SOFT COMPUTING

Subject Code:18KP3CS10

A.Thirumalairaj
Department of Computer Science

Unit-I

•Artificial Intelligence-AI problems-The Underlying Assumption-AI Techniques

The Level of Models-Criteria of Success-Heuristic Search Techniques-Generate and test-Hill Climbing-Best-first-search-Problem Reduction-Constraint satisfaction

Means –ends Analysis.

Unit-II

Introduction of (Artificial Intelligence)Systems-Neural Networks(NN)-FuzzyLogic -Genetic Algorithm- Fundamentals of the NN Basic concepts-Human Brain-

Model of Artificial Neuron-NN Architectures-characteristics of NN- Learning Methods

Taxonomy of NN Architecture- History of NN Research –Early NN Architectures-

Application Domain

Artificial Intelligence

- Introduction:
- Father of Artificial intelligence is John McCarthy
- What is Artifical Intelligence?:
- Artificial Intelligence is the study of how to make computers do things, which, at the moment, people do better.
- Artificial Intelligence, John McCarthy, it is “The science and engineering of making intelligent machines, especially intelligent computer programs”.
- A computer programming controlled a robot, for human think.
- Artificial intelligence is based on the principle that human intelligence can be defined in a way that a machine can easily mimic it and execute tasks, from the most simple to those that are even more complex. The goals of artificial intelligence include learning, reasoning, and perception.

AI PROBLEMS

- Problems in AI early focused in on formal task with Applied Mathematical Theorem
- Another Early day focused on sort the problem is called Commonsense reasoning.
- These include perception natural language understanding, and problem solving in Specialized domains medical diagnosis and chemical analysis.

Now days AI problems and solution techniques it is important to discuss the following Question:

- 1. What are the underlying assumptions about intelligence?
- 2. What kinds of techniques will be useful for solving AI problems?
- 3. At what level human intelligence can be modelled?
- 4. When will it be realized when an intelligent program has been built?

The Underlying Assumption

- Physical Symbol
- A Physical Symbol entities called symbol.
- A physical symbol is machine that produces through time on evolving collection of symbol structures.
- Computer provide the perfect of medium for this expermination can be programmed to physical symbols
- A physical Symbol include a visual perception its influence of sub symbolic process.
- Sub symbolic models are beginning to challenge symbolic ones at such low level tasks.
- The important of the physical symbol it’s a significant theory of the nature of human Intelligence and great interest to psychologists.
- That its possible to build programs that can perform the intelligent task performed by people

AI TECHNIQUES

AI Techniques:

Ai techniques is a method that exploits knowledge the should be represent several way

Its inculding:

- It is voluminous
- Its is hard to characterize accurately
- Its is constantly changing
- Its differ from data by organized.

AI techniques is a method that exploits knowledge represented

- The Knowledge captures' generalization. Amount of memory and updating will be required.so we call this property ‘data’ rather than knowledge.
- A bulk of the data can be acquired automatically.
- It can easily be modified to correct errors and to reflect changes in the world
- Its can be used in great many situations even if its not totally accurate or complete.

AI TECHNIQUES PROBLEMS

- Tic-Tac-Toe
- The Tic-Tac-Toe game consists of a nine element vector called BOARD; it represents the numbers 1 to 9 in three rows.

1 2 3

4 5 6

7 8 9

An element contains the value 0 for blank, 1 for X and 2 for O.

A MOVETABLE vector consists of 19,683 elements (39) and is needed where each element is a nine element vector.

The algorthim:

1. View the vector board as a ternary number. Convert it to a decimal number.
2. Use the number computed in step1 as in index into movetable and access the vector stored there.
3. The vector selected in step 2 represents the way the board will look after the move that should be make so set Board equal to the vector

Comments

1. Lot of space to store the table that specifies the correct move to make from each board
2. Lot of work specifying all the entries in the movetable
3. Movetable enteries determined and entered without any errors.

THE LEVEL OF MODELS

- Artifical Intelligence Model
- What is our goal in trying to produce programs that do the intelligent things that people do?'
- Are we trying to produce programs that do the tasks the same way that people do? OR Are we trying to produce programs that simply do the tasks the easiest way that is possible?
- . AI techniques is a search method .To use the knowledge about the objects involved in the problem area and abstraction
- To allows the element of pruning to occur, and to enable a solution to be found in real time.
- Examples : EPAM (Elementary Perceiver and Memorizer) which memorized garbage syllables.
- The second class of problems attempts to Human performance
- 1. To test psychological theories of human performance. Ex. PARRY [Colby, 1975]which exploited a model of human paranoid behaviour to simulate the conversational behavior of a paranoid person.
- 2. To understand computer reasoning. In many circumstances, people are reluctant to rely on the output of a computer unless they can understand how the machine arrived at its result.
- 3. To exploit what knowledge we can glean from people. Since people are the best-known performers of most of the tasks with which we are dealing, it makes a lot of sense to look to them for clues as to how to proceed.

CRITERIA FOR SUCCESS

- T*uring Test*.
- In this method 1950, Alan Turing proposed the a machine can think. This method has become known as the T*uring Test*.
- *Ex:* One person plays the role of the interrogator, who is in a separate room from the computer and the other person. The interrogator can ask questions of either the person or the computer by typing questions and receiving typed responses.
- Interrogator: ln the first line of your sonnet which reads “Shall l compare thee to a summer’s day,” would not “a spring day” do as well or better?
- A:It wouldn’t scan.
- Interrogator :How about “a winter’s day.” That would scan all right. Yes, but nobody wants to be compared to a winter’s day.
- Interrogator:Would you say Mr. Pickwick reminded you of Christmas?
- A:In a way.
- Interrogator:Yet Christmas is a winter’s day, and l do not think Mr. Pickwick would mind the comparison.
- A:I don’t think you’re serious. By a winter’s day one means a typical winter’s day, rather than a special one like Christmas.
- It will be a long time before a computer passes the Turing test.

HEURISTIC SEARCH TECHNIQUES

- Heuristic is a mathematical optimization techniques.
- *Heuristic* is a technique that improves the efficiency of a search process, possibly by sacrificing claims of completeness.
- To construct special-purpose heuristics that exploit domain-specific knowledge to solve particular problems.
- Well-designed heuristic functions can play an important part in efficiently guiding a search process toward a solution
- sometimes a high value of the heuristic function indicates a relatively good position
- At other times a low value indicates an advantageous situation (Ex: traveling salesman)
- AI research techniques:

- Generate-and-test
- Problem reduction
- Best-first search
- Means-ends analysis
- Hill climbing
- Constraint satisfaction

GENERATE-AND-TEST

- Algorithm: Generate-and-Test
- The heuristic function is sometimes also called the *objective function*, particularly in the literature of mathematical optimization.
- Generate a possible solution. For some problems, this means generating a particular point in the problem space. For others, it mearts generating a path froin a start state.
- Test to see if this is actually a solution by comparing the chosen point or the endpoint of the chosen path to the set of acceptable goal states.
- If a solution has been found, quit. Otherwise, return to step 1.
- The generate-and-test algorithm is a depth-first search procedure since complete solutions must be generated before they can be tested.
- The most straightforward way to implement systematic generate-and-test is as a depth-first search tree with backtracking.
- The most straightforward way to implement systematic generate-and-test is as a depth-first search tree with backtracking.

HILL CLIMBING

- Hill climbing is used good heuristic function is available for evaluating states.
- Simple Hill Climbing

Algorithm: Simple Hill Climbing

1.Evaluate the initial state, If it is also a goal state. then return it and quit. Otherwise, continue with the initial state as the current state.

2. Loop until a solution is found or until there are no new operators left to be applied in the current state.

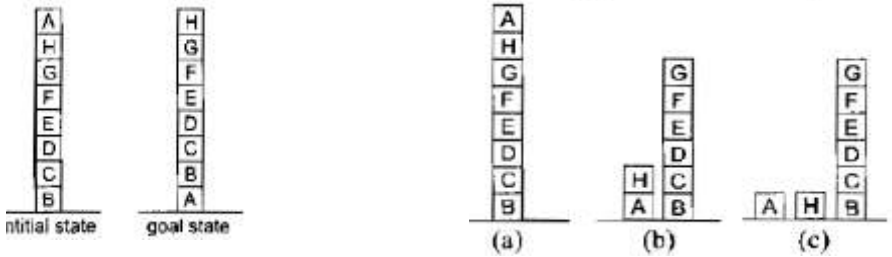
   - » Select an operator that has not yet been applied to the current state and apply it to produce a new state.
   - » Evaluate the new state.
     - If it is a goal state, then return it and quit.
     - If it is not a goal state but it is better than the current state, then make it the current state.
     - If it is not better than the current state, then continue in the loop

- The key difference between this algorithm and the one we have for generate-and-test is the use of an evaluation function as a way to inject task-specific knowledge into the control process.

Steepest-Ascent Hill Climbing

- simple hill climbing considers all the moves from the current state and selects the best one as the next state. This method is called Steepest-Ascent Hill Climbing or *gradient search*

- **Algorithm: Steepest-Ascend Bill Climbing**
- 1. Evaluate the initial state. If if is also a goal state, then return it and quit. Otherwise, continue with the initial state as the current state.
- 2. Loop until a solution is found or until a complete iteration produces no change to current state:
- (a) Let SUCC be a state such that any possible successor of the current state will be better than SUCC.
- (b) For each operator that applies to the current state do:
- (i) Apply the operator and generate a new state.
- (ii) Evaluate the new state. If it is a goal state, then return it and quit. lf not, compare it to SUCC.
- If it is better, then set SUCC to this state. If it is not better, leave SUCC alone.
- (c) If the SUCC is better than current state, then set current state to SUCC.
- Ex: Coloured Block Problems

- Ex: Blocks world problem
- 

- Using this function, the goal state has a score of 8. The initial state has a score of 4
- Hill climbing will halt because all these states have lower scores than the current state. The process has reached a local maximum that is not the global maximum.
- The problem is that by purely local examination of support structures. the current state appears to be better than any of its successors because more blocks rest on the correct objects
- . To solve this problem to disassemble a good local structure
- This new heuristic function captures the two key aspects of this problem: incorrect structures are bad and should be taken apart and correct structures are good and should be built up.
- A heuristic function that does this conveits the local hill-climbing procedure into a global method by embedding a global method.

Simulated Annealing

- Simulated annealing [Kirkpatrick 1953] its a computational process is patterned after the physical process *of annealing*, in which physical substances such as metals are melted and then gradually cooled until some solid state is reached, The goal of this process is to produce a minimal-energy final state.

- $p=e^{-\Delta E/kT}$

A£ Postive Change,T-Tempature,k-Boltzmann's constant,

The probability of a large uphill move is lower than the probability of a small one.
During the beginning of the process when the temperature is high. and they become less likely at the end as the temperature becomes lower. this process is that downhill moves are allowed anytime.

The rate at which the system is cooled is called the *annealing schedule.*
These properties of physical annealing can be used to define an analogous process of simulated annealing

$$p'=e^{-\Delta/T}$$

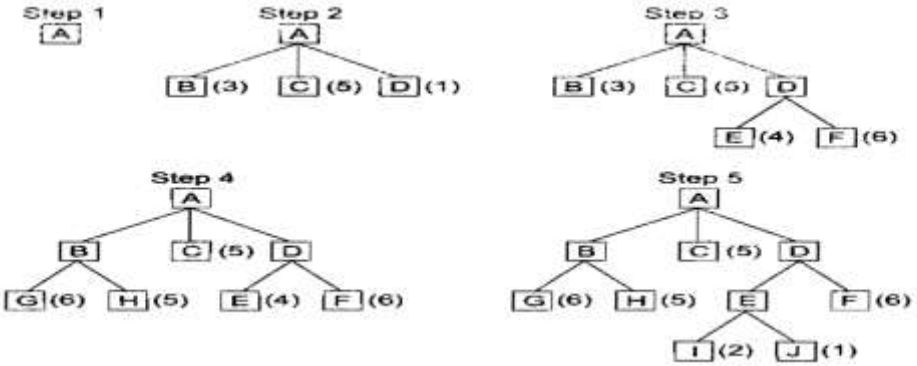
- The annealing schedule must be maintained.
- Moves to worse states may be accepted.
- It is a good idea to maintain, in addition to the current state, the best state found so far. Then, if the final state is worse than that earlier state (because of bad luck in accepting moves to worse states), the earlier state is still available.

- ***Algorithms Simulated Annealing***
- Evaluate the initial state. If it is also a goal state, then return it and quit. Otherwise, continue with the initial state as the current state.
- Initialize *BEST-SO-FAR* to the current state.
- Initialize *T* according to the annealing schedule.
- Loop until a solution is found or until there are no new operators left to be applied in the current state.
  - Select an operator that has not yet been applied to the current state and apply it to produce a new state.
  - Evaluate the new state. Compute
- ΔE = (value of current) — (value of new state)
- » If the new state is a goal state, then return it and quit.
    - If it is not a goal state but is better than the current state, then make it the current state. Also set
- *BEST-SO-FAR* to this new state.
    - lf it is not better than the current state, then make it the current state with probability *p'* as defined above. This step is usually implemented by invoking a random number generator to produce a number in the range [0,1]. If that number is less than y', then the move is accepted. Otherwise, do nothing.
  - Revise *T 'us* necessary according to the annealing schedule.
- Return *BEST-SO-FAR*, as the answer.

BEST-FIRST SEARCH

- OR Graphs
- It is a problem solving algorithm to decomposing to smaller problem.
- *OR graph* each of its branches represents an alter native problem-solving path.
- To implement the graph-search procedure to two listed nodes
- Open: Its actually a priority queue in which the elements with the highest priority are those with the mosl promising value of the heuristic function.
- Closed: A new node is generated to check it has been generated before.

•Sum of two Components: call g and h'
•g=intiate state of the current node
•H'-Additional cost of the current node to goal state.
•Steps: It generates a node that corresponds to a goal state.
•It picks the most promising of the nodes that have so far been generated but not expanded.
•It generates the successors of the chosen node, applies the heuristic function to the list of open nodes, After checking the node before generated



Algorithm.- Best-First 5earch

Start with *OPEN* containing just the initial state.

Until a goal is found or there are no nodes left on *OPEN* do:

a)Pick them best node on *OPEN*.

b)Generate its successors.

For each successor do:

(i) If it has not been generated before, evaluate it, add it to *OPEN*, anal record its parent.

(ii) If it has been generated before, change the parent if this new path is better than the Previous one. In thut case, update the cost of getting to this node and to any successors that this ntide may already. have.

A* Algorithm

- Algorithm: A*
- A* Search algorithm is used to path-finding and graph traversal
- To find all the paths that are available to us from the source to the destination.
- The best path that can be taken from its current state .
- **Algorithm**
- Add start node to list
- For all the neighbouring nodes, find the least cost F node
- Switch to the closed list

– For 8 nodes adjacent to the current node

– If the node is not reachable, ignore it. Else

- If the node is not on the open list, move it to the open list and calculate f, g, h.
- If the node is on the open list, check if the path it offers is less than the current path and change to it

if it does so.

•Stop working when

  – find the destination

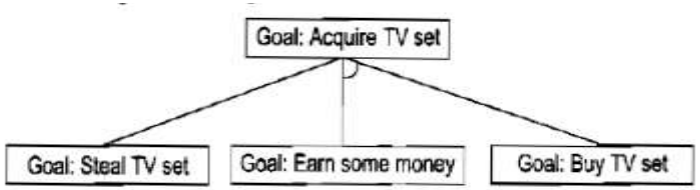
  – find the destination going through all possible points

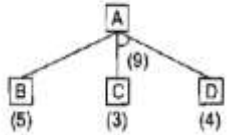
Agenda

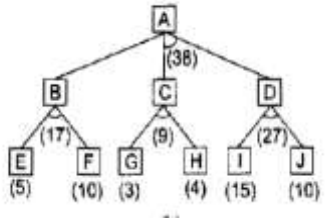
- An agenda is a list of tasks a system could perform.
- Associated with each task there are usually two things: a list of reasons why the task is being proposed and a rating representing the over all weight of evidence Suggesting that the task would be useful.
- until a goal state is reached or the agenda is empty:
  - Choose the most promising task from the agenda. Notice that this task can be represented in any desired form. It can be thought of as an explicit statement of what to do next or simply as an indication of the next node to be expanded.
  - Execute the task by devoting to it the number of resources determined by its importance. The important resources to consider are time and space. Executing the task will probably generate additional tasks (successor nodes). For each ot them, do the following:
    - same reason for doing it is already on its list of justifications., ignore this current evidence. It this justification was not already present, add it to the list. If the task was not on the agenda, insert it
    - Compute the new task's rating, combining the evidence from all its justifications. Not all justifications need have equal weight. It is often useful to associate with each justification a measure of how strong a reason it is. These measures are then combined at this step to produce an overall rating for the task.

PROBLEM REDUCTION

- The AND-OR GRAPH (or tree) is useful for representing the solution of problems that can solved by decomposing them into a set of smaller problems, all of which must then be solved. This decomposition, or reduction, generates arcs that we call AND arcs.
- One AND arc may point to any number of successor nodes, all of which must be solved in order for the arc to point to a solution.
- OR graph, several arcs may emerge from a single node, indicating a variety of ways in which the original problem might be solved. This is called not simply an AND-graph but rather an AND-OR graph.







- Let *G* be a graph with only starting node *INIT.*
- Repeat the followings until INIT is labeled SOLVED or h(INIT) > FUTILITY
- a) Select an unexpanded node from the most promising path from INI*T (call it NODE)*
- b) Generate successors of NODE. If there are none, set h(NODE) = FUTILITY
- *Select an unexpanded node from the most promising path from INIT (call it NODE)*
- b) Generate successors of NODE. If there are none, set h(NODE) = FUTILITY (i.e., NODE is unsolvable); otherwise for each SUCCESSOR that is not an ancestor of NODE do the following:
- i. Add SUCCESSSOR to G.
- ii. If SUCCESSOR is a terminal node, label it SOLVED and set h(SUCCESSOR) = 0.
- iii. If SUCCESSPR is not a terminal node, compute its h
- c) Propagate the newly discovered information up the graph by doing the following: let S be set of SOLVED nodes or nodes whose h values have been changed and need to have values propagated back to their parents. Initialize S to Node. Until S is empty repeat the followings:
- i. Remove a node from S and call it CURRENT.
- ii. Compute the cost of each of the arcs emerging from CURRENT. Assign minimum cost of its successors as its h.
- iii. Mark the best path out of CURRENT by marking the arc that had the minimum cost in step ii
- iv. Mark CURRENT as SOLVED if all of the nodes connected to it through new labelled arc have been labeled SOLVED
- v. If CURRENT has been labeled SOLVED or its cost was just changed, propagate its new cost back up through the graph. So add all of the ancestors of CURRENT to S.
-

AO*Algorithm

- *OPEN* and ***CLOSED,*** that were used in the A* algorithm, the AO* algorithm will use a single structure *GRAPH,* representing the part of the search graph that has been explicitly generated.
- Each node in the graph will point both down to its immediate successors and up to its immediate predecessors. Each node in the graph will also have associated with it an *h'* value, an estimate of the cost of a path from itself to a set of solution nodes
- 1. Let GRAPH consist only of the node representing the initial state. (Call this node INIT.) Computeh'(INIT)
- 2. Until *INIT* is labeled *SOLVED or* until *INIT's h'* value becomes greater than *FUTILITY,* repeat the following procedure:
- Trace lhe labeled arcs from INIT and select for expansion one of the as yet unexpanded nodes that occurs on this path. Call the selected node NODE.
- Generate the successors of NODE. If there are none, then assign FUTILITY as the h' value of NODE. This is equivalent to saying that NODE 'is not solvable. If there are successors, then for each one (called SUCCESSOR) that is not also an ancestor of NODE do the following:
-

- If possible, select from S a node none of whose descendants in GRAPH occurs in S. If there is
- no such node, select any node from S. Call this node CURRENT, and remove it from S.
- (ii) Compute the cost of each of the arcs emerging from CURRENT. The cost of each arc is equal to the sum of the h' values of each of the nodes at the end of the arc plus whatever the cost of the arc itself is. Assign as CURRENT'S new h' value the minimum of the costs just computed for the arcs emerging from it.
- (iii) Mark the best path out of CURRENT by marking the arc that had the minimum cost as computed in the previous step.
- Mark the best path out of CURRENT by marking the arc that had the minimum cost as computed in the previous step.
- (iv) Mark current solved if all of the nodes connected to it through the new labeled arc have been labeled SOLVED.
- (v) If CURRENT has been labeled SOLVED or if the cost of Current was just changed, then its
- new status must be propagated back up the graph. So add all of the ancestors of CURRENT to S
-

CONSTRAINT SATISFACTION

- Constraint satisfaction is a search procedure that operates in a space of constraint sets. The initial state contains the constraints that are originally given in the problem description. A Goal State is any state that has been constrained "enough," where "enough" must be defined for each problem.
- Constraint satisfaction is a two-step proces
- 1. propagation, arises from the fact that there are usually dependencies among the constraints, These dependencies occur because many constraints involve more than one object and many objects participate in more than one constraint.
- 2. constraints is the cryptarithmetic problem, for example, this usually means guessing a particular value for some letter.
- Ex: One constraint, N = E + 1.

  Added the constraint N = 3

  To get a stronger constraint on E.

First, a contradiction may be detected, If this happens, then there is no solution consistent with all the known constraints.

The second possible reason for termination is that the propagation has run out of steam and there are no further changes that can be made on the basis of current knowledge.

- Algorithm: Constraint Satisfaction
- Propagate available constraints. To do this, first set OPEN to the set of all objects that must have values assigned to them in a complete solution. Then do until an inconsistency is detected or until OPEN is empty:
- Select an object OB from OPEN. Strengthen as much as possible the set of constraints that apply
- to OB.
- If this set is different from the set that was assigned the last time OB was examined or if this is the first time OB has been examined, then add to OPEN all objects that share any constraints with Od.
- Remove OB from OPEN.
- If the union of the constraints discovered above defines a solution, then quit and report the solution.
- If the union of the constraints discovered above defines a contradiction, then return failure.
- If neither of the above occurs, then it is necessary to make a guess at something in order to proceed. To do this, loop until a solution is found or all possible solutions have been eliminated:
- Select an object whose value is not yet determined and select a way of strengthening the constraints on that object.
- Recursively invoke constraint satisfaction with the current set of constraints augmented by the strengthening constraint just selected.

MEANS-ENDS ANALYSIS

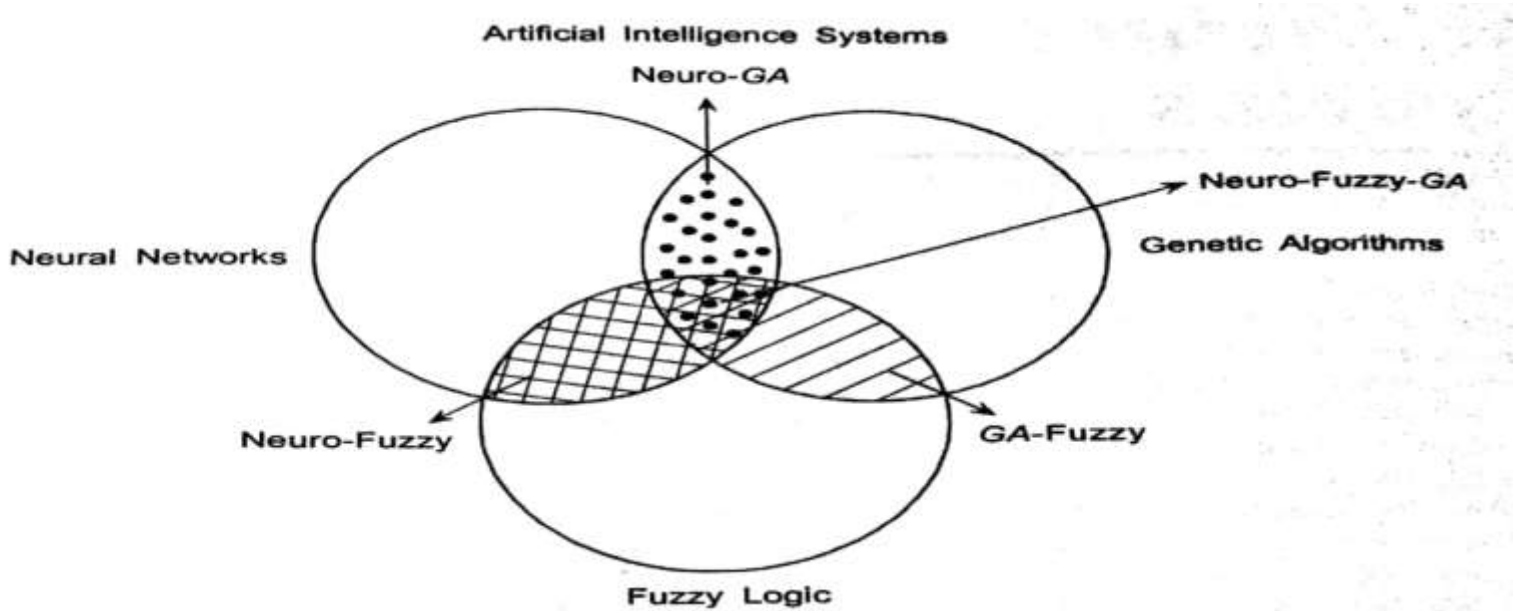
- To solve the major parts of a problem first and then go back and solve the small problems that arise in “gluing” the big pieces together. A technique known as Means-*ends analysis* .
- Operators are selected and then subgoals are set up to establish the preconditions of the operators is called *operator subgoaling*
- Compare ***CURRENT*** to ***GOAL.*** If there are no differences between therri then return.
- Otherwise, select the most important difference and reduce it by doing the following until success or failure is signaled:
  - Select an as yet untried operator 0 that is applicable to the current difference. If there are no such operators, then signal failure.
  - Attempt to appIy *O* to ***CURRENT.*** Generate descriptions of two states: ***O-START,*** **a** state in which 0’s preconditions are satisfied and ***O-RESULT,*** the state that would result if O were applied in 0- ***START.***
  - If
- *(FIRST-PART MEA(CURRENT, O-START))*
- and
- *(LAST-PART MEMO-RESULT, GOAL))*
- are successful, *then* signal success and return the result of concatenating
- *FIRST-PART, 0,a*nd *LAST-PART.*

Neural Networks

UNIT-II

Neural Network

- What is Neural Network?
- Inventor of the first neuro computer, Dr. Robert Hecht-Nielsen
- Define:A computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.”
- .Is a highly interconnected network of a large number of processing elements called neurons in an architecture inspired by the brain. An NN can be massively parallel is called as parallel distributed processing.
- NN architectures have been classified into various types based on their learning mechanisms and other features. Some classes of NN refer to this learning process as training and the ability to solve a problem using the knowledge acquired as inference.



•A fuzzy logic representation founded on fuzzy set to capture way humans represent and reason with real-world Knowledge in the face uncertainly .

•fuzzy set can be defined mathematically by assigning to each possible individual in the *universe of discourse,* a value representing its grade *of membership* in the fuzzy set.

•Fuzzy sets support a flexible sense of member ship of elements to a set

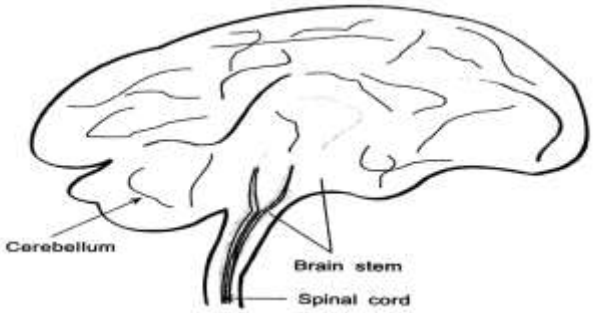
Genetic Algorithms

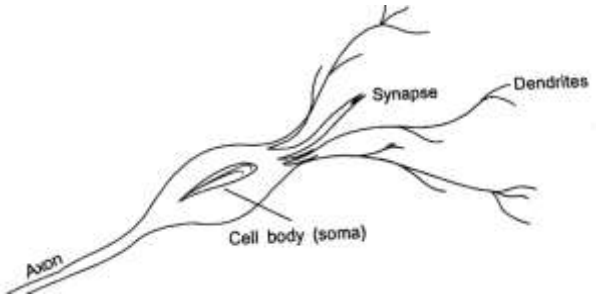
- Genetic Algorithms: Developed by John Hollan
- algorithms are based on the ideas of and random search through a given set of alternatives with finding the best alternatives for using optimization
- Fitness is Defined as figure merit of maximized and minimized .Binary alphabet (0,1) the chromosomes are binary strings and in the case of real alphabet(0-9)

  the chromose are decimal strings.
- The genetic inheritance operators are reproduction, cross over, mutation, inversion,

  Dominance,deletion,duplication,translocation,segregation,speciation,migration,sharing and mating.

Human Brain





•Brain contains about 10" basic units called *neurons.*

•A neurons is a small cell that receives electro –chemical signal from to various sources and turn responds by transmitting electrical impulse to other neurons

Average brain weights 1.5 kg as average neuron has weight of 1.5×10^{-9}

•Soma are irregularly shaped filaments called dendrites

•Soma araies electrical signal cell called a membrane

•The axon terminates in a specialized contact called synapse or synaptic

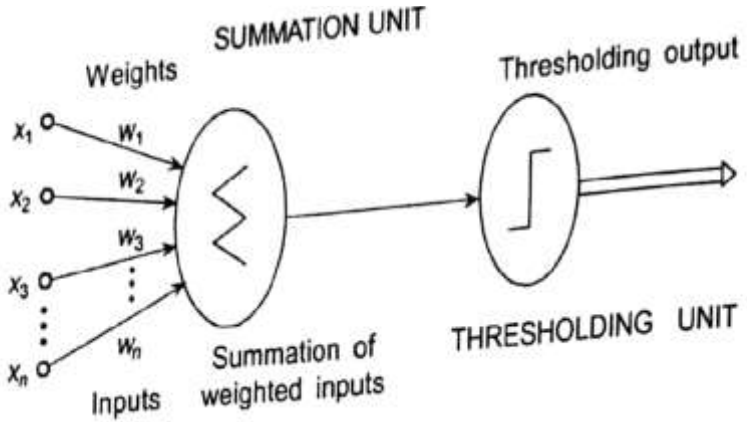
•Synaptic reading the electrical charges to soma.

•Single neuron can many synaptic inputs and synaptic output

•The size of synapse are believed to related to learning

Artificial Neuron

- Biological neuron and hence termed as artificial neuron



- X_1, x_2, x_3 the n inputs to the artificial neuron and weights $w_1, w_2 \ldots w_3$
- Recollect that a biological neuron receives all inputs through the dendrites, sums them tp produces an output if *the* sum is greater than a threshold value.
- Input signals that is modeled by the *weights*
- weights here are multiplicative factors t the inputs to account for the strength of the synapse.

- I received by the artificial neuron.

$$I = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

$$= \sum_{i=1}^{n} w_i x_i$$

To generate the final output *y*, the sum is passed on to a non-linear filter Φ called *function, of Transfer function,* or *Squash function* which releases the output.

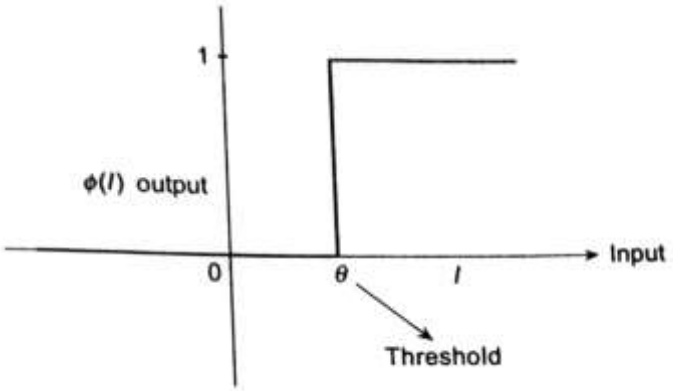
$$y = \phi(I)$$

A very commonly used Activation function is *the Thresholding function*
If I is greater than Θ then the output 1 else is 0.
Where Φ is the step function Heaviside function

$$y = \phi\left(\sum_{i=1}^{n} w_i x_i - \theta\right)$$

$$\phi(I) = \begin{cases} 1, & I > 0 \\ 0, & I \leq 0 \end{cases}$$

- The thresholding function is convenient in the sense that the output signal
- 1 or 0 resulting the neuron being on or off

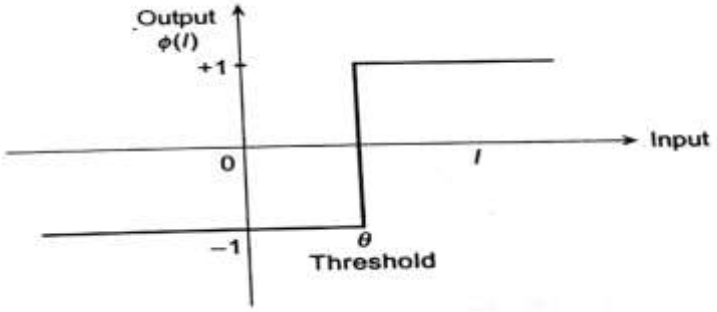


- The other choices for Activation function besides Thersholding function

Signum Function: The Quantizer function ,the function defined as

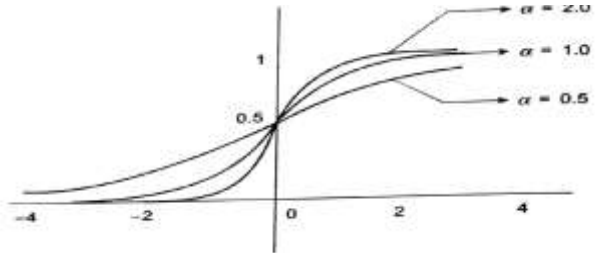
$\Phi(I) = +1 \quad I > \Theta$

$-1 \quad I < \Theta$



Sigmoidal function

This function is a continuous function the varies gradually between the asymptotic values 0 and 1 or -1 and +1



$$\Phi(I) = \frac{1}{1 + e^{-\alpha 1}}$$

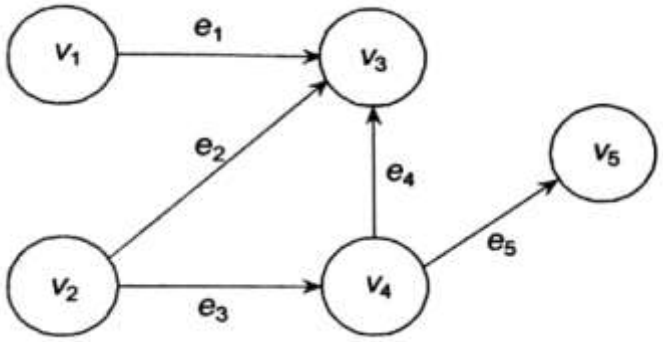
- Hyperbolic Tangent function:
- The function is given by
- $\Phi(I)=\tanh(I)$ produces the negative output values
- This model make as use of bias term whose weight is w_0 but with a fixed input of $x_0 = 1$ input xi and output wi
- The bias is an external parameter for the artifical neuron but serves the purpose of increasing or decreasing the net input of the activation function depending postive or negative

Neural Network Architectures

An Artificial Neural network :

As a data processing system consisting of a large number of simple highly interconnect processing element in an architecture inspried by the structure of the cerebral cortex of the brain.

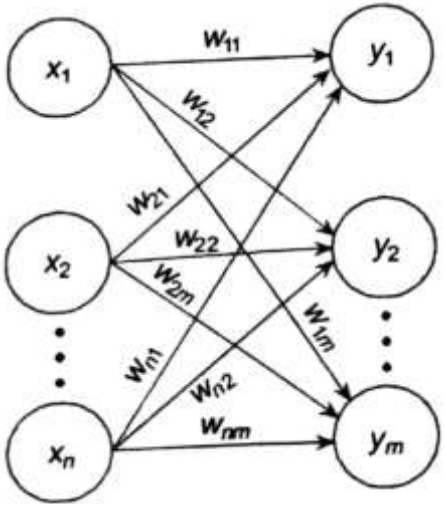
- ANN can be represented using direct graph
- A graph G is an ordered 2-tuple (V,E) consisting of set V vertices and set E edges.
- Each edge is assigned an orientaton the graph is directed and called a direct graph.
- The vertices of the graph may represent neurons (input/output)Synaptic .
- The edges are labelled by the weights attached to the synaptic links.
- Vertices $V = \{v_1, v_2, v_3, v_4, v_5\}$
- Edges $E = \{e_1, e_2, e_3, e_4, e_5\}$



Single Layer Feed Forward Network

- This type of network comprises two layer namely input layer and output layer
- Input layer neurons receive the input signal and output layer receives the output signals.
- Weights connect every input neuron to the output neuron
- Synaptic links carrying the weights connect every input neuron to the output neuron.
- This network is Single layer feed forward Neural Network.

INPUT LAYER

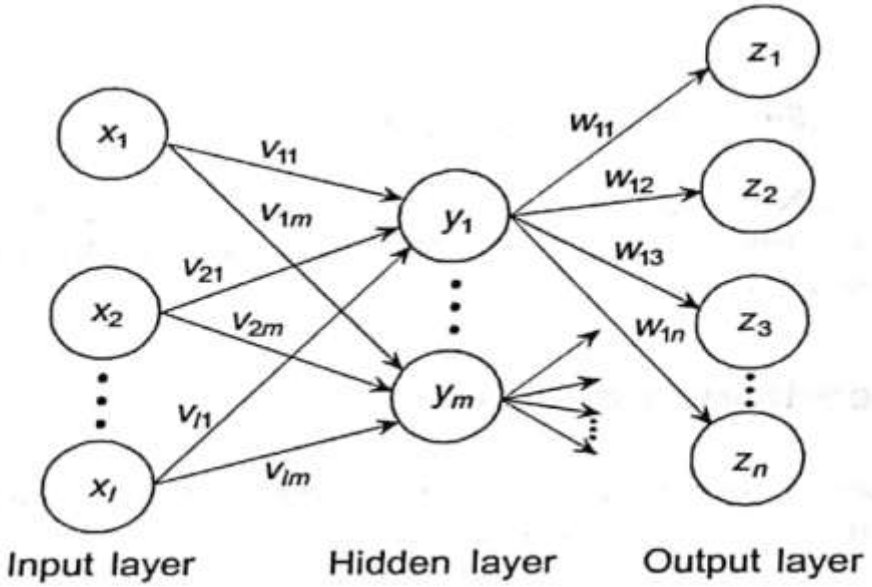


OUTPUT LAYER

X_i :Input neurons
Y_j : Output neurons
W_{ij} :Weights

Multilayer Feed forward Network

- This feedforward nerual network is called Multilayer Feedforward Network
- The processing as input and output layer also have one or more intermediary layers called hidden layers.
- The input layers neurons are linked to the hidden layer neurons and the weighted on the this link as referred to as input –hidden layer weights.
- A hidden layer neurons weights are linked to the output layer neurons corresponding weights are referred to as hidden output layer weights
- I input neurons ,m_1 neurons is the first hidden layer,m_2 second hidden layer neurons.n out put neurons
- Output layer 1-m_1-m_2-n

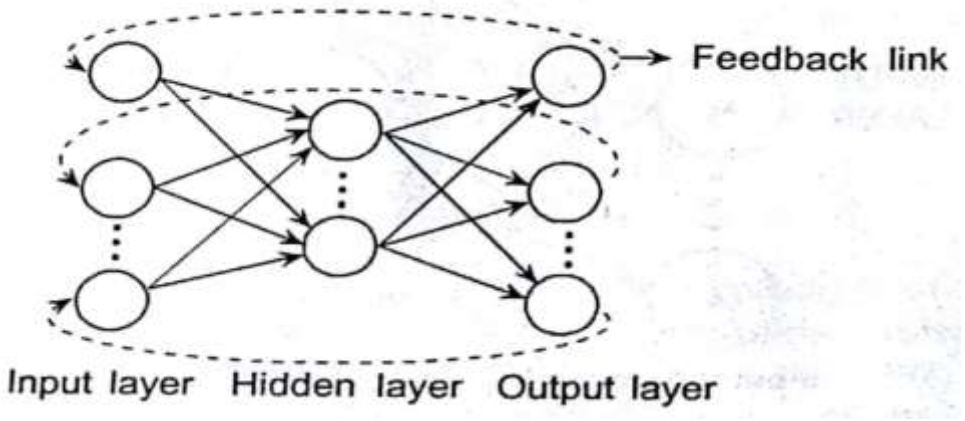


Recurrent Networks

- These networks differ from feedforward network architectures in the sense

that there is at least one feedback loop

- There one layer with feedback connection .There could also be neuron self feed back link.



Characteristics of Neural Network

- The NN exhibit mapping capabilities. That map input patterns to their associated
- Outptu patterns
- The Neural Network Architecture can be trained with know examples of a problem
- Before they are tested for their infernce capability on unknown instances of the problem.
- The NN posses the capability to generalize. They outcome to the past trends.
- The NN are robust system are fault tolerant. They recall full patterns from incomplete, partial or noisy patterns.
- The NN can process information in parallel at high speed and in distributed manner

Learning Methods

- Supervised learning:
- A teacher is assumed to be present during the learning process, when a comparison is made between the network's computed output and the correct expected output, to determine the error.
- The error can then be used to change network parameters, which result in an improvement in performance.
- Unsupervised learning:
- The target output is not presented to the network. It is as if there is no teacher to present the desired patterns and hence, the system learns of its own by discovering and adapting to structural features in the input patterns.
- Reinforced learning:
- A teacher though available, does not present the expected answer but only indicates if the computed output is correct or incorrect. The information provided helps the network in its learning process. A reward is given for a correct answer computed and a penalty for a wrong answer.

- Hebbian learning:
- In this input –output pattern pairs(X_i,X_j) are associated by the weight matrix W known as the correlation matrix, its computed as

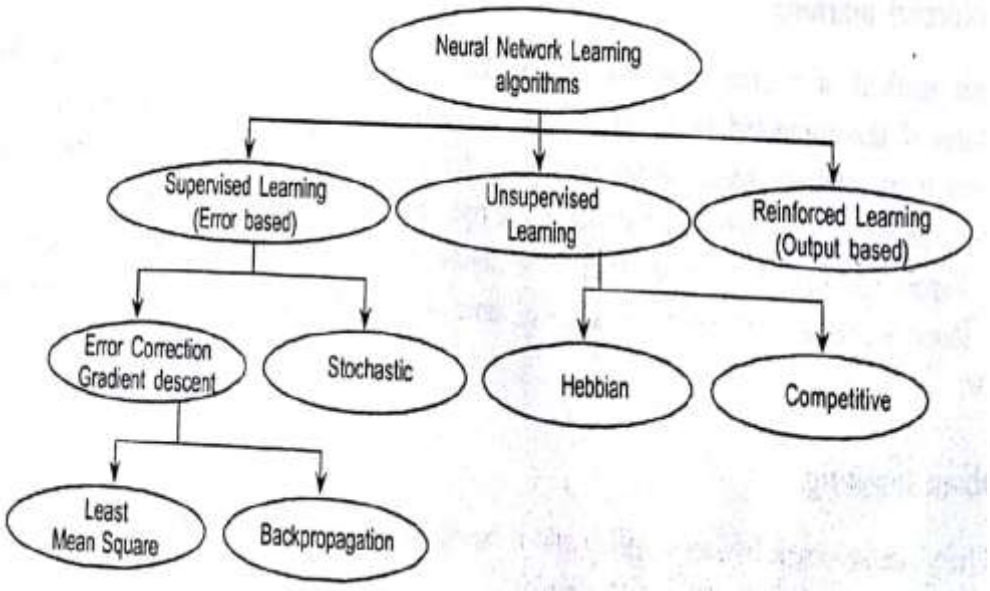
$$W = \sum_{i=1}^{n} X_i Y_i^T$$

- Y_i^T Is the transpose of the associated ouput vector Y_j
- Gradient Descent Learning:
- This based on the minimization of error
- E define terms as weights and the activation function of the network.
- As weight update is depent on the gradient of the error E
- i and j neuron of the neighbouring layers Δw_{ij} defined as

$$\Delta W_{ij} = \eta \frac{\partial E}{\partial W_{ij}}$$

Competitive Learning

- Competitive Learning: When an input pattern is presented ,all neurons in the layer
- Compete and the wining neurons. Undergoes weight adjestments it's a "winner-takes –all"
- Stochastic learning: In this method weight are adjusted the probabilistic fashion.
- Ex : Anneling- the learning mechanism employed by Boltzmann
-

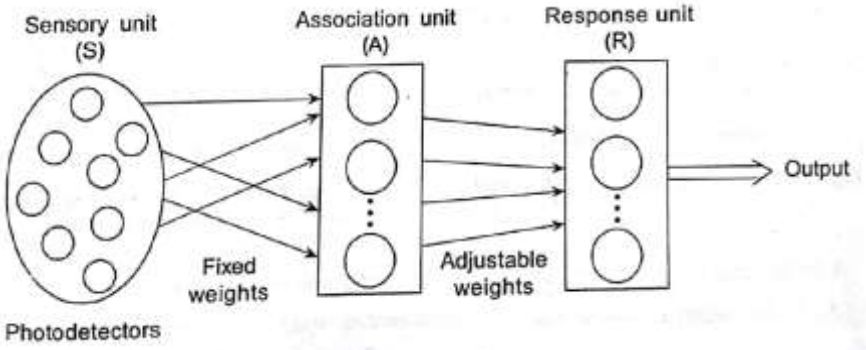


TAXMONY OF NERUALNETWORK ARCHITECTURES

- ADALINE (Adaptive Linear Nerual Elements)
- ART(Adaptive Resonance Theory)
- AM(Associative Memory)
- BAM(Bidirectional Associative Memory)
- Boltz MANN Machine
- BSB(Brain –Sate –in-box)
- CNN(Cascade Correlation)
- Cauchy Machine
- CPN(Counter Propagation Network)
- Hamming Network
- Hopfiled Network
- LVQ(Learning Vector Quantization
- MADALINE(MANY ADALINE)
- MLFF(Multilayer Feedforward Network)
- Neocogination
- Perceptron
- RBF(Radial basic Function)
- RNN(Recurrent Neural Network)
- SOFM(Self-organizing feature map)

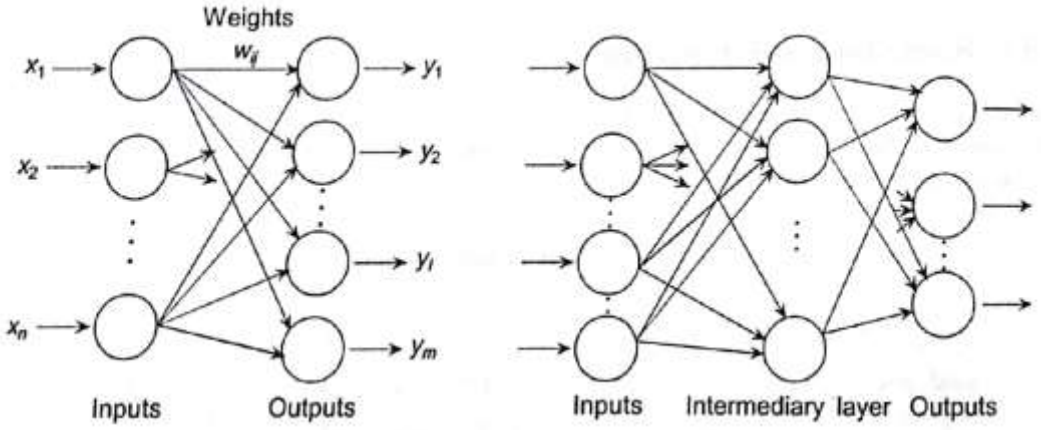
Early Neural Network Architectures

- Rosenblatt's Perceptron:
- The perceptron is a computional model of the retina of the eye is named as perceptron.
- The network compries three units the Sensory Unit S, Association unit A and Response Unit R

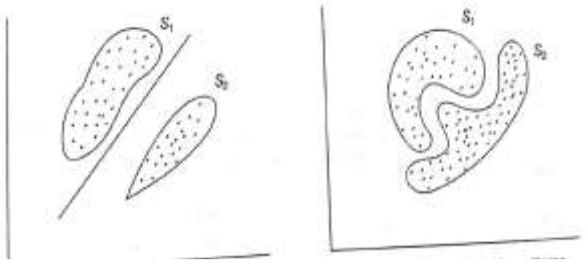


- The s unit comprising 400photodetector receives input images and provide 0/1 electric signal as output.the photo detector outputs 1 else 0.
- The photodectors are randomly connected to the Association UnitA
- The A unit comprised feature demons or predicates.
- The predicates examine the output of the s unit.
- The third unit R compries pattern recognizers or perceptrons.
- The weight of the S and A units are fixed ,R are adjustable.
- Ouput R unit is the weighted sum of the inputs is less than or equal to 0

- $Y_i = f(net_j) = 1$ if $net_j > 0$
- $= 0$ other wise
- Net j= $\sum_{i=1}^{n} x_i w_{ij}$
- X_i is the input w_{ij} is the weight on the connection leading to the output units and y_j is the output
- The training algorithm of the perceptron is supervised learning algorithm weight adjusted to minimize the error the computed output does not match the target output.



- A basic learning algorithm for training tire perceptron is as follows:
- If the output is correct then no adjustment of weights is done
- $W_{ij}^{(k+1)} = W_{ij}^{(k)}$
- If the output is 1 but should have been 0 then the weights are decresed as the active links input links.
- $W_{ij}^{(k+1)} = W_{ij}^{(k)} - \alpha.x_i$
- If the output is 0 but should have been 1 then the weights are increased on the active input links.
- $W_{ij}^{(k=1)} = W_{ij}^{(k)} + \alpha.x_i$
- $W_{ij}^{(k+1)}$ new adjusted weights, $W_{ij}^{(k)}$ old weights,input α to fast learning parameter.
- Large α risk of allowing weights to oscillate about which increment algorthim.
- Peceptron and Linearly separable tasks
- Sets of points in two dimensional spaces are linearly separable if the set can be separated by a straight line.



- (a) Linearly separable patterns (b) Non-linearly separable patterns

XOR Problem

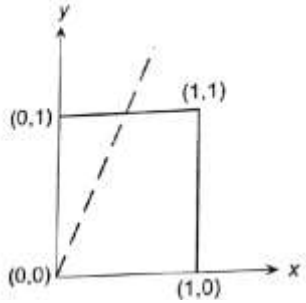
- XOR is logial operation as a truth table

Inputs	Inputs	Output
0	0	0
1	1	0
0	1	1
1	0	1

Event parity

odd parity

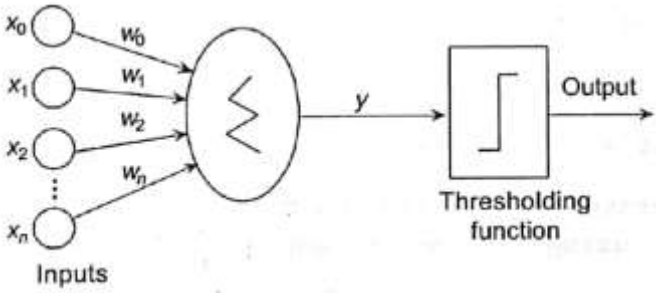
- The problem for the ANN is to classify the inputs as odd parity or even parity
- Odd parity Means odd number of 1bits input,
- Even party referes the even number of 1 bits inputs.
- Fig represented the perceptron is unable to find a line separting even parity input
- Patterns from the odd parity input patterns.



- the non linear separable pattern of the XOR problem

ADALINE NETWORK

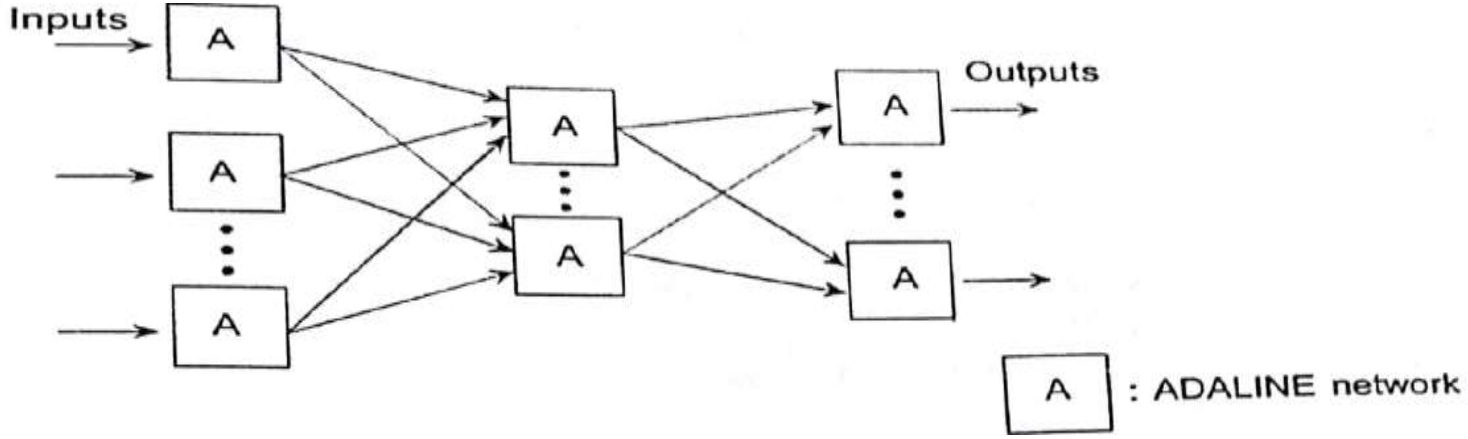
- The Adaline network framed by Bernad Widrow of standardford Universiy use of supervised learning.
- There is only one output neuron and the ouput values.are bipolar(-1 or +1)
- Bias weight the input x_0 with an input link of x_0 =+1
- If the sum weighted inputs is greater than or equal to 0 then the output is 1
  otherwise -1
- The supervised learning algorthim adopted by the network is similar perceptron learning algorithm.
- $W_{inew} = W_i^{old} + \alpha(t-y)x_i$
- A is the learning coefficent ,t is the target output,y is the computed output.x_i input



- ADALINE Network to used modem and telecommunication to reduce the echo

MADALINE Network

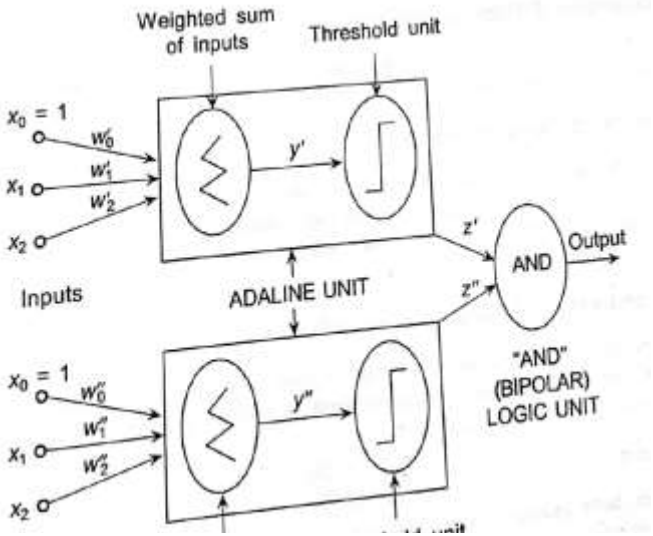
- The MADALINE netrwork with two units exhibits the capability to slove the XOR
- Problems.
- Each ADALINE unit recevies the input bits x_1,x_2 and the bias input x_0=1 as its input
- The weighted sum of the inputs is calculated and passed on to the bipolar thershold units.
- The logical anding of the two thershold output are computed to obtain the final output.
- Therhold outputs are both +1 or -1 then the final output +1
- Thershold output are differnet (+1,-1) then the final out puts -1
- Even parity produces postive outputs and inputs of odd parity produce negative outputs. .

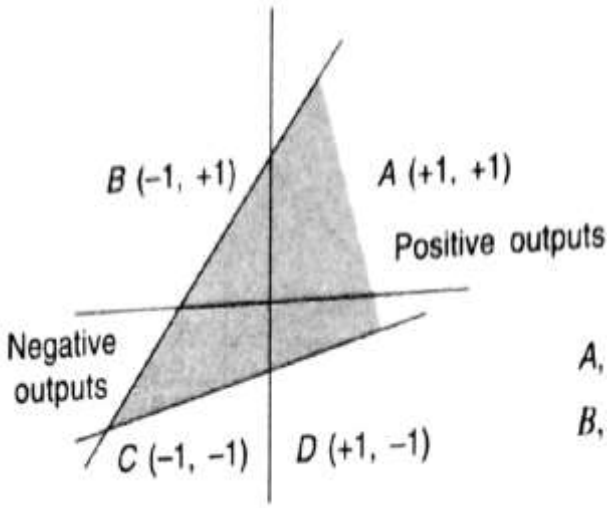


MADALINE NETWORKS

MADALINE NETWORK TO SLOVE XOR PROBLEM

- The learning rule adopted by MADALINE network is termed as MADALINE adaption rule (MR) and is a form of supervised learning
- In this method to adjust the weight error minimum for the current training pattern
- But the with as little damage to the learning acquried through the pervious traning pattern.





Some Application Domains

- Neural networks have been successfully applied for the solution
- Pattern recognition (PR)/image processing
- Neural networks have shown remarkable progress in the recognition of visual images, handwritten characters, printed characters, speech and other PR based tasks.
- Optimization/constraint satisfaction
- This comprises problems which need to satisfy constraints and obtain optimal solutions. Example of such problems include manufacturing scheduling, finding the shortest possible tour given a en of cities
- Forecasting and risk Assessment
- Neural networks have exhibited the capability to predict situations from past trends.
- Therefore found sample application in area such as meterology, stock market banking and econometrics with high success rate.
- Control systems:
- Neural Networks have gained commerical ground by finding application in control system.Dozen of computers products especially by the japenese compaines incorporating NN technology used for the control of chemical plants,robots .
-