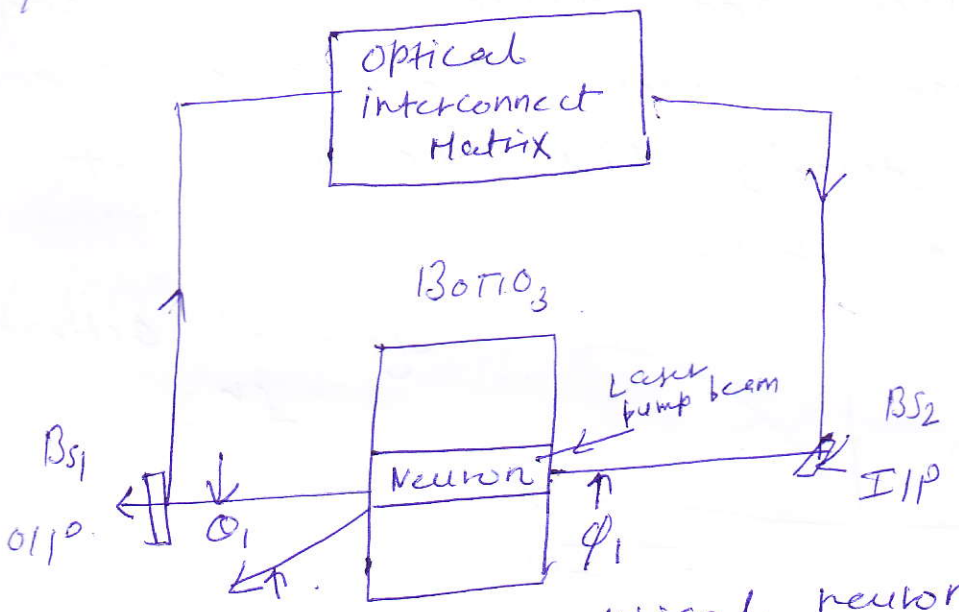
Optical Hopfield N/W.

Here it is amplified, and a sigmoidal f^n is applied by a saturating two beam amplifier. The 'squashed' o/p vector is partially deflected by beam splitter BS_1 to lens L_1, where it enters the optical interconnect matrix. A portion of the o/p light also passes through BS_1 and constitutes the system o/p.

The optical interconnect matrix consists of two volume holograms that store the reference images as diffraction patterns written by laser beams. These serve to weight the input components and direct each weighted sum to the correct element of the optical o/p vector.

The optical Neuron:- Fig shows the Construction of a typical element in the optical neuron array. It operates as an optically pumped, two beam saturating amplifier in a crystal of $BaTiO_3$. A laser beam applied at an angle θ interacts with the input beam to produce an amplified replica.

An optical gain of approximately sixty has been achieved through this technique.

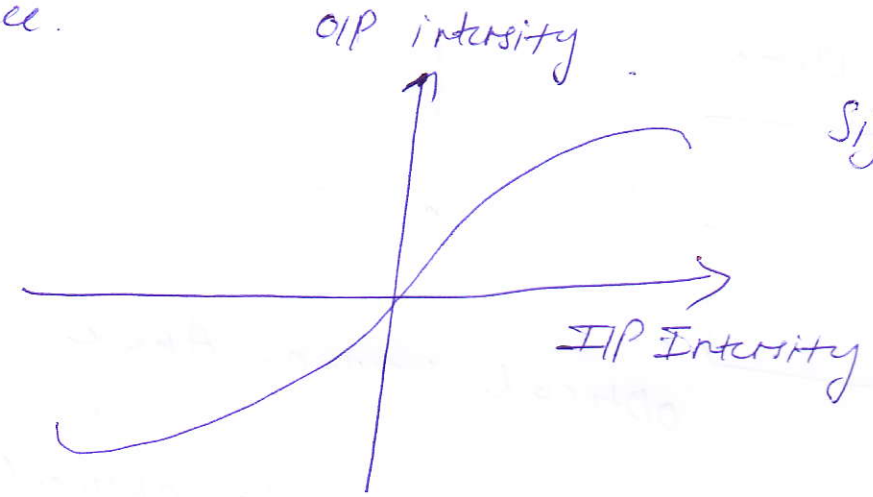


optical neuron Array

Optical interconnect Matrix:- In the optical interconnect matrix, the signal from the optical neuron array passes into an optical system containing two volume holograms. The optical fourier transform of the input is first produced using standard fourier optics techniques.

Then this is applied to the first volume hologram, where the reference vectors are stored in phase-encoded fourier space. The O/P of this hologram is applied to a two beam optical amplifier similar to that of the optical neuron, but operated in a nonsaturating mode.

The inverse fourier transform of the amplified O/P is then produced optically and optically and applied to the second volume holograms; where the same reference images are stored, this time in object space.



Sigmoidal activation f^n.

The O/P of the system then superposition of the vector matrix products b/w the input vector and the stored reference vectors. This optical pattern emerges from the interconnect matrix and is applied to optical neuron array to close the feedback loop.

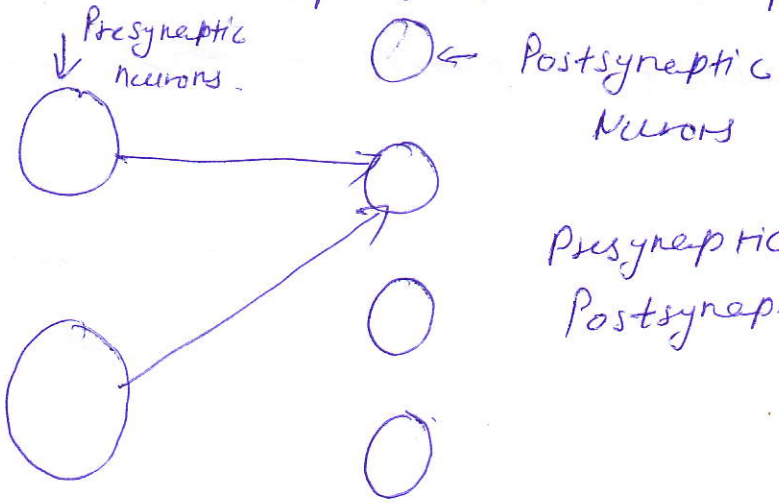
The Cognitron:- Building on current knowledge of brain anatomy and physiology

Fukushima has developed the Cognitron, a hypothetical mathematical model of the human perceptual system.

Structure:- The cognitron is constructed of layers of neurons connected by synapses. As shown in fig, a presynaptic neuron in one layer feeds a postsynaptic neuron in the next layer. There are two types of neurons: excitatory cells, which tends to cause the postsynaptic cell to fire, and inhibitory cells, which tend to cause the postsynaptic cell to fire. and inhibitory cells, which reduce this tendency.

The firing of a neuron depends upon the weighted sums of its excitatory and inhibitory inputs, however, the actual mechanism is more complex than simple summation.

Fig 1.

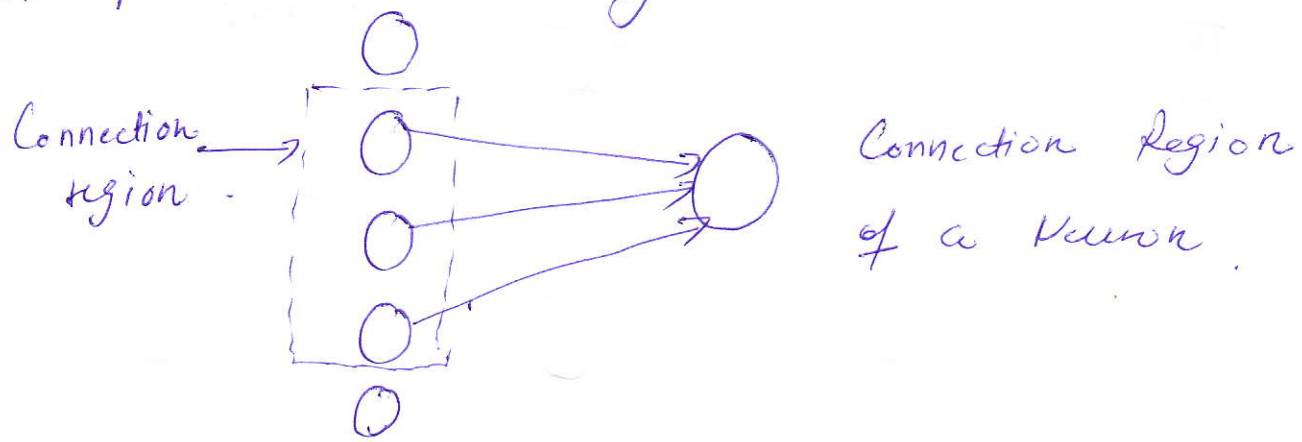


Presynaptic and Postsynaptic Neurons.

Fig 2 shows the each neuron connects only to neurons in the nearby area, called connection weights. The limited range is consistent with the anatomy of visual cortex, where connections are seldom made b/w neurons further than one millimeter apart. In Fukushima's model, neurons are arranged in layers, with the conn^n from one layer going to the next.

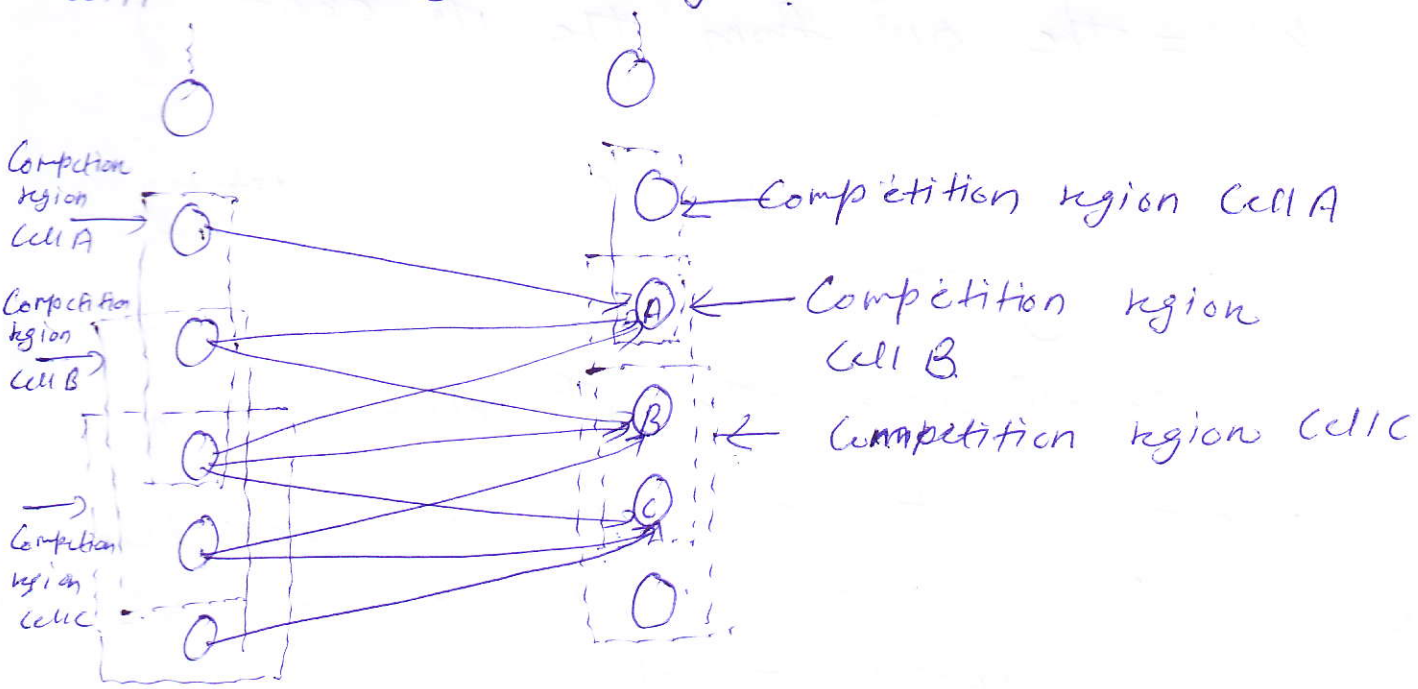
Training:- Beco Z fukushima implemented the cognitron as a multilayer network he was obliged to face the perplexing training problems associated with the structure. He rejected supervised training as biologically implausible, using instead an algorithm that trains without a teacher.

Given a training set of I/P patterns, the network self-organizes by adjusting its synaptic strengths. There are predetermined O/P patterns representing the desired response



Yet, the n/w adjusts itself to recognize patterns on its input with remarkable accuracy.

In fig 3 that the Connection regions of nearby Cells have Considerable overlap; Thus there is a tendency for groups of cells to have similar response patterns. This wasteful duplication of f^n is avoided by incorporating Competition among nearby cells. Even if cells starts out to have identical responses, minor variations will occur; One cell in a Completion region with usually respond more strongly Than its suppresses the firing of nearby cells. and only its synapses will be reinforced; Those of its neighbors will remain unchanged.



Connection and Competition Regions.

The excitatory Neuron: The O/P of the excitatory Cognitron neuron is determined by the ratio of its excitatory inputs to inhibitory inputs.

The total excitatory input to a neuron E is simply the weighted sum of the I/Ps from the excitatory neurons in the previous layer. Similarly, the total inhibitory input I is the weighted sum of the inputs from the inhibitory neurons. In symbols.

$$E = \sum_i a_i u_i$$

$$I = \sum_j b_j v_j$$

where a_i = the weight of the ith excitatory synapse.

u_i = the O/P from the ith excitatory neuron.

b_j = the weight of the ith inhibitory synapse.

v_j = the O/P from the ith inhibitory neuron

Note that weights takes on only +ve value

The O/P of a neuron is then calculated as:

$$NET = [(1+E)/(1+I)] - 1$$

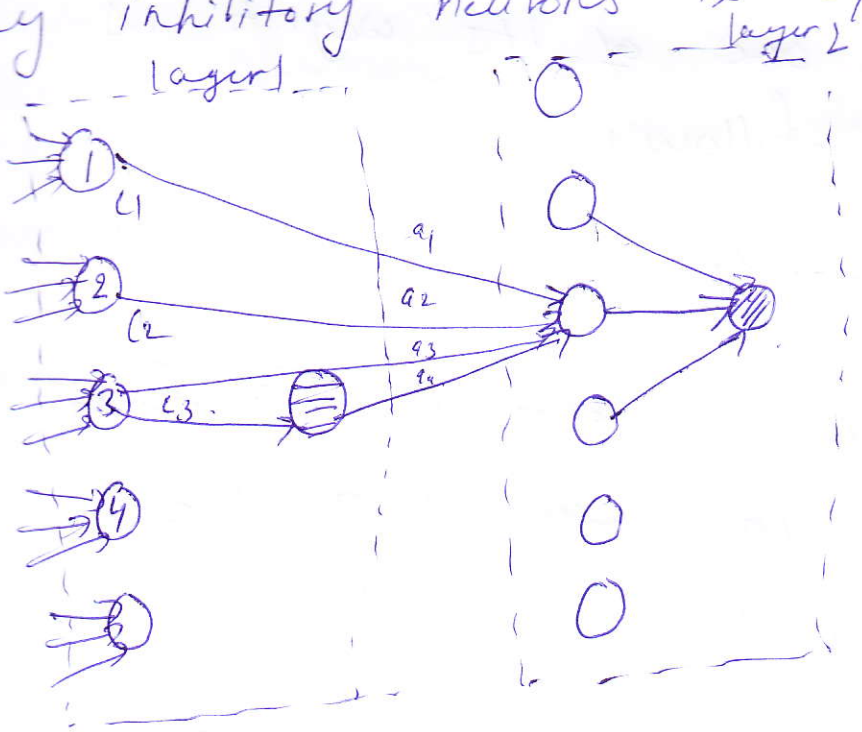
$$OUT = NET \text{ for } NET >= 0$$

$$OUT = 0 \text{ for } NET < 0$$

Assuming +ve values for NET, as

$$OUT = (E-I)/(1+I)$$

The inhibitory Neuron:- In the Cognitron, a layer consists of both excitatory and inhibitory cells. As shown in fig, a layer 2 neuron has a connection region over which it has synaptic conn^ns to a set of layer 1 neuron o/Ps. Similarly in layer 1, there is an inhibitory neuron with the same conn^n region. Synaptic weights coming into inhibitory cells are not modified during training. Their weights are preselected so that the sum of weights into any inhibitory neurons is equal to one.



Cognitron Connection layers

With this detection, inhibitory cell INHIB is simply the weighted sum of its I/Ps which in this case is the arithmetic mean of excitatory O/Ps to which it connects. Hence.

$$INHIB = \sum_i c_i \, out$$

when

$\sum_i c_i = 1$

c_i = inhibitory weight i

Training Procedure:- The weights associated with an excitatory neuron are adjusted only when it is firing more strongly than any of the neighbouring cells in its competition region. When this is so the change in one of its weights is calculated as follows.

$$\delta a_i = q \, c_j \, u_j$$

when.

c_j = the inhibitory weight coming from ~~to~~ neuron j in layer l to the inhibitory neuron i

u_j - the O/P of neuron j in layer l

a_i = excitatory weight i

q = the learning rate coefficient.

Necognitron: We human beings are able to recognize people whom they know when they see them, from a long distance, Human beings are able to identify letters and numbers in different fonts sizes even if they are distorted or displaced.

This type of recognition is possible by human eye. When Perception was used to recognize patterns, the system failed if the same pattern was either distorted or displaced. It is also failed, if any pattern that was superficially similar to the trained input was presented.

Thus there is a need to develop a n/w which is able to recognise patterns similar to human eye. Then n/w Necognitron was developed, which was successor to the Cognitron. A Cognitron is a self-organising multilayer NN. In this model, each layer receives input from the previous layer and also from units in its own area.

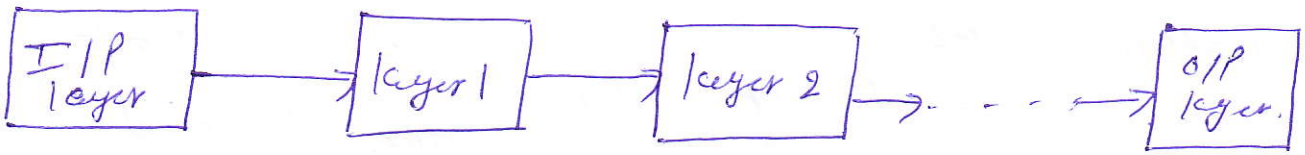
Training on this n/w is based on Completion mechanism.

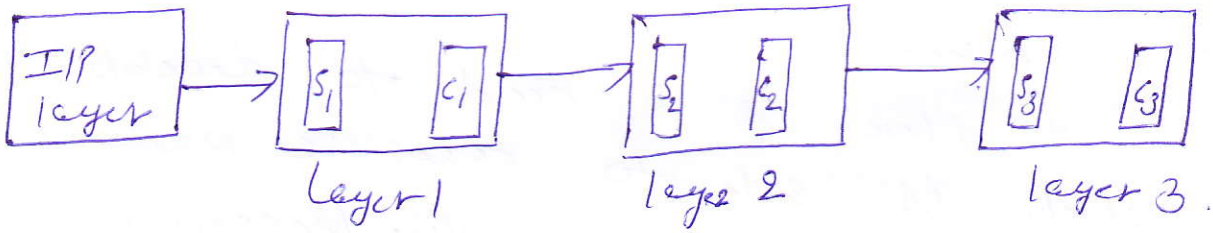
Cognitron could recognize patterns once trained, it failed if the patterns was oriented or dissolved. Thus to overcome this problem, neo-cognitron was developed in 1982.

A Neocognitron is a multilayer feed forward neural n/w model for visual pattern recognition. This n/w is trained using Competitive learning Procedure. It is based on supervised learning.

This model can accept two dimensional patterns like those imaged into the retina processes them in successive layers as that of human visual cortex.

Architecture:-





In this there are various layers, each layer has units arranged in no. of square arrays. Very limited no. of signals are transmitted from one unit to other.

Layers are arranged in pairs.

S-Cells: Simple Cells:. The S arrays are trained to a particular pattern or group of pattern. All cells present in S cells respond to the same pattern. Each S cell is sensitive to a restricted

area by the input pattern, which is called its receptive range. The respective ranges of the cells overlap to cover the entire input pattern for that layer.

C-Cells : Complex Cells :- The C-cells combines the O/P from S-cells.

C-cells make the system less sensitive to the position of the patterns in the input field. The C-cells receive O/P from a set of S cells. The S-cells cover a range called as respective range. Each layer of complex cells responds to the layer range of the input pattern than that in the preceding layer.

The units in each layer are arranged in several sq. cells. The size of the cells may be given as

19x19, 7x7, 16x16, 3x3 depend on pattern used.

Topology :- In this n/w we have various module. Each module consists of two layers.

- A layer of S-cells
- A layer of C cells.

S-Cells receives the input from the previous layer, while C-cells receive the input from the S-layer.

Inputs of the S-layer can be modified during training. The S-layer is designed to detect specific features and the complexity of the features increases as we go higher up the hierarchy.

Each node in the C-layer corresponds to one relative position independent feature. This node receives the input from a subset of S-layer nodes for that feature.

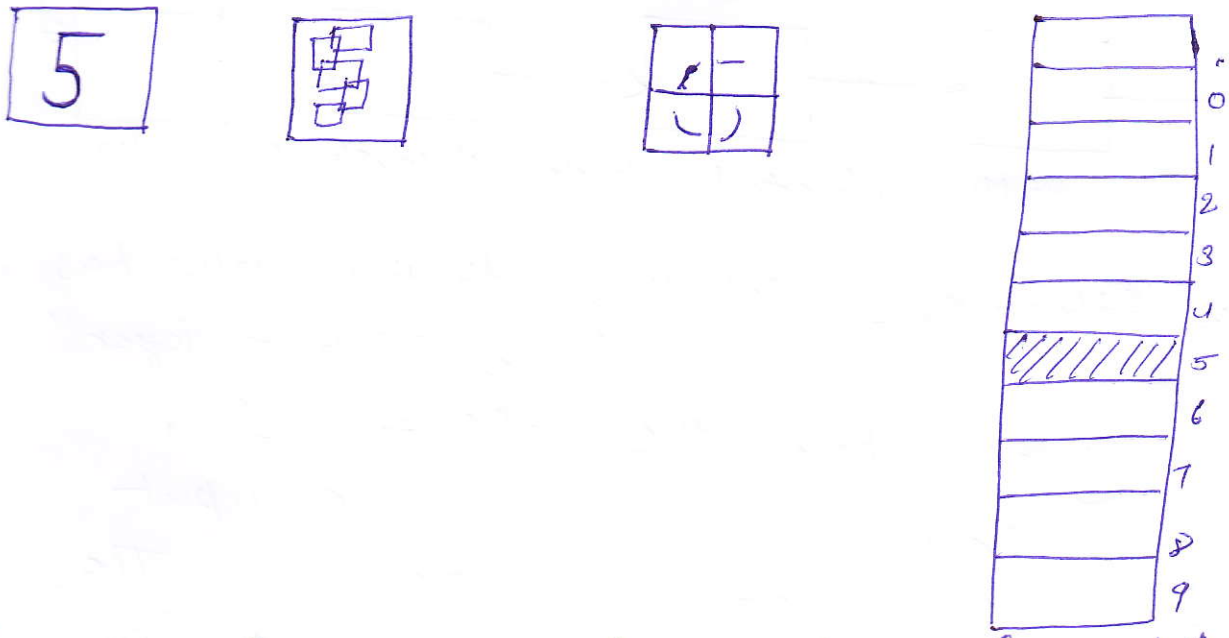
Module-1 closer to input layer is trained first or before module-2 and module-3.

The 'Respective field' of each C node must be fixed by the arr before training. This is beco3 the inputs to the C-layer can't be modified.

The lower level modules have smaller receptive field while the higher level modules represent complex position independent features that depend on the detection of simpler features in the hidden layer.

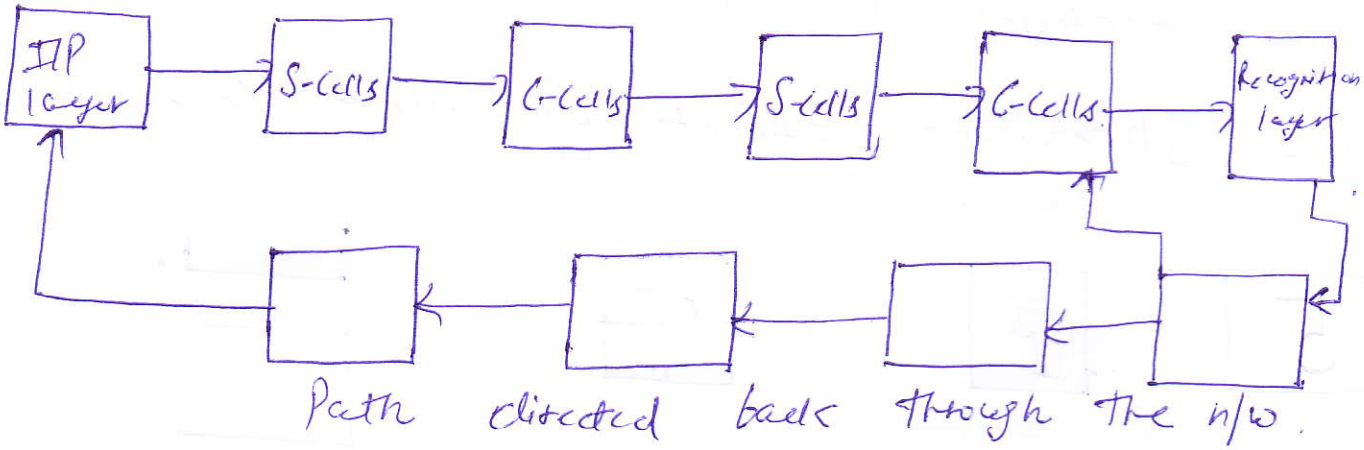
The lower level modules have smaller receptive fields while the higher level modules represent complex position independent features that depend on the detection of simpler features in the hidden layer.

Eg:- Spreading Effect by using Neocognitron.



Selective Attention:- Neocognition can recognize a particular pattern from a group of pattern. This model will focus on a given pattern. This model will focus on a given pattern, segment it from others and then recognize it. After that, attention is then switched on to the next pattern & the process is repeated until all the patterns are recognized.

Such a model is found to recognized an imperfect pattern and recall the Complete pattern with the defects and noise removed.



Selective Attention :- Selective attention n/w has an "Active Input layer"

Reverse signals have the effect of suppressing those portions of the input image that did not contribute to the activation of the currently active node in the previous layer.

To shift the attention of the n/w to another pattern, it is necessary to suppress the active node and allow some other node to become active. That node then reinforces its O/P by using reverse inhibitory signals.

By doing this, all the pattern in the given sample can be extracted.

Algo Calculations :- The S-type cell passes excitatory signals received from the units in the previous layer and passes, inhibitory signals obtained within the same layer.

$$V = \sqrt{\sum \sum t_1 c_i^2}$$

$t_1 \rightarrow$ fixed wt.

$c_i \rightarrow$ O/P from C unit.

S- unit has its scaled I/P as :-

$$x = \frac{1+e}{1+Vw_0} - 1$$

where $e = \sum c_i w_i$

$w_i \rightarrow$ wt's adjustable from C to S.

$w_0 \rightarrow$ " " b/w V & S

$e \rightarrow$ Excitatory input from C units.

Activation of O/P signals is.

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

Net I/P of C-layer is

$$C_{in} = \sum s_i x_i$$

$s_i \rightarrow$ O/P from S-unit.

$x_i \rightarrow$ fixed wt from S-unit to C unit.

Output Is

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

'a' → Is a parameter that depends on the level of the n/w performance.

Training:- The recognition net is trained layer by layer. The S-Cells only have adjustable wts. These wts connect to C-cells in the previous layer by modifiable synaptic wts, which are adjust during training process.

Few wts are inhibitory, which causes increases the o/p, few are excitatory. The S cells responds to a set of C-cells within its <u>Receptive Range</u>

The wts of the Inhibitory cell are not trained, they are selected so that the cells responds to the average of all cell o/Ps to which it connects. The single inhibitory wts from the inhibitory cells to the S-cell is trained like any other synapse.

The desired response of each layer may be chosen. The wts were then adjusted using conventional 2-layer training methods to produce the desired response. The I/P layer wts have to be adjusted to recognize line segments in various orientation similar to the first layer processing in the visual system.

Genetic Algorithms: Introduction: Genetic algorithms are the algorithms that dictate how population of organisms should be formed, evaluated, & modified.

→ for eg: There is a genetic algorithm that determines how to select organisms for sexual reproduction and another that will determine which organims will be deleted from population.

→ The problems that may be solved by genetic algorithms vary from a variety of data mining techniques such as neural nets to optimization of negotiation strategies for oil rights.

→ The trick is to determine how to convert proposed soln to real world problem into simulated genetic material on a computer

Elements of Genetic Algorithm:-

→ Population of Chromosomes
→ Selection according to fitness
→ Crossover to produce new offspring
→ random mutation of new offspring

→ The chromosomes in a GA population typically take form of bit strings. Each locus in the chromosomes has two possible alleles 0 & 1.

→ Each chromosome can be thought of as a point in the search space of candidate sol^ns. The GA processes population of chromosomes successively replacing one such population with another.

→ The GA often required a fitness f^n that assigns a score to each chromosome in current population. The fitness depends on how will the chromosome solve the problem at hand.

(a) Example of fitness f^n:- Suppose we want to minimize a real valued one dimensional f^n.

$$f(y) = y + |\sin(32y)|$$

Here The Candidates Sol^{n}s are value of y which can be encoded as bit strings representing real no. The fitness calculation translates a given bit string X into real no. y & Then Calculates f^n at That value.

(b) GA operators:- The simplest form of Chromosomes in The GA

3 operators.

→ Selection:- This operator Selects chromosomes in population for reproduction.

→ Cross over:- This operator randomly chooses a locus & exchanges the subsequences before and after that locus b/w two chromosomes to create two offspring.

for eg:- The strings 10000100 and 11111111 Could be crossed over after Third locus in each to produce two offstring. 10011111 and 11100100.

→ Mutation:- This operator randomly flips some of the bits in a chromosome. for eg. 00001100 might be mutated in its second position to yield 01000100

A Simple Genetic Algorithms :- A simple GA works as follows.

1). Start with a randomly generated population of n-bit chromosomes.

2) Calculates the fitness $f(x)$ of each chromosome x in the population

3). Repeat following steps until n-offsprings have been created

(a) Select a pair of parent chromosomes from current population, probability selⁿ being an increasing fⁿ, selⁿ is done with replacement.

With Probability P_c, crossover the pair at a randomly chosen point to form two offspring. If no crossover takes space, form two offsprings that are exact copies of the respective parents.

Mutate two offspring at each locus with probability P_m & place resulting chromosomes in new population.

4) Replace current population with new population.

5 Go to step 2.

→ Each iteration of this process is called generation. At the each iteration there one or more chromosomes that fit highly in the population.

→ The procedure described above in the basis for most applications of GA's. The success of the algo. often depends upon details such as size of population probabilities of crossover & mutation.

Consider an example of simple GA. Suppose the string length in 8 that f(x) is equal to the no. of ones in the bit string X that n is 4, $P_c = 0.7$ & that $P_m = 0.001$

Chromosome Label	Chromosome String	Fitness
A	00000110	2
B	11101110	6
C	00100000	1
D	00110100	3

A common selection method in GA's is fitness proportionate selection in which the no. of times an individual is expected to reproduce is equal to the fitness divided by avg. of fitnesses in population.

A roulette wheel is spun, the ball comes to rest on one wedge shaped slice, the corresponding individual is selected.

In the above example, roulette wheel is spun 4 times, the first 2 spins might choose B and D as parents and the next two spins might be B & C to be parents.

→ Once pair of parents is selected, they crossover to produce two offsprings. If they not, then offsprings are exact copies of parents.

→ Suppose B and D cross over after the first bit in string to produce two offsprings, If they not, then offsprings are exact copies of parent E = 10110100 & F = 01101110 & Parents B & C do not cross over instead forming of spring that are exact copies of B & C. Next each off spring is subject to mutation at each locus with probability P_m.

Suppose off spring E is mutated at sixth locus to form E' = 10110000 & off spring B is mutated at first locus to form B' = 01101110. The population will be

Chromosome Label	Chromosome string	Fitness
E'	10110000	3
F	01101110	5
C	00100000	1
B'	01101110	5.

Thus in the population the best string is lost but the avg. fitness rose from 12/4 to 14/4.

Working of Gentic Algorithm:- Steps are:-

1). Create an initial population of chromosomes.
2) Evaluate the fitness of each chromosomes will mate.
3). Based on this fitness, selected chromosomes to produce the off spring
4). Cross over or mate the selected chromosomes to produce the kid off spring
5). Mutate some of genes of chromosomes.
6). Repeat steps 3 through 5 until new population is created.
7). Algorithm ends when best soln. has not changed for present no. of no. of generations.