SIMATS
School of Engineering

Artificial Intelligence

Computer Science and Engineering

Saveetha Institute of Medical And Technical Sciences,Chennai.

INDEX

1

WHAT IS AI ?

* Thinking humanly - cognitive modelling approach
* Thinking rationally - laws of thought approach
* Acting humanly - Turing test approach
* Acting Rationally - Rational agent approach

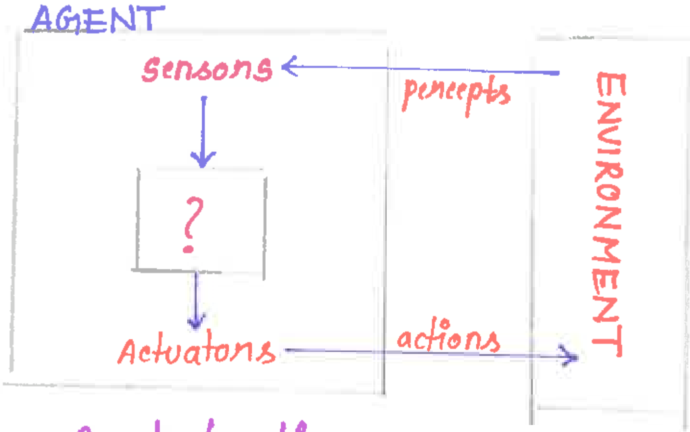
AI PROBLEMS :

→ Psycology
→ Mathematics
→ Economics
→ Linguistics
→ Philosophy
→ Planning
→ Robotics
→ Neuroscience
→ Computer Engineering
→ NLP
→ knowledge representation
→ Speech processing
→ Control therapy & cybernetics

FOUNDATION & HISTORY OF AI :

- 1943-1955 — The gestation of AI
- 1956 — Birth of AI
- 1952-1969 — Early enthusiasm, great expectation
- 1966-1973 — A dose of reality
- 1969-1979 — Knowledge based systems (The key to power)
- 1980-present — AI becomes an Industry
- 1986-present — Return of Neural networks
- 1987-present — AI adopts scientific method
- 1995-present — Emergence of Intelligent Agents
- 2001-present — Availability of very large Datasets

INTELLIGENT AGENT :



→ Agent function
→ Agent Program
→ Agent Properties

CONCEPT OF RATIONALITY :

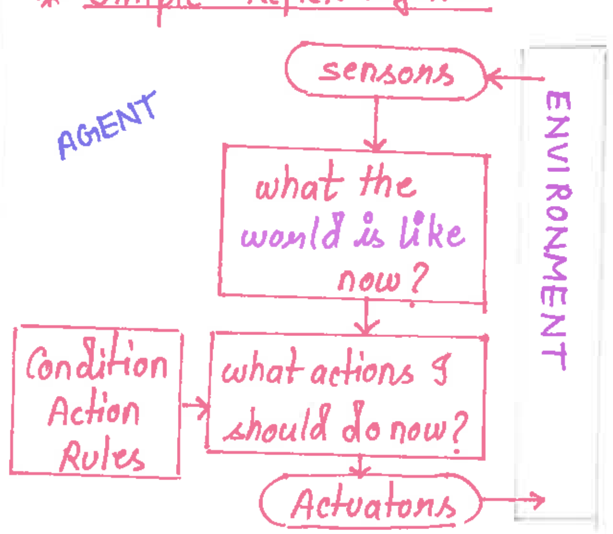
* Nature of Environment -
  → Performance Measures
  → Nature of Environment
  → Actuators
  → Sensors

* Properties of Task environment -

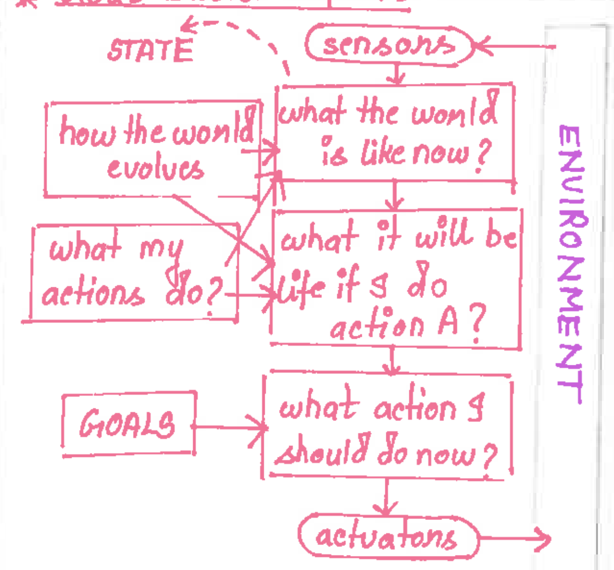
Agent Type	Performance Measure (P)	Environment (E)	Actuators (A)	Sensors (S)
Path-Picking Robot	% of parts in correct bins	Conveyor belt with parts; bins	Jointed arm & hand	Camera, joint angle sensors
Interactive English Tutor	Student's score on test	Set of students, Testing agency	Display of exercises, suggestions, corrections	keyboard Entry

STRUCTURE OF AGENTS :-

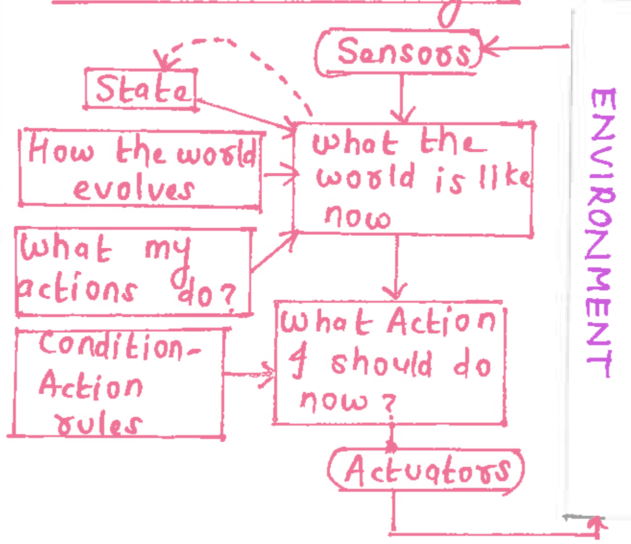
* Simple - Reflex Agents



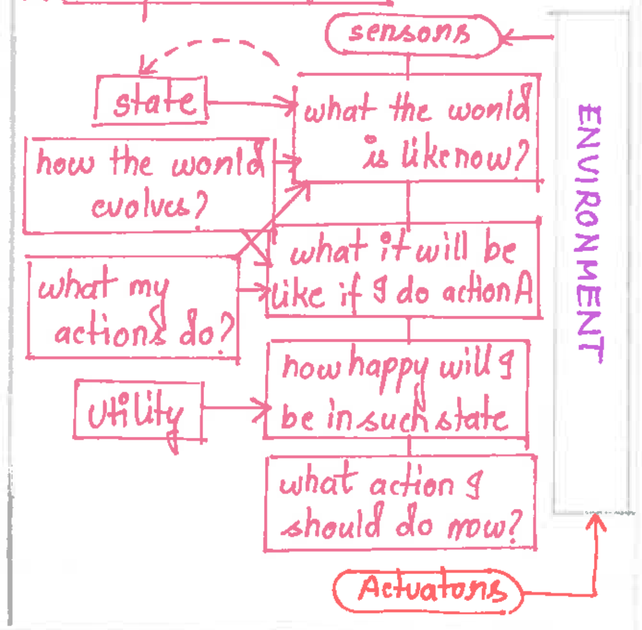
* Goal Based Agents



* Model-Based Reflex Agent

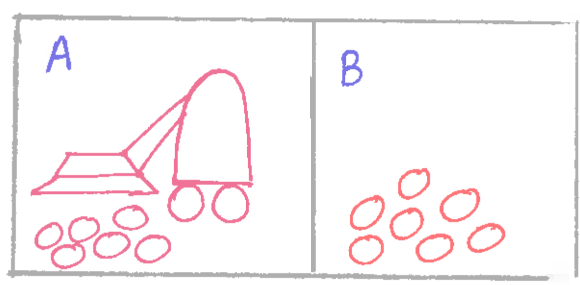


* Utility Based agents.

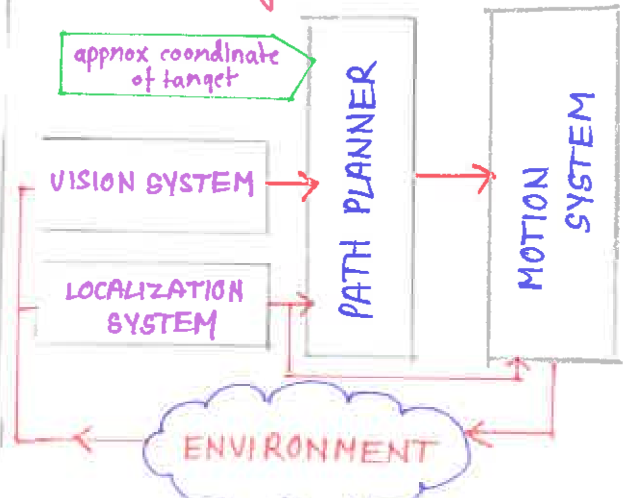


APPLICATIONS OF AGENTS

* Vaccum cleaner Agent -



* Path Picking Robot -



PROBLEM SOLVING AGENTS :-

STEPS →

→ Goal formulation

→ Problem formulation

→ Search

→ Execute solution

TOY PROBLEM :-

* 8 Puzzle Problem -

∘ Initial State -

5	4	
6	1	8
7	3	2

∘ Goal State -

1	2	3
4	5	6
7	8	

* 8-Queens / n-Queens Problem -

→ Queens should not clash with each other.

* Cript Arithmetic

* Vaccum World

* Missionaries & Cannibals

REAL WORLD PROBLEMS :-

* TRAVELING SALEPERSON * VLSI Layout

* ROBOT NAVIGATION * ASSEMBLY SEQUENCING

* WATER-JUG PROBLEM →

- state Representation

∘ x = 0,1,2,3,4 ∘ y = 0,1,3

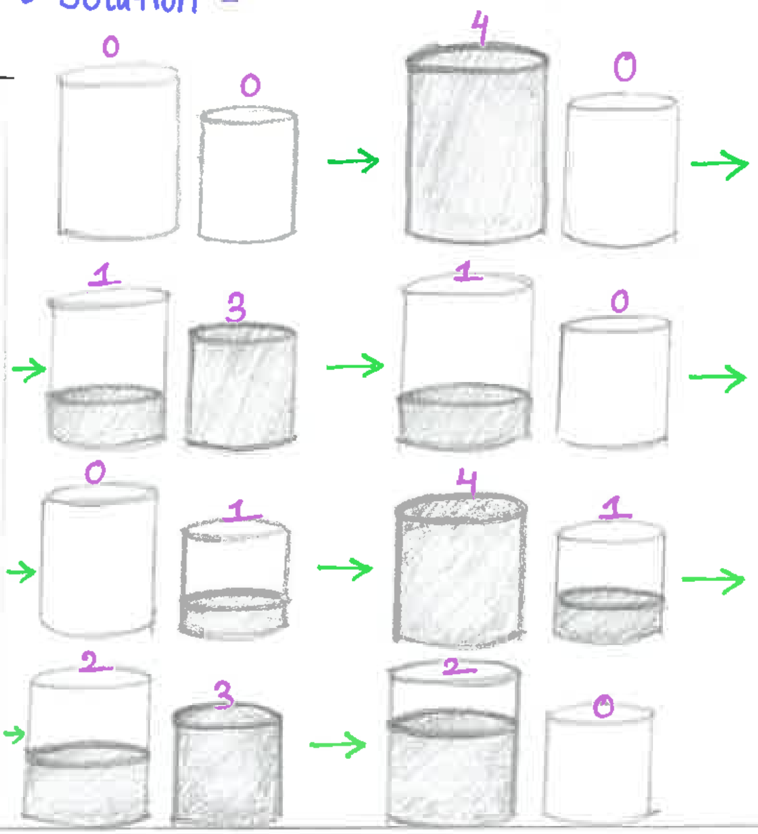
∘ Start State : (0,0) ∘ Goal State : (2,0)

∘ Operations -

1) Empty a Jug 2) Fill a Jug

3) Pour water from one jug to another.

∘ Solution -



SEARCHING FOR A SOLUTION -

Components of Search tree

- STATE — state Space
- PARENT NODE — Initial Node
- ACTION — applied to parent node
- PATH COST — g(n)
- DEPTH — no. of steps

Measuring Problem-solving performance

- Completeness — guarantee to find a solution
- Optimality — find the highest quality solution
- Time Complexity — time to find a solution
- Space Complexity — memory requirement to find a solution

UNINFORMED SEARH STRATEGY

Breadth First Search

- Start from ROOT
- Travel down level by level
- Add node at end
- COMPLETE (YES)
- TIME $O(b^{d+1})$
- SPACE $O(b^{d+1})$
- OPTIMAL (YES)

Uniform Cost Search

- BFS with Uniform Cost
- → BFS Optimal → Expand Node first
- follows priority Queue
- ADVANTAGE : - solution is cheapest

Depth first Search

- SINGLE PATHDOWN - until reach goal - reach dead end
- FOLLOW BACTRACK UP - until reach branch
- EXPAND DEEPEST (Root, left, Right)
- complete : (NO) - optimal : (NO) - time : $O(b^m)$ - space : $O(bm)$

SEARCHING WITH PARTIAL INFORMATION

- SENSORLESS PROBLEMS
- CONTIGENCY PROBLEMS
- EXPLORATION PROBLEMS

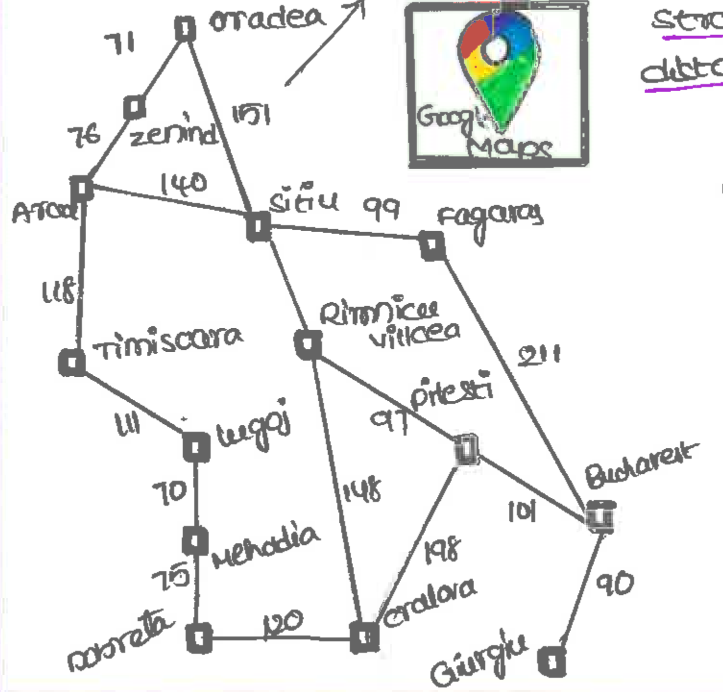
BEST-FIRST SEARCH :-

- Node is selected for Expansion
- lowest evaluation function is selected
- expanding the first node first
- Different evaluation function
- h(n)= cheapest path node(n)-goal node
- example:- Route planning (cheapest path)
- g(n)= cost - initial - current state state
- f(n)= evaluation function (node for expansion)

Greedy Best-First Search :-

- expand the node - closest to the goal
- f(n)= h(n) [most desirable]
- straight line distance
- complete (NO)
- Time $O(b^m)$
- space $O(b^m)$
- optimal (NO)

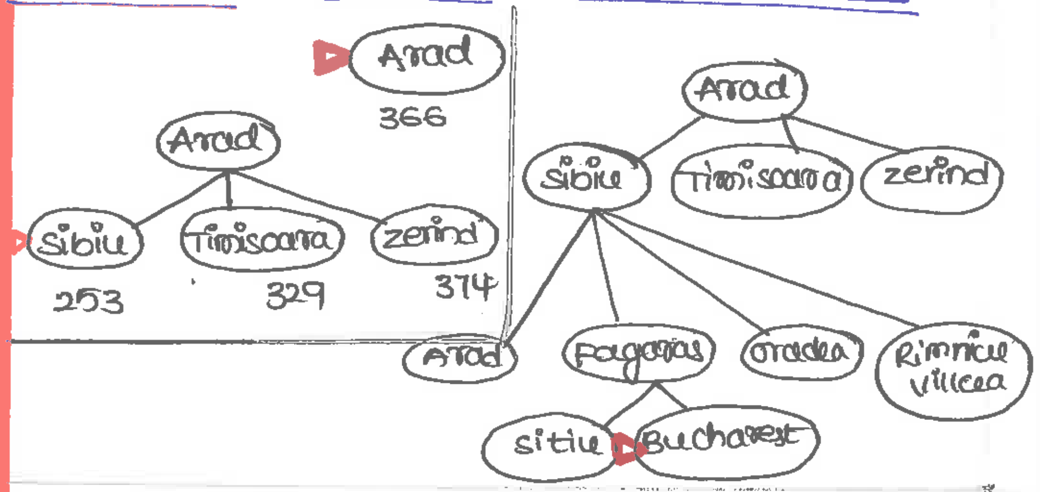
Realtime example:- Google Maps



Straight line distance to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Fagaras	178
Giurgiu	77
Lugoj	244
Mehadia	241
Oradea	380
Pitesti	98
Timisoara	329
Rimnicu Vilcea	193

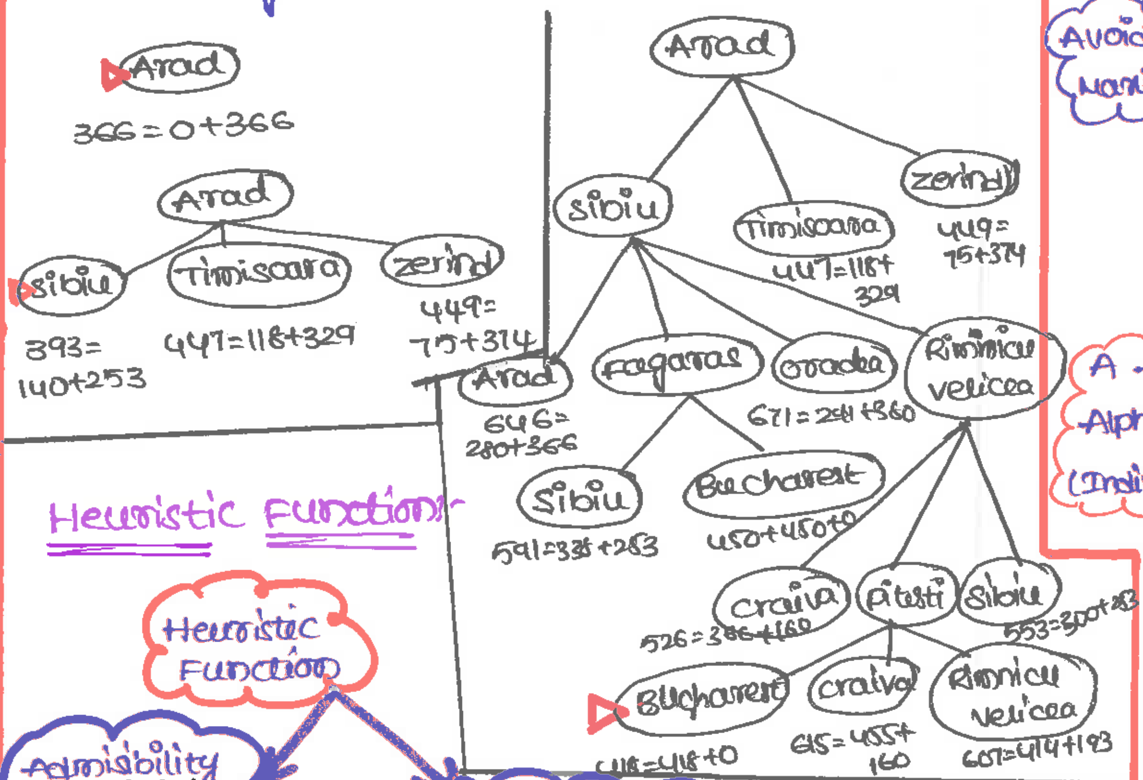
INFORMED SEARCH STRATEGIES



A* Search :-

f(n)= g(n)+ h(n)

* f(n)= estimated total cost from initial node to goal node
* g(n)= cost from initial to current node
* h(n)= estimated cost from current to goal node



Heuristic Function:-

Heuristic Function

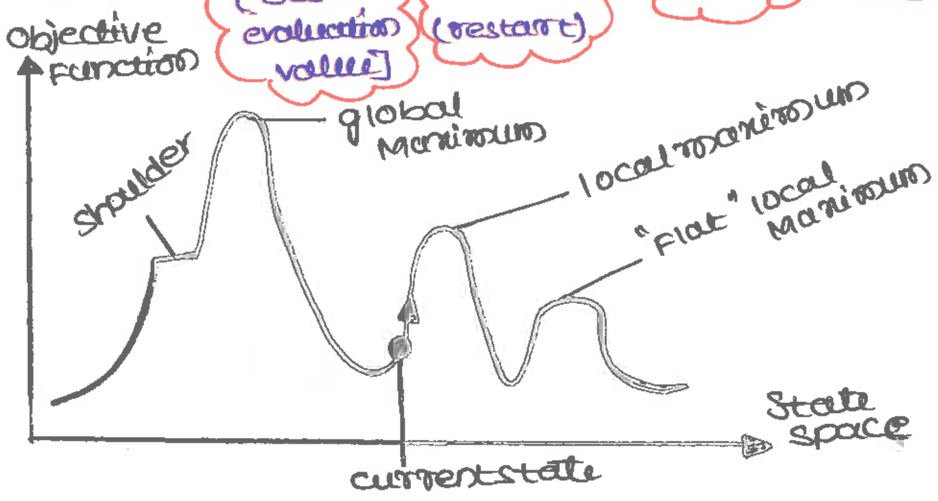
Admissibility
If and only if $\forall s \in$ state space: $h(s) \le \min\{cost(a_1, a_2, \ldots a_n) \mid$ transition-model(t-M(t-M(s, a_1), a_2), ... a_n) is goal$\}$ holds

Consistency
If and only if $\forall s \in$ state space, action a applicable, $s' =$ transition-model(s,a): $h(s) \le cost(a) + h(s')$

LOCAL SEARCH ALGORITHMS :-

Hill climbing search :-

- Start Randomly
- Move - neighbour (best evaluation value)
- Strict local minimum reached (restart)
- Terminate - peak reached
- Doesn't back track



Simulated annealing (SA) :-

simulated annealing
- Avoid local maxima
- uses random search
- uses a control parameter "T" (starts high - decreases to 0)
- If $\Delta E > 0$ then current ← next else current ← next only with $(e^{\Delta E/T})$

Genetic algorithm

- A finite Alphabet (Individual)
- Population (K-randomly generated)
- Fitness function higher value better state
- selection Individual (gives next generation)
- crossover
- Mutate

Online search Agents

- Agents operate (Interleaving computation & action)
- Action then (observes the involvement)
- Dynamic or semi-dynamic domain
- Exploration problem (actions are unknown)

Realtime Application :- Google Map.

CONSTRAINT SATISFACTION PROBLEM

MAP COLOURING:-

↳ Needs to satisfy set of constraints

Input:- 2D Array graph [V,V]

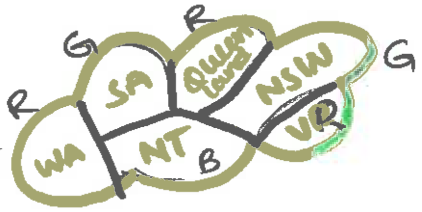
→V→ no. of vertices

→V→ no. of adjacent matrix

→M→ integer M is maximum no. of colour.

Output:- Array colour "V" should have no. from 1 to m.

Example:-



Australian Map

I/P:- graph = {0,1,1,1,}
{1,1,0,1,0}
{1,1,0,1}
{1,0,1,0}

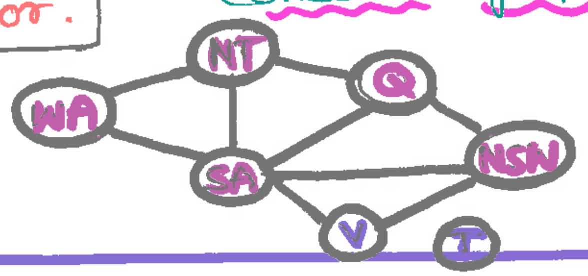
O/P:- solution exists

Following assigned colors

1, 2, 3, 2

By coloring the vertices with colors, adjacent does not have same color.

constraint graph



CRYPTO ARITHMETIC PROBLEM:-

→ For A→Z assign 0→9 Number

→ Note: No 2 Alphabet should have same number

Example:-

* D
\+ E
Y

Addition of both

D=7
E=5
12

Value assumed

* BASE
\+ BALL
GAMES

Both are same value

Need to get two digit Number

B=7 L=5
A=4 G=1
S=8 M=9
E=3

No value should have same number

QUIZ 2:-

* NOON
MOON
SOON
JUNE

What is the value of June?

* YOUR
\+ YOU
HEART

What is the value of Heart?

WORKING OF CAP :-

* Send + More = Money

Initial state

M=1 , S= 8 (or) 9
O= 0 (or) 1
N= E (or) E+1
N+R >8
E<>9

N=3
R= 8(or) 9
2+D= Y

C1=0 | C1=1

2+D=Y
N+R=10
R= 9
S = 8

2+D=10+Y
D= 8+Y
D= 8(or) 9

D=8 | D=9

Y=0 conflict | Y=1 conflict

REAL WORLD CSPs :-

- Assignment problems
  - e.g, who teaches what class
- Timetabling problems
  - e.g, which class is offered when and where?
- Transportation scheduling
- Factory scheduling

4- QUEEN PROBLEM

Rule:- Queen can travel

→ Row wise

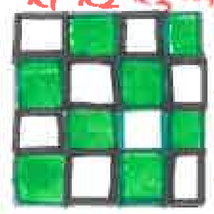
→ column wise

→ Diagonal wise

NOTE:- No queen → should → Attack → each other

Initial state:-

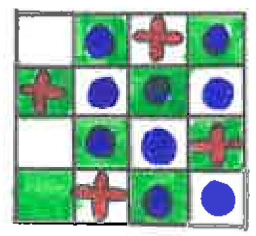
x_1 x_2 x_3 x_4



x_1 {1,2,3,4}
x_2 {1,2,3,4}
x_3 {1,2,3,4}
x_4 {1,2,3,4}

Goal:- All queen should be placed

Final state:-



x_1 { ,2,3,4}
x_2 { , , ,4}
x_3 {1, , , }
x_4 { , ,3, }

REAL-TIME APPLICATIONS

→ Games which satisfies the set of constraints.

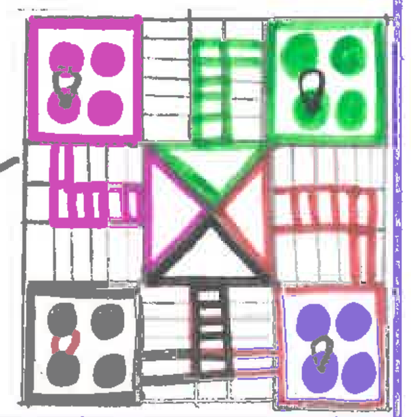
Example:- Monopoly Games

→ Chess

→ Electronic Banking

→ Ludo

→ Snake & Ladder



GAME PLAYING

SEARCH VS GAMES

Search	Games
* NO adversary	* Adversary
* Solution - path from start → goal	* Solution is strategy
* Ex:- path planning, scheduling activities	* Ex:- chess checkers.

TIC TAC TOE PROBLEM

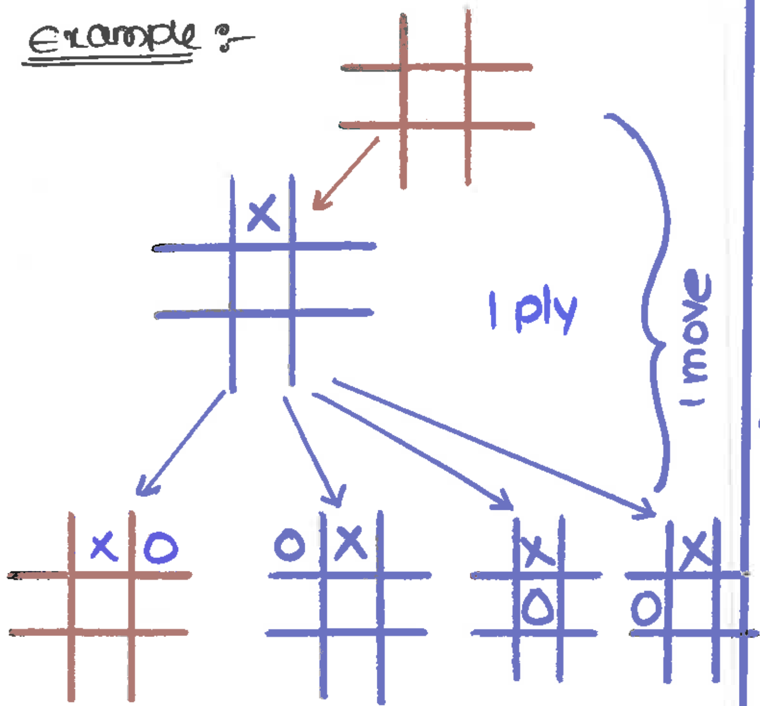
↳ Needs to satisfy set of constraints.

→ Game consists of Noughts and crosses (or) Xs and Os

→ 3X3 grid play

→ If 3 consecutive marks formed then the player who owns the move will won.

Example :-



MINI-MAX ALGORITHM

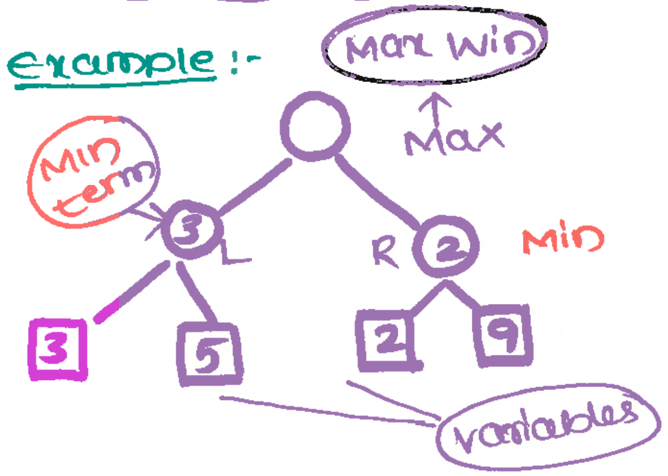
Perfect play for Deterministic environments with perfect Information

Basic Idea: choose move with highest minimax value

Max-position: largest successor

Min-position: smallest successor

Example :-



2 player gem

→ player 1 :- Min- highest score

→ Player 2 :- Max- lowest score

properties :-

- performs a complete depth first exploration of the game tree
- optimal

Time complexity :-

$O(b^d)$

Space complexity :- $O(bd)$

∴ B is branching factor

D is Depth of tree.

ALPHA BETA PRUNING (α-β)

↳ Needs to satisfy set of constraints

→ Similar like Min Max

→ Reduce computation time.

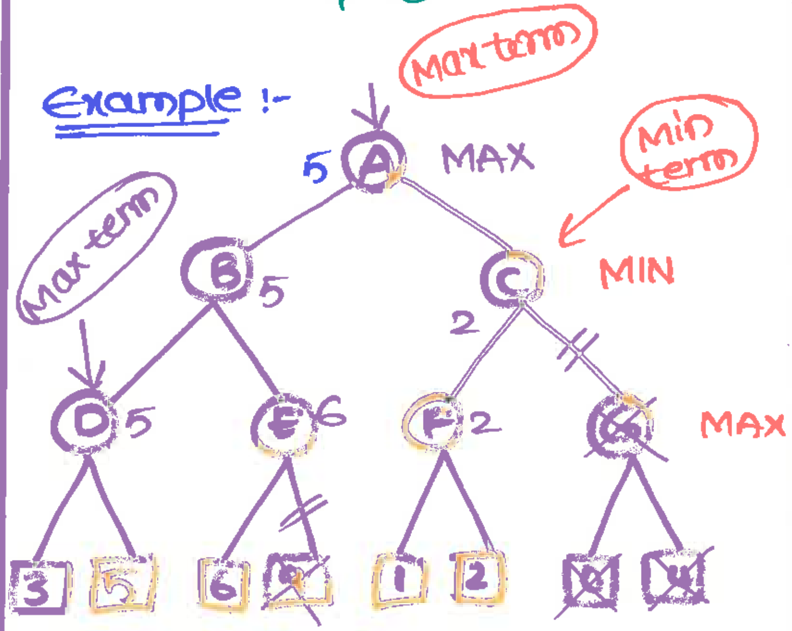
→ Cuts of branches in game tree.

Parameters :-

① Alpha :- Max value $-\infty$

② Beta :- Min value $+\infty$

Example :-



∴ Here optimal value acheived.

∴ un used nodes are deleted

pruning

EVALUATION FUNCTIONS

→ order the terminal states

→ computation must not take too long.

→ for non-terminal states → evaluation function should strongly correlated

Ideal function :- returns the utility of the position.

In practice :- Typically weighted linear sum of features

$Eval(s) = w_1f_1(s) + w_2f_2(s) + \dots + w_nf_n(s)$

Example :- chess game

$f_1(s)$ = (num white queens - num black queens) etc,

REAL TIME APPLICATIONS

Games

Single Agent Games

→ Temple run

→ Subway surfers

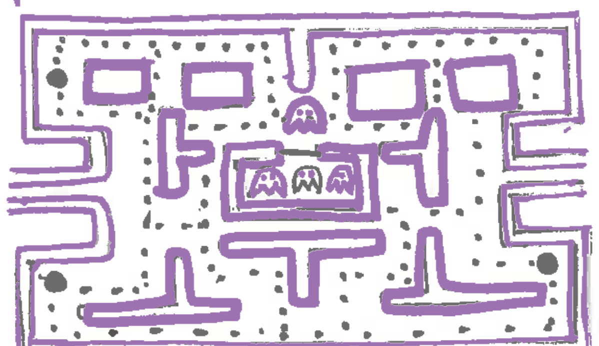
→ pac-Man

Multi Agent Games

→ pubg

→ consta

→ KOO



Knowledge And Reasoning

Logical Agents

Representation — How are the things stored?

Reasoning — How is the knowledge used?

Two Generic Functions

Tell — Add new sentences (facts) to the KB

Ask — Query what is known from the KB

Knowledge Based Agents

Domain Specific Content

Domain Independent Algorithm

Inference Engine

Knowledge Base

Domain Specific Content

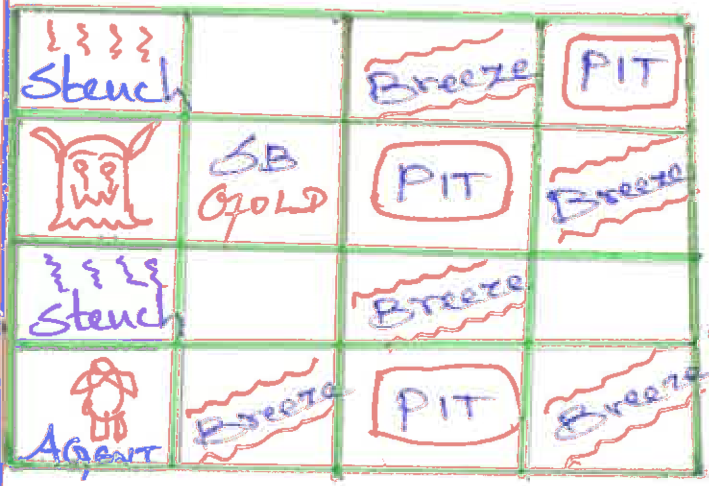
Inference Engine

A systems that applies logical rules to the knowledge base to deduce new result.

Domain Independent Algorithm

Initial state — Operators — Goal test function

Wumpus World



1,4	2,4 P?	3,4	4,4
1,3 W!	2,3 A	3,3 P?	4,3
1,2 S OK	2,2 OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P?	4,1

Final Solution

Performance Measure

- Gold + 1000
- Death − 1000

Environment

- Square adjacent to the Wumpus are smelly
- Squares adjacent to the pit are breezy
- Shooting kills Wumpus if you are facing it.
- Grabbing picks up the gold if in the same square

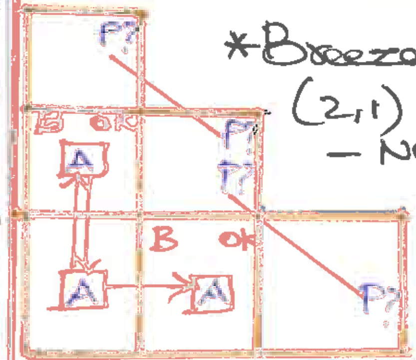
Actuators

- Left turn, right turn
- forward, release, shoot

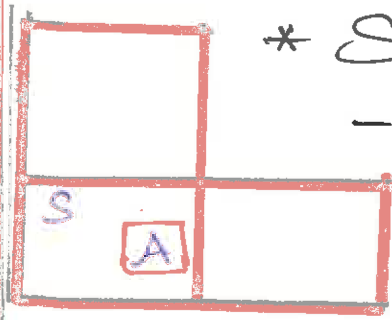
Sensors

- Breeze, glitter and Smell

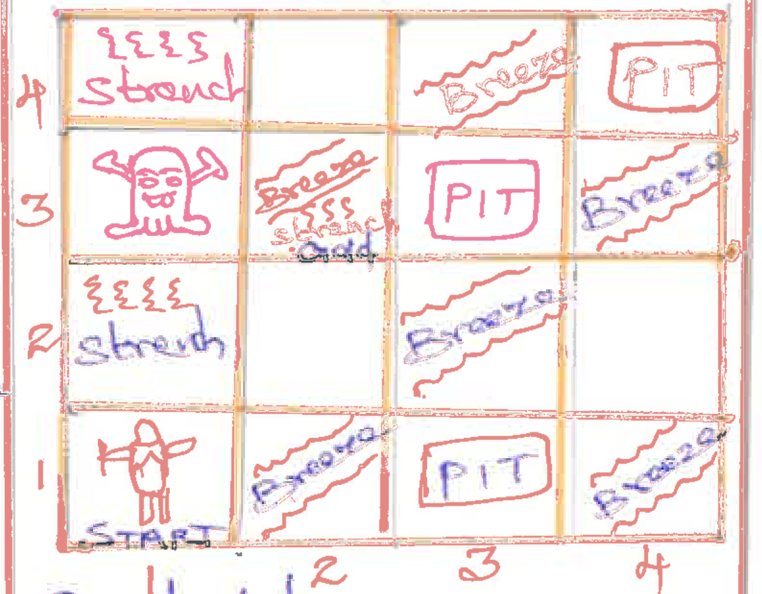
Sticky Situations



* Breeze in (1,2) and (2,1)
  — No safe actions



* Smell in (1,1)
  — Cannot move

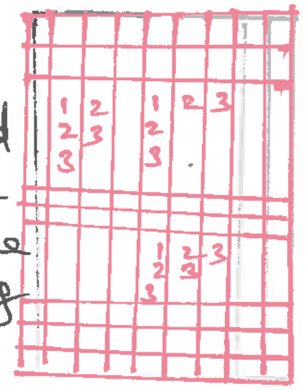


Constraints

1. Rooms consists of Pits, Wumpus, Stench, Gold, Breeze
2. Adjacent → Room → Wumpus → Smelly
3. Adjacent → Room → Pit → Breeze
4. Adjacent → Room → Glitter → Gold
5. Reward → +1000
6. Loss → −1000
7. Each Iteration → −1
8. Using Arrow → −10

Real-Time Applications

* Minesweeper Game
  * To locate a predetermined number of randomly placed mines in the shortest possible time by clicking on safe



Propositional Logic

- Declarative statements which either True / False.

Atomic Sentences

↓

Single Propositional Symbols

True = always true,
False = always false

Complex Sentences

Negation

- If S is a sentence, $\boxed{\neg S}$ is a sentence

Conjunction

- If S_1 and S_2 are sentences, $\boxed{S_1 \wedge S_2}$ is a sentence

Disjunction

- If S_1 and S_2 are sentences, $\boxed{S_1 \vee S_2}$ is a sentence

Implication

- If S_1 and S_2 are sentences, $\boxed{S_1 \Rightarrow S_2}$ is a sentence

Biconditional

- If S_1 and S_2 are sentences, $\boxed{S_1 \Leftrightarrow S_2}$ is a sentence

Wumpus World Sentences

* Let P_{ij} be true if there is a Pit in [i, j]
* Let B_{ij} be true if there is a breeze in [i, j]

Start: $\neg P_{1,1}$
$\neg B_{1,1}$
$B_{2,1}$

* "Pits cause breezes in adjacent Squares"

$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
$B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$

Entailment & Derivation

Entailment: $KB \models q$

No logically possible world where q is false

True in every logically possible world

Derivation: $KB \vdash q$

- Valid inference Steps

Soundness:

If $KB \vdash q$ then $KB \models q$

⇓

Inference produces only real entailments

Completeness:

If $KB \models q$ then $KB \vdash q$

⇓

Inference produces all entailments

Reasoning Patterns in Propositional logic

Reasoning Patterns

- Inference Rules
- Modus Ponens: $S_1 \Rightarrow S_2$ and S_1, derive S_2
- And-Elimination: $S_1 \wedge S_2$, derives S_1
- De morgan's law: $\neg(A \vee B) = \neg A \wedge \neg B$, $\neg(A \wedge B) = \neg A \vee \neg B$

And Elimination: $\dfrac{\alpha \wedge \beta}{\alpha}$

Modus Ponens: $\dfrac{\alpha \Rightarrow \beta, \ \alpha}{\beta}$

Proof By Deduction

S → state, P – pit, B – Breeze

Knowledge

Rule:

$S_1: B_{22} \Leftrightarrow (P_{21} \vee P_{23} \vee P_{12} \vee P_{32})$

Observation

$S_2: \neg B_{22}$

Inferences

S_1, Bi elimination

$S_3: (B_{22} \Rightarrow (P_{21} \vee P_{23} \vee P_{12} \vee P_{32})) \wedge ((P_{21} \vee P_{23} \vee P_{12} \vee P_{32}) \Rightarrow B_{22})$

S_3, And Elimination

$S_4: ((P_{21} \vee P_{23} \vee P_{12} \vee P_{32}) \Rightarrow B_{22})$

Contra Pos

$S_5: (\neg B_{22} \Rightarrow \neg(P_{21} \vee P_{23} \vee P_{12} \vee P_{32}))$

S_2, S_6, MP

$S_6: \neg(P_{21} \vee P_{23} \vee P_{12} \vee P_{32})$

S_6, De morgan law

$S_7: \neg P_{21} \wedge \neg P_{23} \wedge \neg P_{12} \wedge \neg P_{32}$

REAL-TIME APPLICATIONS

OUT DOOR GAMES

- CRICKET
- FOOT BALL
- BASKET BALL
- VOLLY BALL

→ Climate Condition

- Good → MATCH CONDUCTED
- BAD → MATCH CANCELLED

First order Logic

Basic Elements

- Constant symbols — stand for objects
- Predicate symbols — stand for relations
- Function symbols — stands for functions

Steps for Knowledge Engineering

1. Identify the task
2. Assemble the relevant knowledge
3. Decide on a Vocabulary of predicates, functions and constants
4. Encode general knowledge about the domain
5. Encode a description of the specific problem instance
6. Pose queries to the inference procedure and get answers
7. Debug the knowledge base

FOL Version of Wumpus World

- Percept Sentence
  - Stench
  - Breeze
  - Glitter
  - None
- Actions
  - Right
  - Left
  - forward
  - shoot
  - Grab
  - Release
  - climb
- Best action Construct query
  - Best action {a,5}
- Ask Solve this and returns
  - TELL about the action

Inference With FOL

QUANTIFIERS

- Universal
  - $\forall$ (Variables) (Sentence)
  - $\forall x$, Person(x) $\Rightarrow$ likes(x, McDonalds)
- Existential
  - $\exists$ (Variables) (Sentence)
  - $\exists x$, likes(x, McDonalds)

Propositional Vs FOL

Unification & Lifting

- Unification: Finding all legal substitutions
- Lifting: Transformed from PL → FOL

Unify(a,b) = θ if $a\theta = b\theta$

Forward & Backward Chaining

Forward chaining

Bottom up Approach → Known fact → Goal

Backward chaining

Top down Approach → Goal → Known fact

Resolution

(*) To prove a proposition F by resolution

eg: 1. John likes all kinds of food.
2. Apples are food
3. Chicken is food
4. Anything that anyone eats and isn't killed by is food.
5. Bill eats peanuts and is still alive
6. Sue eats everything Bill eats.

Prove Using Resolution: John likes Peanuts.

1. $\forall x$: food(x) → likes(John, x)
2. food(apple)
3. food(chicken)
4. $\forall x$: ($\exists y$: eats(y, x) $\wedge$ ¬ killed by(y, x)) → food(x)
5. A. eats(Bill, peanuts) B. alive(Bill)
6. $\forall x$: eats(Bill, x) → eats(Sue, x)
7. $\forall x$: $\forall y$: alive(x) → ¬ killedby(x, y)

Resolution Proof

¬ likes(John, Peanuts)　　¬ food(x_1) ∨ likes(John, x_1)

¬ food(peanuts)　　¬ eats(y_4, x_4) ∨ killedby(y_4, x_4) ∨ food(x_4)

¬ eats(y_4, peanuts) ∨ killedby(y_4, peanuts)　　eats(Bill, Peanuts)

killed by(Bill, peanuts)　　¬ alive(x_8) ∨ ¬ killedby(x_8, y_8)

¬ alive(Bill)　　alive(Bill)

$\boxed{}$

REAL WORLD APPLICATIONS

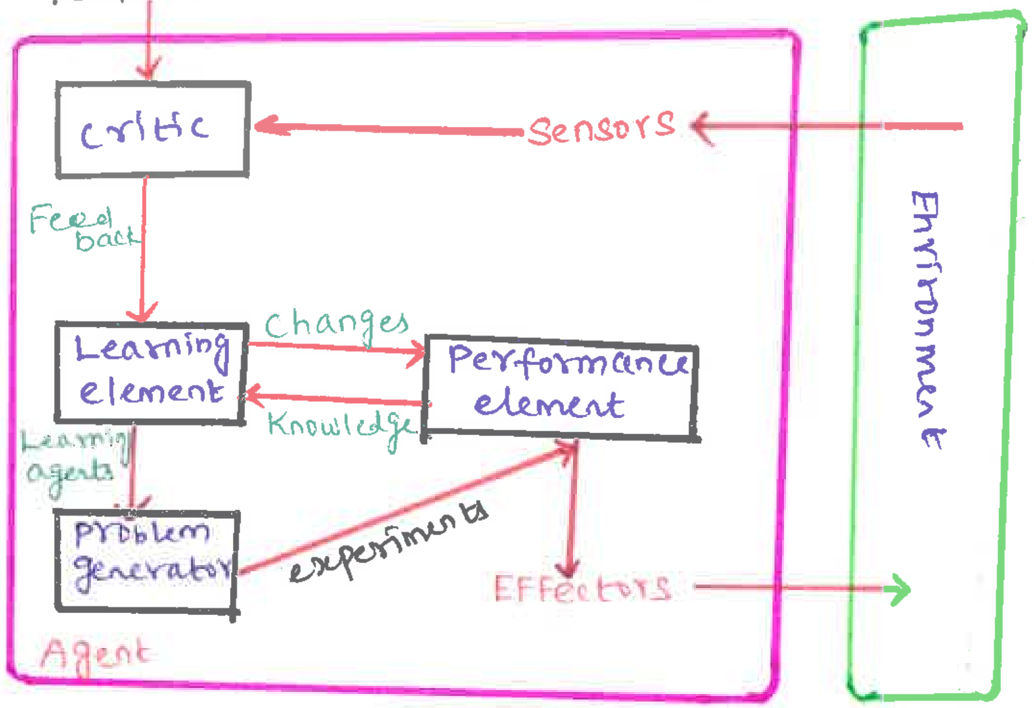
KNOWLEDGE BASE FOR FOL (WUMPUS WORLD)

1,4	2,4 P?	3,4	4,4
1,3 W?	2,3 S? B	3,3	4,3
1,2	2,2 V P?	3,2	4,2
1,1 A ok	2,1 B V ok	3,1 P?	4,1

Learning From Observation:

* Learning agents
* Types of Learning.

Learning agents:



Types of learning:

Supervised learning → Correct Answer Each Training

Unsupervised learning → Dont know anything

Reinforcement learning → Get occasional Rewards

Inductive learning:

Information provided by feedback

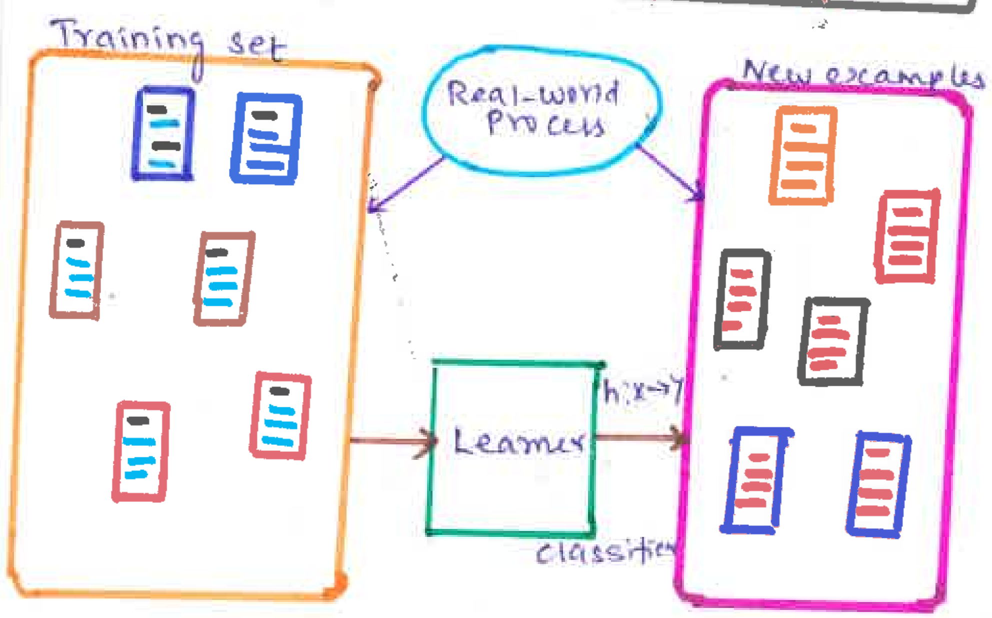
Instance Space X: Set of all Possible objects described by attributes. (Often called features)

Target Function: Mapping From attributes to Target feature. (often called label)

Hypothesis Space H: Set of all classification rules h_i, we allow

Training Data D: Set of instances labeled with Target feature

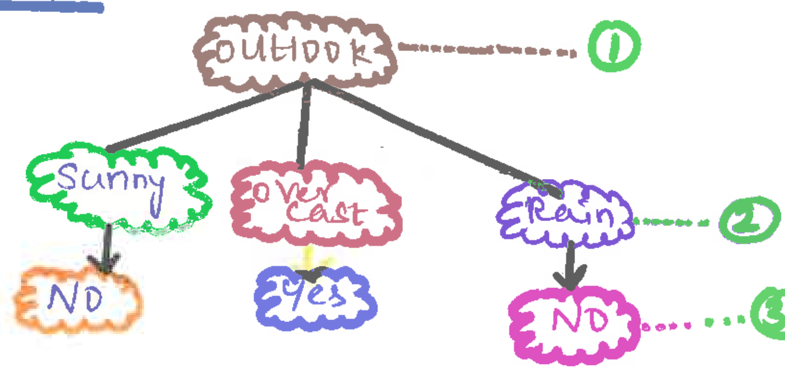
Food	Chat	Fast	Price	Bar	Big Tip
Great	yes	yes	normal	no	yes
Great	no	yes	normal	no	yes
Mediocre	yes	no	high	no	no
Great	yes	yes	normal	yes	yes



Decision Tree:

* Input → Set of attributes
* Output → returns a decision.

Example:



* Each Node Test as attribute
* Each branch corresponds to an attribute value.
* Each Assign classification.

Training Example:

Day	Outlook	Temp	Humidity	wind	Tennis
D1	sunny	Hot	high	weak	NO
D2	sunny	Hot	high	Strong	NO
D3	overcast	Hot	high	weak	yes
D4	Rain	mild	high	weak	yes
D5	Rain	cool	Normal	weak	yes
D6	Rain	cool	Normal	Strong	NO
D7	overcast	cool	Normal	Strong	yes
D8	sunny	mild	high	weak	NO
D9	Sunny	cool	Normal	weak	yes
D10	Rain	mild	Normal	weak	yes
D11	Sunny	mild	Normal	Strong	yes
D12	overcast	mild	high	strong	yes
D13	overcast	HoT	normal	weak	yes
D14	Rain	mild	high	strong	NO

* The Decision tree to represent learned target functions:
* Each internal node tests as attribute

if outlook = sunny and humidity = normal

if outlook = overcast

if outlook = rain and wind = weak.

Applications of Decision Trees:

* Instances describable by a fixed set of attributes and their values.
* Target function is discrete valued
  * 2-valued
  * N-valued
  * Approximate continuous functions.
* Disjunctive hypothesis space
* Possibly noisy training data.
  * Errors, missing values...
* Examples:
  * Medical Diagnosis
  * credit risk analysis
  * calendar scheduling preferences.

Explanation based Learning: EBL

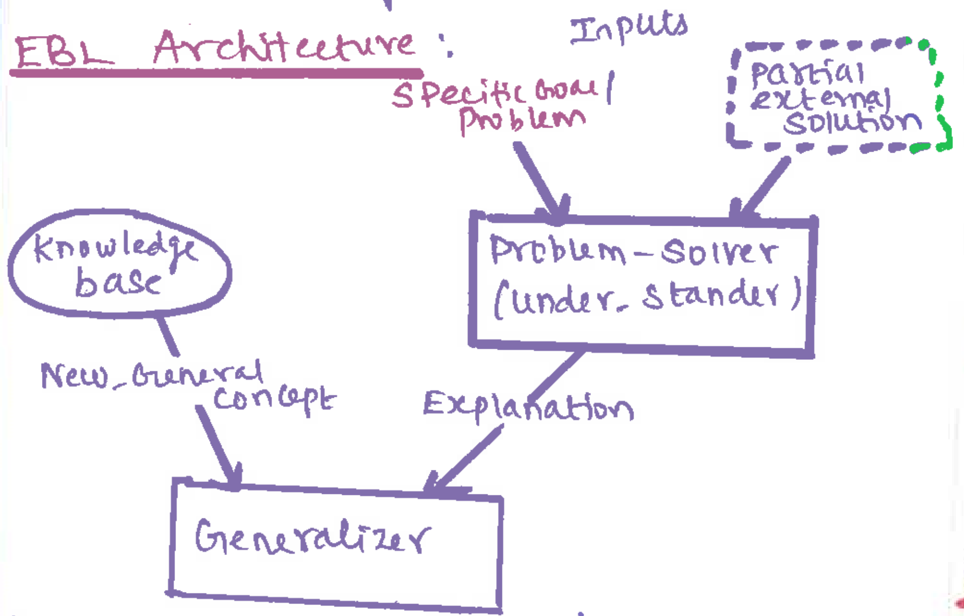
Learning general problem

Solving techniques by observing and analyzing

human solutions to specific problems

* Training Example
* A Goal concepts
* An operational criterian
* A Domain theory.

EBL Architecture:

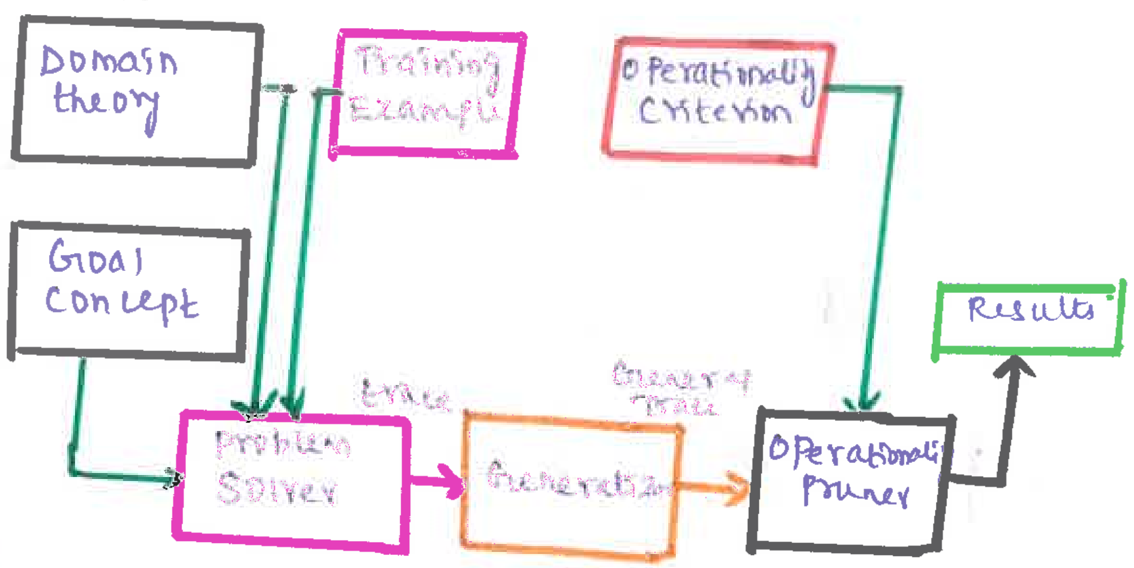


The overall architecture of the EBL Learning Method.

EBL algorithm:

* The two steps of EBL algorithm are
  1. Explain → explain about the components
  2. Generalize → explain about Goal concept.

EBL System Schematic:



Statistical learning:

* learning of uncertainity
* Agent → handed → probability → uncertainity

Baye's Rule:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Posterior P(A) = True Given P(B) = True

likelihood probability B=True; A=True

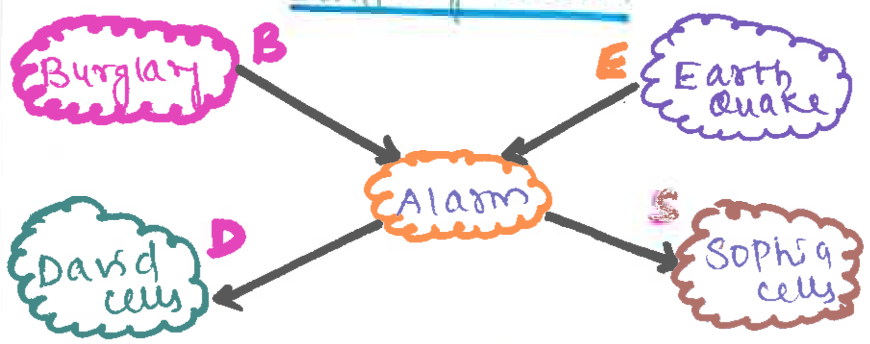
prior P(A) = True

marginalization P(B) = True

Example candy Flavor Problem

Real Time example for Statistical learning:

Burglary alarm



Prove with the key of truth table and probability.

Learning with complete data:

* likelihood Formula
* Max likelihood Formula.

Likelihood formula:

It's actually easier to work with the log-likelihood:

$$L(d|\theta) = \log P(d|\theta)$$

$$= \sum_{j=1}^{N} \log P(d_j|\theta)$$

$$= c \log \theta + l \log (1-\theta)$$

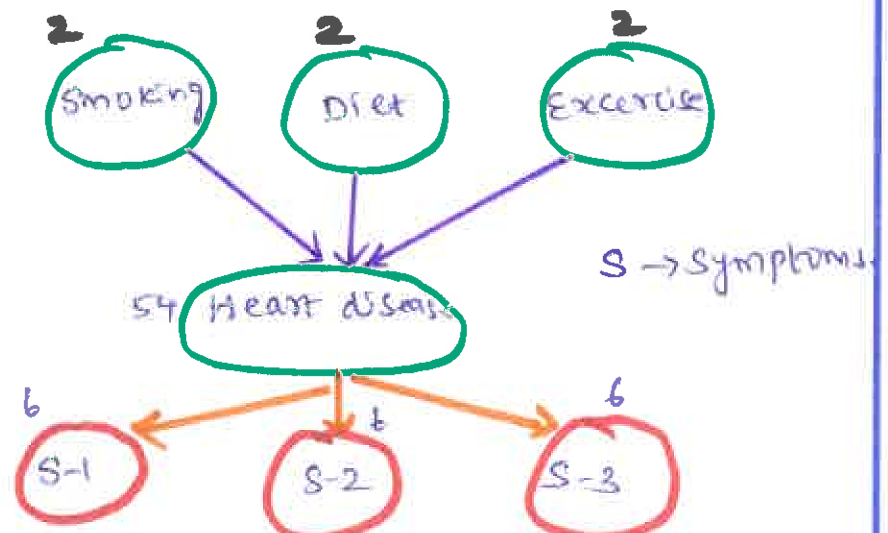
Maximizing log-Likelihood:

$$\frac{dL(D|\theta)}{d\theta} = \frac{c}{\theta} - \frac{l}{1-\theta} = 0$$

$$\Rightarrow \theta = \frac{c}{c+l} = \frac{c}{N}$$

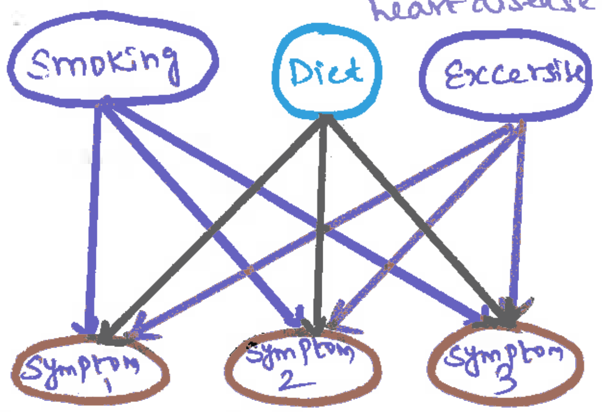
Learning with hidden variables:

* Real world problems have hidden variables
* No training data available on hidden variables
* Model cannot be built without training data
* Inference cannot be done without model
* How to learn models with hidden variables



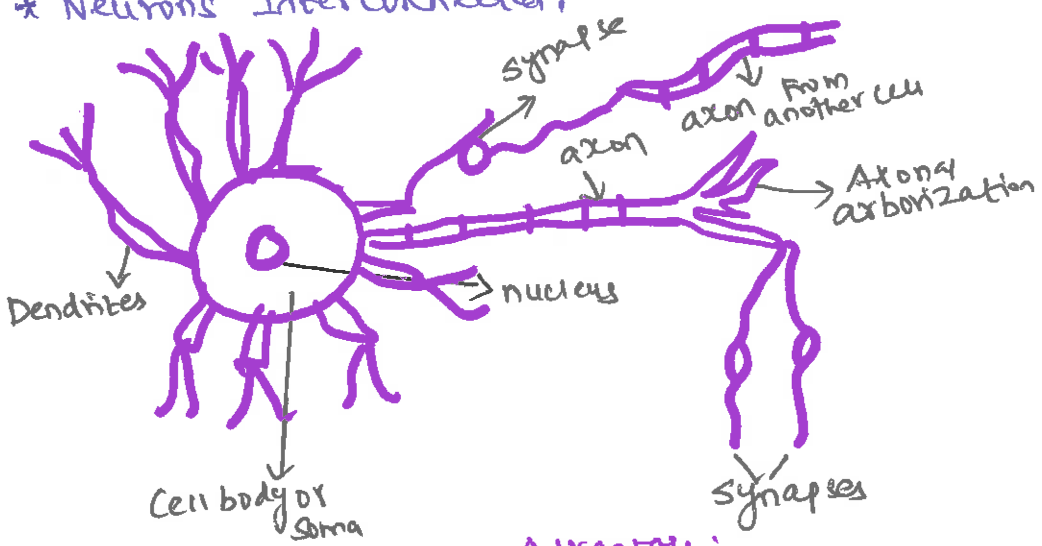
A Simple diagnostic network for heart disease.

The equivalent network with heart disease removed.



Neural Networks :

* Like human brain
* Neurons Interconnected.



* Dendrites – Inputs
* Cell Nucleus – Nodes
* Synapse – weights
* Axon – outputs

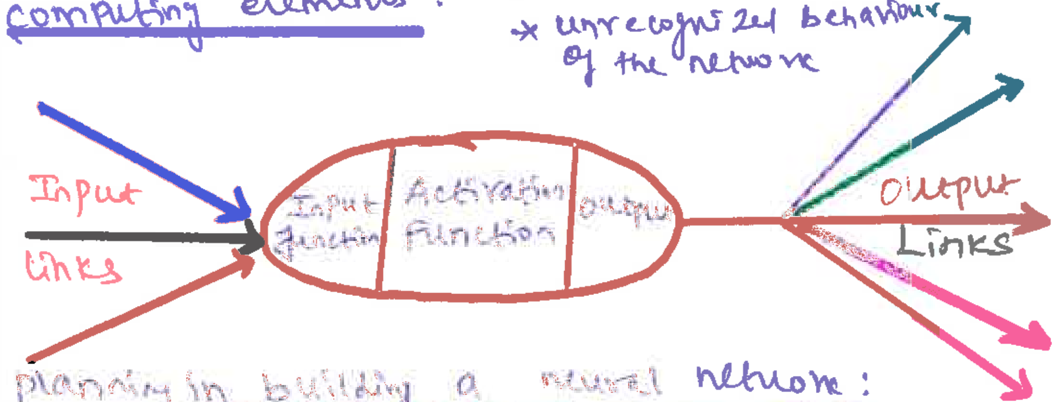
Advantage :

* Storing data on entire network
* Having memory distribution
* Having fault tolerance

Disadvantage :

* Hardware dependance
* unrecognized behaviour of the network

computing elements :

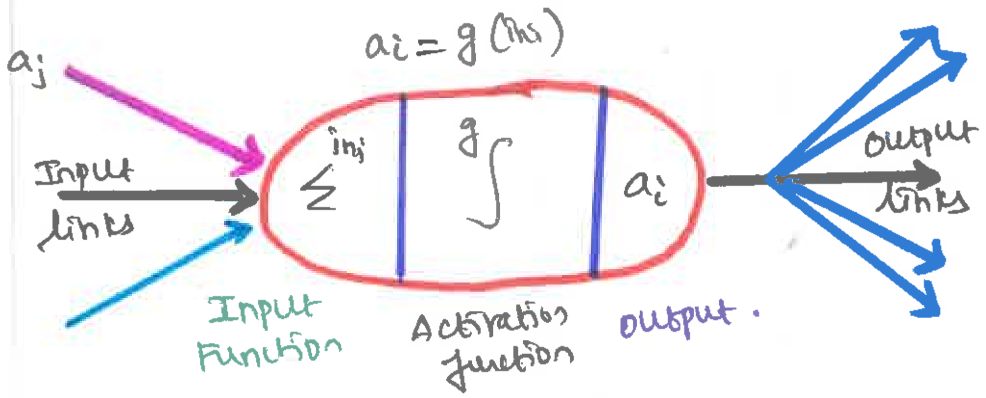


planning in building a neural network :

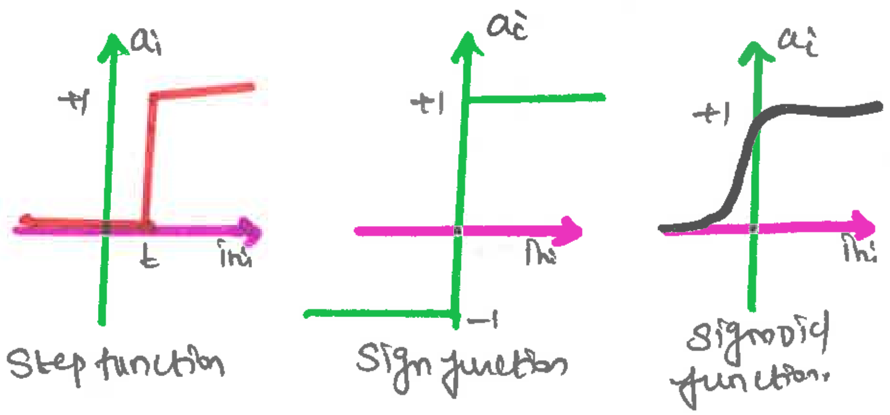
* The number of units to use
* The type of units required
* connection between the units

A computing unit :

* Now in more detail but for a particular model only.



Activation functions :



Different activation function for units :

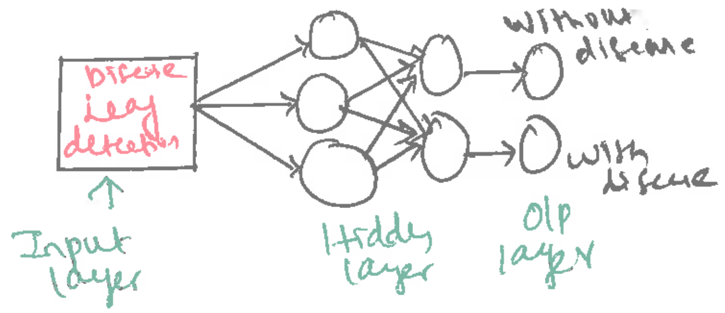
$$\text{Step}(x) = \begin{cases} 1, & \text{if } x \geq t \\ 0, & \text{if } x < t \end{cases} \qquad \text{Sign}(x) = \begin{cases} +1, & \text{if } x \geq 0 \\ -1, & \text{if } x < 0 \end{cases}$$

$$\text{Sigmoid}(x) = \frac{1}{1+e^{-x}}$$

Types of Neural Networks :

* Feedback ANN → o/p returns into the network
* Feed Forward → consists of 2 Input layer and o/p layer.

Example : (Real time applications).

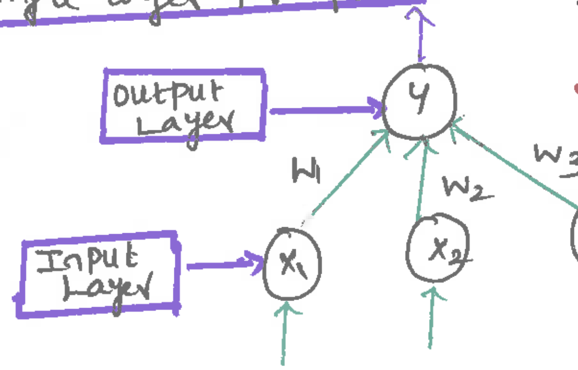


Perceptron network :

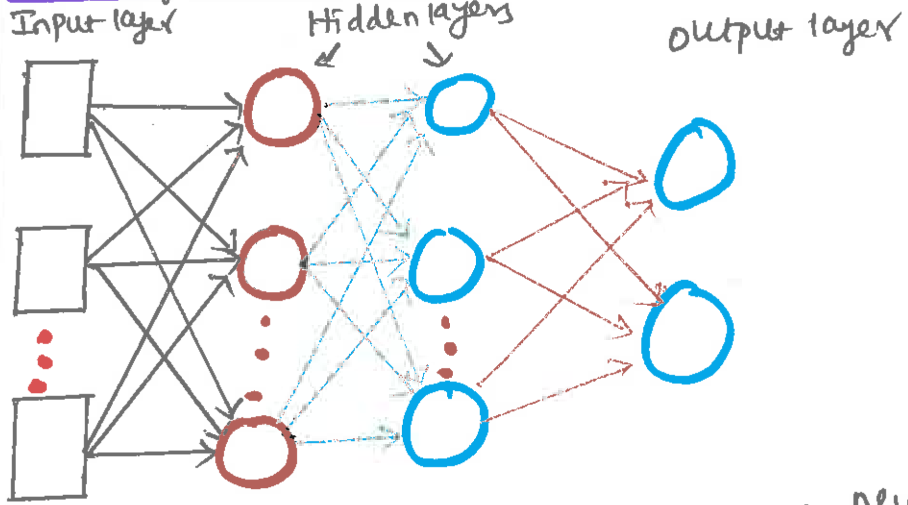
Perceptron → 4 para

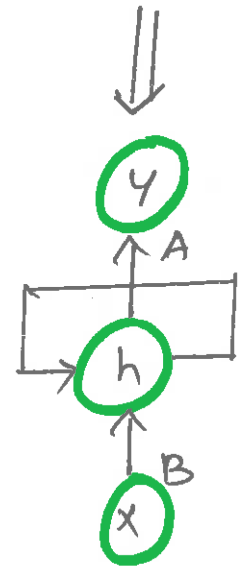
* Input and output layer
* weight bias
* Net sum
* Activation function

Single layer perceptron :



Multi layer perceptron :





Recurrent Neural network :

Why Recurrent neural network?

* cannot handle sequential data
* considers only the current input
* cannot memorize previous Inputs.

Applications : (RNN)

* Image captioning
* Time series prediction
* Natural language processing
* Machine Translation

Types of RNN :

* One to one
* One to many
* many to one
* many to many

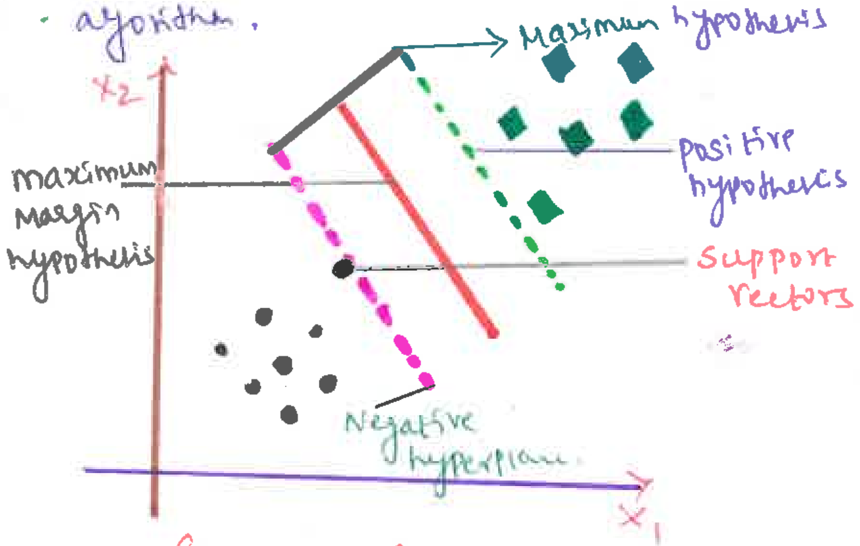
Kernel Machines:

* Single layer networks: → Simple and efficient learning algorithm, limited expressive power
* Multi layer networks: → more expressive

Support Vector Machine:

Supervised learning

* It is popular algorithm.



Graph of SVM

Real Time Example:

Cat ····· Input → Model Training → Prediction → Output
Training data → Model Training
New data → Prediction
Prediction ····· Predict earlier
Dog — Input

* SVM used for face detection
* Image classification
* Text categorization ... etc

Types:

1. Linear .... Used for linearly separable data
2. Non_linear ...... Used for non_linear separable data

Reinforcement learning:

* The problem is this: without some feedback about what is good / what is bad
* The agents will have no grounds for deciding which move to make
* The agent needs to know: Something good has happened when it wins / Something bad has happened when it loses
* This kind of feedback is called a reward or reinforcement.

Agent
State / Reward → Agent
Agent → Action
Environment

Applications:

Applications of RL: Robotics, Control, Game playing, Chemistry, Business, Manufacturing, Finance Sector

Terms used in Reinforcement learning:

* Agent() * Environment() * Action() * State()
* Reward() * Policy() * Value() * Q value()

Element of Reinforcement learning:

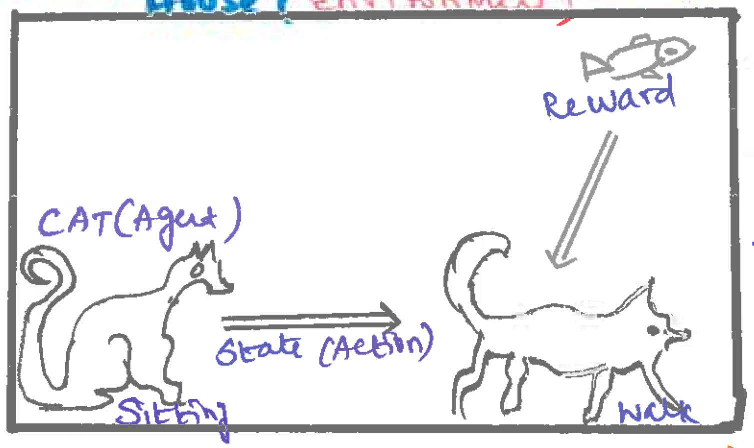
* Policy
* Reward Signal
* Value function
* Model of the Environment

Types of RL learning:

* Positive Reinforcement
* Negative Reinforcement

Example of Reinforcement learning:

House (Environment)



In this case,

* Your cat is an agent that is exposed to the environment.
* The agent reacts by performing an action transition from one state to another state.
* The reaction of agent is an action.
* For example, your cat goes sitting to walking.
* They may get a reward or penalty in return.

Types of Learning:

* Supervised learning
  * Classification → Naive Bayes, SVM, Logistic Regression
  * Regression → Linear, Non_linear
* Unsupervised learning
  * K_Means
  * Neural Networks
  * PCA
* Reinforcement learning
  * Supervised
  * (or)
  * Unsupervised
  * No_Prediction data

Types of data in ML:

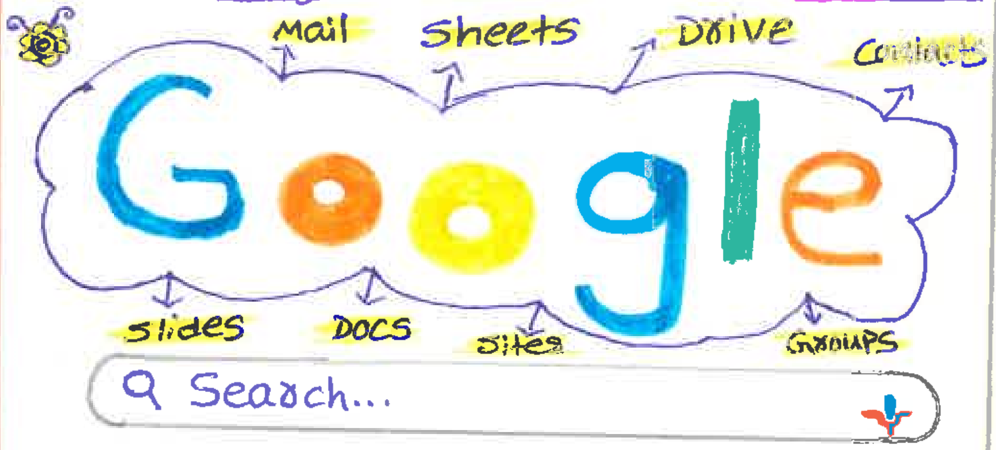
Supervised ML → Labelled Data
Unsupervised ML → Unlabelled data

AIM: Supervised → calculate outcomes; Unsupervised → Discover underlying patterns; Reinforcement → Learn a series of action

* Introduction To Search Engine Optimization. *

Different Search Engine :-

1) A search engine is a web-based tool that enables users to locate information on the World Wide Web.
2) Popular examples of search engines are Google, Yahoo!, and MSN Search.



3) Yahoo was founded by → David Filo
→ Jerry Yang
4) Google was founded by → Larry Page
→ Sergey Brin

Uses of SE :-

* SE utilize automated software applications that travel along web, follow like Page to Page.
* SE search their own databases when a user enters in a search to find related documents.

Types of SE :-

1) Crawler based Search Engines [use "spiders" or "crawlers"]
2) Search directories or indexes [Depends on human editors]
3) Hybrid Search Engines [Both crawler & search directories]
4) Meta Search Engines [via the WWW.]

Evaluating Search Engine :-

Measures for a search engine :

* How fast does it index
  → Number of documents/hour
  → Average document size
* How fast does it search
  → Latency as a function of Index size.
* Expensiveness of query language
  → Ability to express complex information needs
  → Speed on complex queries

Measuring User happiness :

* Issue : who is the user we are trying to make happy?
  → Depends on the setting
* Web Engine : User finds what they want and return to engine.
  → can measure rate of return users.

Accuracy :-

* Given a query an engine classifies each doc as "Relavent" or "Irrelavent".
* Accuracy of an engine : the fraction of these classifications that is correct.

Search Techniques :-

* Phrase searching
  • Instead of
    → American English
* Try
  → American English
* Truncation
  * Instead of
    * Machine
* Try
  • Machine? for machine, machinery.

Types :-
1) Boolean Searching
2) Subject Searching
3) phrase searching

Wildcard Search :-

* If you want both
  - Woman
  - Women
* Try
  - wom*n

Input =	ba aab ab
Pattern =	xx ba xxxx ab
Output :	true

Boolean Searching :-

* Boolean Search is query technique that utilize Boolean logic to connect individual keyword within single query.

"Phonetics" or "Phonology"

"Phonetics" and "Phonology"

"Phonetics" Not "Phonology"

Real Time Application :-

* Website Design - Layout And Keyword Optimization *

Design and Layout

what type of information goes on a web site?

* Contact information (phone No, email)
* Business hours
* Background information about your business
* product / service information

The URL is composed of 3 parts

→ The protocol * http://
→ The domain name * www.hispanicbusiness.com
→ The pathname * /magazine
→ A web page is written in a special code called "HTML"

[HTML - "Hyper-text markup language"]

Most sources today offer Both domain and web Hosting:

Ex: [* www.godaddy.com
* www.register.com]

Free domain with annual purchase	Standard web Hosting $9.96mo w/annual purchase	Advanced web Hosting $18.30mo w/annual purchase	Pro web Hosting $29.18mo w/annual purchase
Disk space (GB)	7	15	30
Monthly Data transfer (GB)	150	400	500
FTP Accounts	3	6	15
E-mailboxes	50	100	200
Directory pointers	20	40	50

Keywords optimization

* A keyword is a term that is used to match with the query a person enters into a search engine to find specific information.
* Such phrases may be called search phrases, keyword phrases, query phrases or just keywords.

keyword frequency

* This is calculated as how often does a keyword appear in a website title or description.
* Since on some engines if you repeat a word too many times, you are be penalized for "spamming" or keyword stuffing.

Keyword weight

* It refers to the number of keywords appearing on your webpage compared to the total number of words appearing on that same page.

Keyword proximity

* It refers to the placement of keywords on a web page in relation to each other or, in some cases, in relation to other words.

Keyword prominence

* It is a measure of how early or high up on a page, the keywords are found.

Metatags optimization

There are two important metagas:

- Meta description tags
- Meta keyword tags

* For Google, adding the description meta tag does not result in a boost in the search engine results.

What DO the Metatags Look like

Ex:-
* <meta name = "keywords"
* content = " KEYWORD1 KEYWORD2 KEYWORD KEYPHRASE1 etc. about 30 to 40 unique words">
* <meta name = "description"
* content: "An accurate, keyword-rich description about 150 characters">

Real-time application
1. zig-zag Layout
2. F Layout
3. Full-screen photo
4. Grid layout
5. one-column layout

Headlines & thumbnail layout

Play Store

SEO - SEARCH ENGINE OPTIMIZATION

Properties Of Seo

* A "+" before a word in a search will locate for documents which contain the word.
* A "–" before a word will exclude that word from search
* Placing words b/w " " marks will search for phrase b/w the quotes
* Using the "or" b/w search phrase will search each term separately

Examples

→ "+ Blue" results contain word black & word blue

→ "Black Blue" - The document will be returned which contain the word Black but not the word blue

→ "Black Blue" - The result contain the phrase black blue (placed Together)

→ "Black or Blue" - The result contain the term either black or blue

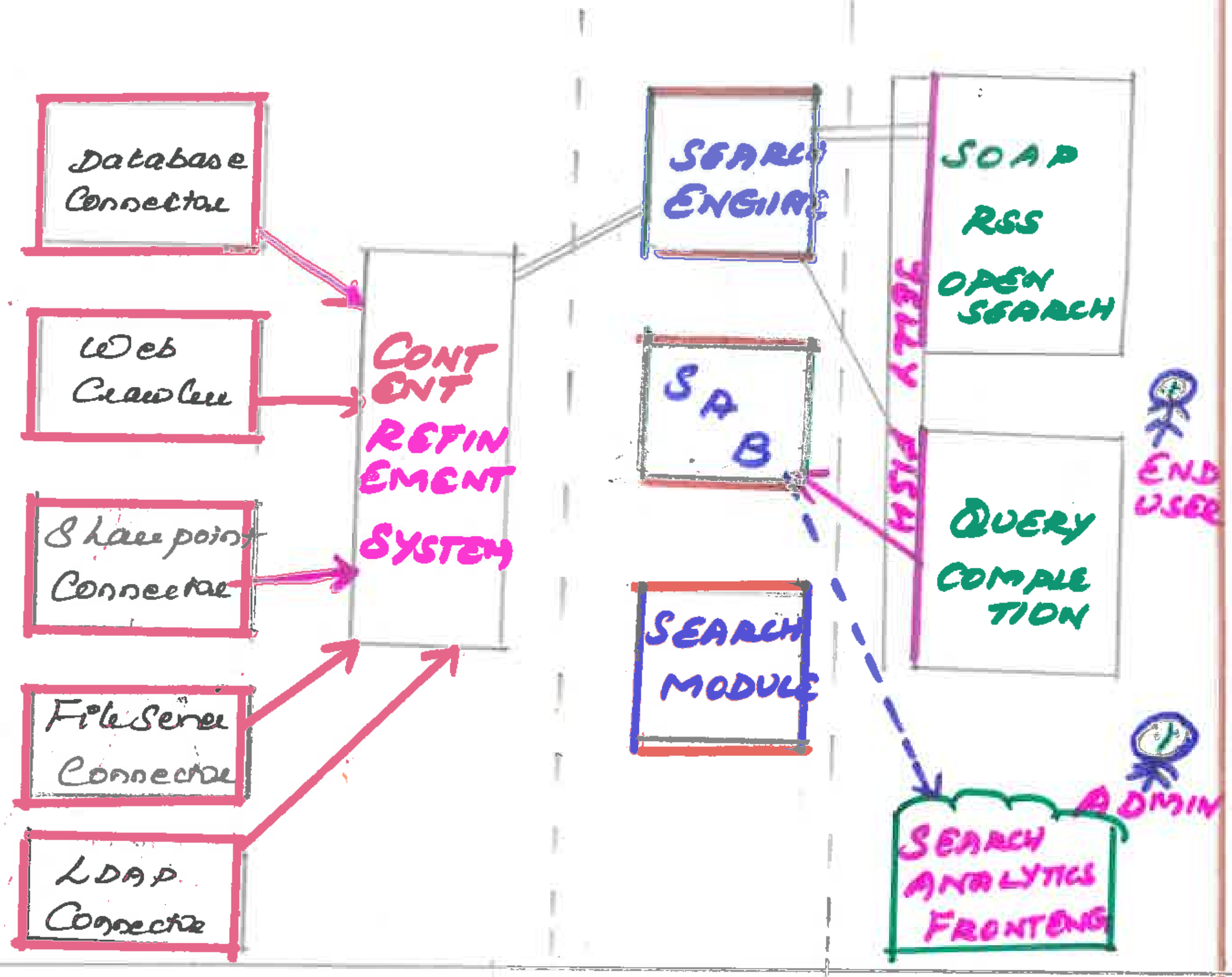
ADVANTAGES

* Easy to Square
* Sorting and ability

DISADVANTAGES

* Too many Website visitors
* Too many website loads

SEARCH ENGINE ARCHITECTURE



WEBSITE DOMAIN

→ A website is a communication tool that allows people to share information

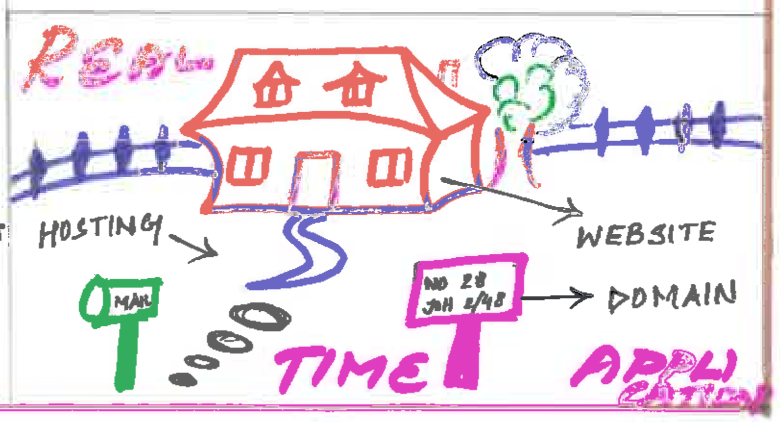
→ A website gives your business visibility and credibility

→ A website is one of the most cost effective forms of advertising

- Brochures
- News letters
- Customer Info. Sheets
- Hand outs

URL COMMON PARTS

* The Protocol
  - http://
* The domain name
* www.amazon.com
* The Path name
  - /magazine



Title optimization:-

* An HTML TITLE tag is put inside the head tag
* This is the one place on a webpage where your keywords must be present
* Correct use of keywords in the title of every page of your website is extremely important to Google - particularly for the homepage

Designing the title of a web page:-

* The title shouldn't consist of more than about 9 words or 60 characters
* use keywords at very beggining of the title
* Do not include your company name in titles unless your company name is very well known

Best Practices for creating Titles:-

* use more specific variations to your primary keyword phrase on your specific product, service, or content page.
* Make sure the "<title>" tag is first element in the <head> section of your page - This make it easier for Google to find the page

Anchors optimization:-

* Use descriptive anchor text for all your text links. Most search engines consider anchor text of incoming links when ranking pages. Here is an example:

```
<a href = " otherpage.htm " title= " Anchortitle"
 > Anchor Text </a>
```

*
```
<a href = " otherpage.htm " title= " Anchor
 Title"> Anchor Text </a>
```

⇒ **Some of Important points to note about anchors:-**

⇒ The Anchor plays a very important role in the "Titles" and is seen by most of search engines

⇒ The Anchor title should have an appropriate keywords.

⇒ Anchor title helps the site visitors using a balloon, and displaying written text.

Anchors verification for website:-

⇒ Another example of an anchor could be as follows:

```
<a href="other page.htm " title=" Anchor Title">
<im g src="im age.gif" alt= "keywords"/></a>
```

⇒ In this case, Anchor Text has been replaced by an image.

⇒ So, while using an image in place of an anchor text, it should be checked that you have put alt tag properly.

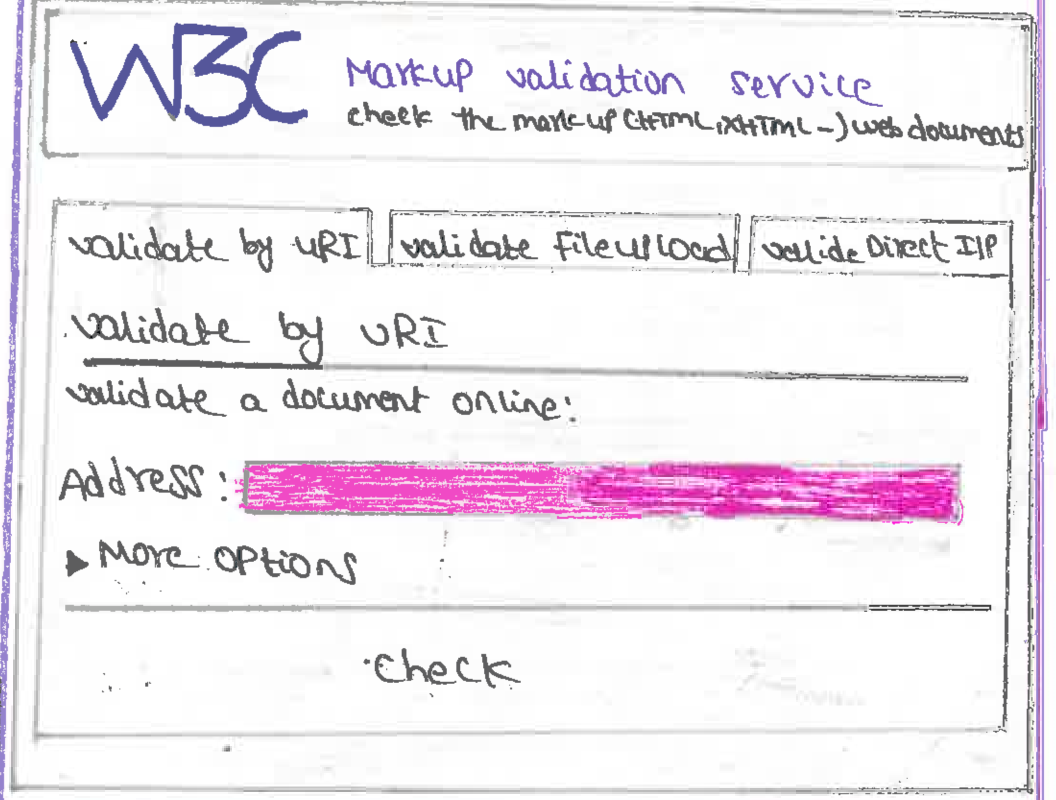
XHTML validation:-

⇒ validation is a method of verifying that we have mad no syntax or rules violations on our web pages

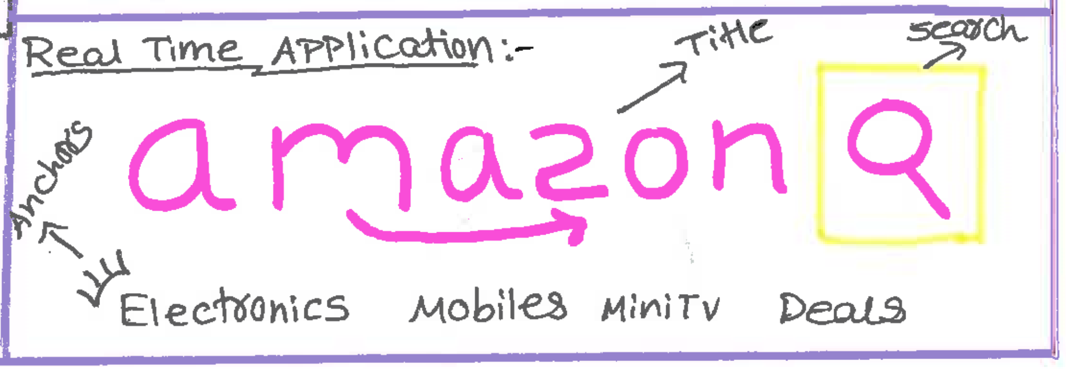
⇒ This step is important, as it helps ensure that our websites display properly and consistently to our visitors.

example:-

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml-transiti
 -onal.dtd">
```



Real Time Application:-



* FUTURE OF ARTIFICIAL INTELLIGENCE *

FUTURE OF AI

Health Care → Automated Vehicles → Security & Defense → Manufacturing → Education → Entertainment → Banking & Finance → Workplace

Artificial Intelligence future :-

→ A PWC report estimates that AI will contribute $15.7 trillion to global economy by 2030.

1. Health care :

→ AI robots would be employed.

→ To perform complex surgeries.

→ High degree of precision.

2. Autonomous Vehicles :

→ Autonomous vehicles are equipped with multiple sensors.

→ like Cameras, radars and lidar.

→ AI is projected to have a valuation of $ 127 billion by 2025.

3. Security and Defence

→ AI are often embedded into weapons and Surveillance Systems.

→ Securing the borders of the country can be delegated to AI robots, UAV's etc.

4. Manufacturing:

→ AI optimizes manufacturing supply chains.

→ Helping companies anticipate market changes.

→ future process would be able to inspect, improve, and quality check

→ No need of any human intervention.

5. Education

→ The future of classroom is digital.

→ It will be redefined from comfort of homes

→ Personalized according to every students with AI systems.

6. Entertainment :

→ OTT's like Aha, Amazon prime etc are increasing user base.

→ In future, AI will be predicting according to your mood and display content.

7. Workplace :

→ Businesses are using AI

→ To enhance productivity of their employees

→ One example is chatbot.

8. Banking and finance :-

→ AI is best way for banking

→ It brings facility of advanced data analytics to combat fradulent transactions.

→ Improve compliance.

* The ways in which AI can be implemented in different industries are left only to one's imagination

* There is one thing that is for sure which is :

"The future would be an exciting time to live in."

SIMATS

SCHOOL OF ENGINEERING

Engineer to Excel

Approved by AICTE | IET-UK Accreditation

Saveetha Nagar, Thandalam, Chennai - 602 105, TamilNadu, India