Pokhara University
Semester : Fall
Year : 2022

1.a. What is Artificial Intelligence ? Explain the current trends of AI and its impact on society.

Artificial Intelligence (AI) refers to the develop-ment of computer systems that are capable of performing tasks that would typically require humann intelligence.

It involves the creation of intelligence machine that can analyze, reason, learn, and make decisions similar to humans.

Some perspectives of AI

1. Acting Humanly (The Turing Test Approach)
2. Thinking Humanly (The cognitive modeling Approach)
3. Acting Rationally (The Rational Agent Approach)
4. Thinking Rationally. (The laws of Thought Approach)

Here are the some current trends of AI

1. Minimize Uncertainty with AI.
2. AI Ethics and Privacy concerns:
   As AI use grows, ethical issues like privacy breaches and bias become important. AI in courtrooms and facial recognition present both opportunities and risks. It's crucial to deve-lop ethical AI to avoid biased application.
3. AI in Transportation:

AI will revolutionize transportation with autonomous vehicles, fleet management, and improved road safety measures.

4. AI in Hollywood:
AI is used in movie production for visual effects, motion capture, and script analysis. Future applications could include AI-written scripts, AI-directed films, and algorithm predicting box office success.

Impact of AI on Society:

- Job displacement
Automation through AI can lead to significant job losses in sectors like manufacturing, retail, and even some professional services. Routine and repetitive tasks are the most vulnerable

- Job Creation
While some jobs are lost, new roles may emerge in AI development, maintenance, and oversight. There will be a growing need for skills in data analysis, machine learning and AI ethics.

- Remote work
AI tools support remote collaboration, enabling more flexible work arrangements but also blurring the lines between work and pro

- Healthcare and Education
  AI helps in diagnosing diseases, predicting patient outcomes, and personalizing treatments. This leads to better healthcare and improved patient care.

  AI-powered tools provide personalized learning experiences, helping students learn at their own pace. Virtual tutors and AI-driven educational platforms make education more accessible and engaging.

- Security Risks
  AI systems can be vulnerable to hacking and misuse, leading to potential security breaches and malicious attacks.

1.b Suppose you are traveling from pokhara to Kathmandu. You have three agents to travel by flight or by bus or by bike. Suppose your preference is to reach Kathmandu as soon as possible. which agent do you choose? what is type of this agent? write the details of task environment of the agent you chose.

Here the priority is to reach Kathmandu as soon / quickly as possible, I would choose to travel by flight.
The type of agent in this scenario is a goal-based agent. This is because the primary goal is to reach Kathmandu as soon as possible, and the agent selects the mode of transport that best achives this objective.

I choose the flight agent which is a goal based agent and it focus on

- Object oriented:
The agent focuses on achieving the primary objective of reaching the destination quickly
- Decision-Making
The agent makes decisions based on the goal, such as choosing the fastest mode of transport.
- Flexibility:
The agent adapts its actions depending on available

options and conditions to achieve the goal efficiently.

The task environment for this agent can be defined using the performance measure, Env, Agent, Actuators, Sensors i.e, PEAS frame work.

2. Performance Measure includes
   - Travel time : The total time ~~duration~~ duration departure to arrival
   - Punctuality : Adherence to the scheduled departure and arrival times.
   - Availability: Frequency and timing of available flights
   - Cost Efficiency : Balancing the cost of the flight with the speed of travel

3. Environment
   The environment include all external factor influencing the journey.
   - Airports: Conditions at Pokhara Airport and Tribhuvan International Airport in Kathmandu.
   - Flight schedules: Timetable and frequency of flights between the two cities.
   - Weather Conditions: weather reports that can affect flight schedules and safety
   - Air traffic control

3. Actuators

Actuators are the mechanisms through which the agent interacts with the environment.

- Booking system : Online or offline system to book the flight
- Check in process : Procedures for check-in, security checks, and boarding
- Flight operations : The actual flight carring the traveler from Pokhara to Ktm.

4. Sensors.

The agent uses sensors to perceive the environment, which include real-time flight-status updates, weather reports, and airport information systems.

First of all agent searches for available flights and compare options then book ticket and checks security & boarding. The agent travels from pokhara to Kathmandu by air.

Q.2.a What are the constraint satisfaction problems? Solve the following crypt-arithmetic problem.

In artificial intelligence constraint satisfaction is the process of finding a solution to a set of variables. Vi (V_1, V_2, V_3, ... V_n), where each variable has a domain of allowed value.

There is no specified rule to defined the procedure to solve the constraint satisfaction problem (CSP). The main feature of CSP are:

- It is high level description of a problem
- A model for the problem is represented by a set of variables and their domain.
- The problem is stated as constraints specifying relationship between the variable.
- The constraints only specify the relation-ship with specifying a computational proce-dure to enforce the relationship.

The CSP consist of two steps.

i) constraints are discovered

ii) propagate the constraints throughout the system.

If there is no solution are found a new guess is made about some values, added as constraints and the propagate.

Given,

Crypto - Arithmetic problem is

```
  B A S E
+ B A L L
---------
G A M E S
```

soln:

Here Each Alphabet can take only one distinct value i-e, $A \neq B \neq C$

we assume the values between 0 to 9

step 1: Let $E = 3$, $L = 5 \rightarrow S = 8$

step 2: $S = 8$, $L = 5 \rightarrow E = 3$

Here carry occur (8+5=13 ie. carry=1)

step 3: $A = 4 \rightarrow M = 4+4+1 = 9$

step 4: and $B = 7$ and $G = 1$ (7+7 is 14 and again carry occur so G=1)

step 1:

```
  B A S 3
+ B A L 5
---------
G A M E 8
```

Step 2:

```
  B A 8 3
+ B A 5 5
---------
G A M 3 8
```

step 3:

```
    1
  B 4 8 3
+ B 4 5 5
---------
G A 9 3 8
```

step 4:

```
    1
  7 4 8 3
+ 7 4 5 5
---------
1 4 9 3 8
```

Hence the required solutions are:

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

✓

Q.2.b. What are the problem of depth first search and breadth first search? How are these problems resolved? Explain with a suitable example.

In depth first search algorithm, it starts from the root node and follows each path to its greatest depth node before moving to the next path.

problems:

- DFS can go infinitely deep into a branch if the search tree is very deep or infinite, which can lead to non-termination.
- There is the possibility that many states keep re-occuroing, and there is no guarantee of finding solution.

To solve these problem, Depth limited search algorithm is introduced. Depth limited search can solve the drawback of the infinite path in DFS. Also we can use iterative Deepening Depth-First search (IDDFS). This ensure completeness and avoids getting stuck in infinite loop.

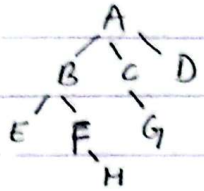
Breadth First search (BFS) Problems are:

- Memory space: BFS requires storing all nodes at the current level, which can be highly memory-intensive if the search tree is wide.

- Time complexity : BFS can be slow for trees with many branching factors because it explores all neighbors level by level. Take more time if the goal is located deep in the tree.

To solve above problem Best first algorithm is introduced. It is the combination of depth first search and breadth-first search algorithms. It uses the heuristic function and search. uses ~~hei~~ heuristic to prioritize which nodes to explore. reducing time and memory usage by focusing on more promising paths.

Consider a graph :



Find the path from A to H

Using DFS:

(1.) problem : DFS may explore A → B - E - F and continue along F's neighbors before reaching H.
If DFS doesn't track visited nodes, it may follow H → G → H endlessly. i.e, A → B → E - F → H → G → H

solution: Use a visited set to avoid revisiting nodes

DFS Traversal : A → B → E

Backtrack to B → F → H (solution is Found)

DFS Tree Traversal with a visited set

A → B → E → F → H

visited visited visited found target.

Path Found by DFS: A → B → F → H

2.) Example (BFS with problem and solution)

problem: High memory usage.

BFS explores all nodes at the same depth level, which can consume significant memory for large graphs. For example:

Traversal Levels:

Level 1: A → (Explore B, C, D)

Level 2: (Explore E, F, G)

Level 3: (Explore H)

solution: Use Bidirectional BFS

search simultaneously from A (start) and H (target) to reduce the number of nodes explored.

start BFS from A and H simultaneously

Forward BFS: A → B → F → H

Reverse BFS: H → F → B → A

The search meets at F or B reducing memory usage.

path Found by BFS: A → B → F → H

Q.3 a Why is A* search better than best-first search? Explain with a suitable example.

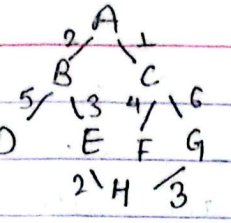
Here Both A* search and Best-first search are informed search algorithms that use a heuristic function to guide the search. A* is better than best-first search because it combines the heuristic with the cost incurred so for, making it both optimal and complete. (under certain conditions).

Key Difference Between A* search and Best First Search

S.N	Best First search (Greedy)	A* search
1.	Uses only the heuristic h[n]	Combines h[n] with the path cost g(n) f(n) = g(n) + h(n)
2.	Not guaranteed to find the shortest path.	Guaranteed to find the sho path if h[n] is admissible consistent.
3.	May fail in infinite search spaces or loops	Gauranteed to complete if graph is finite
4.	Explores nodes that seem closest to the goal (greedy)	Explores nodes based on lowest total estimated cos

Hence Best-First search is greedy algorithm which only see the h[n] less value without considering actual costs and may get stuck in suboptimal paths or fail entirely. But A* balances short term-heuristic with long-term cost and giv guarantees the optimal solution.

For example



Heuristic ($h(n)$):

A=6, B=6, C=4, D=8, E=2
F=3, G=4, H=0 (Goal)

Best-First search

1. start at A : $h(A) = 6$
2. Choose C (lowest h; $h(C) = 4$)
3. From C, choose F, $h(F) = 3$
4. From F, there's no direct path to H.
5. path chosen: A → C − F (fails or results in a suboptimal path).

Using A* search. $f(n) = g(n) + h(n)$:

1. start at A: $f(A) = 0 + h(A) = 0 + 6 = 6$
2. Expand neighbors:
   $f(B) = g(B) + h(B) = 2 + 6 = 8$
   $f(C) = g(C) + h(C) = 1 + 4 = 5.$
   Choose C (lowest $f(n)$)
3. From C.
   - $f(F) = g(F) + h(F) = 5 + 3 = 8$
   - $f(G) = g(G) + h(G) = 7 + 4 = 11$
   - Choose F
4. From F:
   - No direct path to H
5. Backtrack to C and explore G!
   - $f(H) = g(H) + h(H) = 9 + 0 = 9.$
   - path A → C → G → H

Optimal path Found A → C → G → H (cost: 9)

Q.3 b. Assume the following facts.

i) John likes all kinds of food.

ii) Apple and vegetable are food.

iii) Anything anyone eats and is not killed is food

iv) Anil eats peanuts and still alive

v) Harry eats everything that Anil eats.

vi) Anyone who's not killed means, he is alive.

Represent these facts in predicate logic.

soln!

i) John likes all kinds of food.

Assume that 'x' denote food.

$\forall x : food(x) \rightarrow likes(John, x)$

ii) Apple and vegetable are food.

$food(Apple) \wedge food(vegetable)$

iii) Anything anyone eats and is not killed is food

$\forall x \forall y : Anything(x) \wedge Anyone(y) \wedge eats(x, y) \wedge \neg killed(y) \rightarrow Food(y)$.

iv) Anil eats peanuts and still alive.

$\exists x \exists y : Anil(x) \wedge peanuts(y) \wedge eats(x, y) \wedge alive(x)$

v) Harry eats everything that Anil eats.

$\forall x : everything(x) \wedge eats(Anil, x) \rightarrow eats(Harry, x)$

vi) Anyone who's not killed means, he is alive.

$\forall x$: Anyone $(x) \wedge \neg$ killed $(x) \rightarrow$ alive (x)

Q.4a. Assume the facts in 3(b) and use resolution to prove "John likes peanuts".

Soln:

The resolution procedure has been illustrated below:-

to prove "John likes peanuts" we first go with the contradiction means it is False

Let us assume,

John doesn't likes peanuts.

$\exists x \exists y$: John$(x) \wedge$ Peanuts$(y) \wedge \neg$ like(x, y)

Now, or, $\neg$ likes (John, Peanuts)

above.

Conversion of FOL into CNF

1. Elliminate all implification ($\rightarrow$) and rewrite.

i) $\forall x : \neg$ food $(x) \vee$ likes (John, x)

ii) food (Apple. $\neg$ food (vegitables)

iii) $\forall x \forall x$: Anything$(x) \wedge$ anyone $(y) \wedge \neg$ [eats $(x, y) \wedge \neg$ killed(y)] $\vee$ food(y)

iv) eats (Anil, peanuts) $\wedge$ alive (Anil)

v) $\forall x \neg$ eats (Anil, x) $\vee$ eats (Harry, x)

vi) $\forall x \neg [\neg$ killed $(x)] \vee$ alive (x)

vii) $\forall x \neg$ alive $(x) \vee \neg$ killed(x)

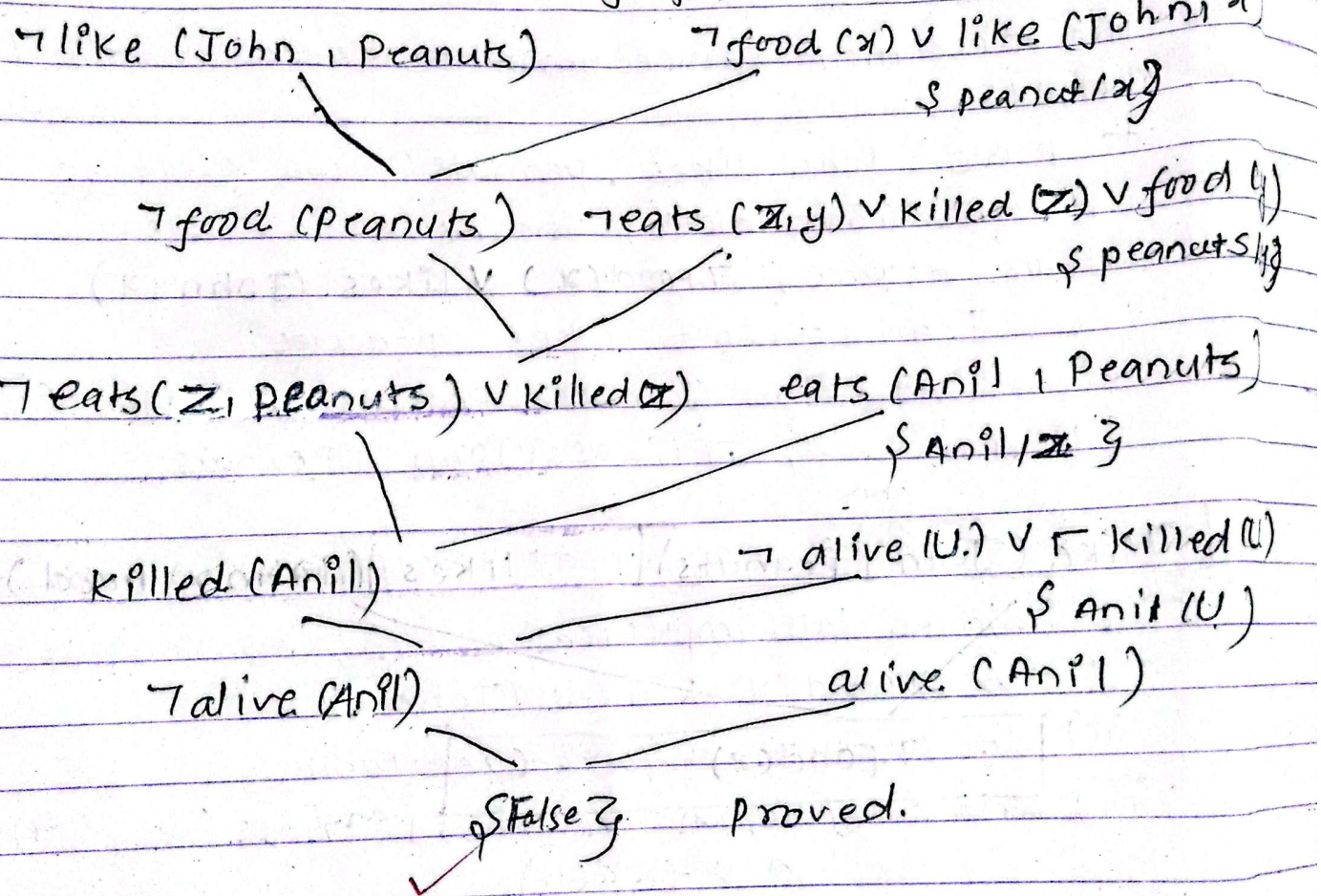
viii) likes (John, Peanuts.)

2. Drop universal quantifier:

i. ¬food(x) ∨ likes(John, x)
ii. food(Apple)
iii) food(vegetable)
iv) ¬eats(y, y) ∨ killed(x) ∨ food(y)
v) eats(Anil, Peanuts)
vi) alive(Anil)
vii) ¬eats(Anil, w) ∨ eats(Harry, w)
viii) killed(v) ∨ alive(v)
ix) ¬alive(u) ∨ ¬killed(u)
x) likes(John, Peanuts)

Step 3: Negate the statement to be proved.
¬likes(John, Peanuts)

Step 4: Draw Resolution graph



Hence the negation of the conclusion has been proved as a complete contradiction with the given set of statements.
So John likes peanuts.

Q.5a. How does support vector machine work? Illustrate with a suitable example.

Support vector machine (SVM) is a supervised learning algorithm primarily used for classification and regression tasks. SVM works by finding the optimal hyperplane that best separates the data into different classes.

Steps of support vector machine (SVM)

1. Data Representation:
The data is represented as points in a high-dimensional space, where each point corresponds to a feature vector.

2. Find the optimal Hyperplane.
A hyperplane is a decision boundary that divides the space into two classes.
The goal is to maximize the margin, which is the distance between the hyperplane and the nearest data points (called support vectors) from both classes.

3. Linear Vs Non-linear Data:
If the data is linearly separable, SVM finds a straight hyperplane.
For non linear data, SVM uses kernel tricks (like the polynomial or RBF kernel) to map the data into a higher-dimensional space where it becom-es linearly seprable.

4. Classification:

Once the hyperplane is determined, new data points are classified based on which side of the hyperplane they fall on.

For example. classifying Fruits. Imagine we want to classify fruits as apples or oranges based on their weight (grams) and size (diameter in cm)

Data Representation.

Fruit	Weight	Diameter
Apple	150	7
orange	180	8
orange	200	9
Apple	140	6

For plot the Data (150, 7) (180, 8) (200, 9) (140, 6)

x-axis → weight
y-axis → Diameter.

Find Hyperplane

SVM identifies the hyperplane that best separates the apples and oranges

- Optimal Hyperplane: A line that maximizes the margin (distance to the closest points from each class)
- Support vectors: The closest data points to the hyperplane.

Step 3: classification

For a new fruit (e.g. weight 160, diameter: 7.5).

- If it falls on one side of the hyperplane, classify as an apple
- If it falls on the other side, classify as an orange.

Q. 5 b Write down the algorithm for K-means clustering. Explain with a suitable example.

K-Means clustering is an unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here k defines the number of pre-defined clusters that need to be created in the process, as if K = 2, there will be two clusters, and for k=3, there will be three clusters, and so on.

Algorithm of K-Means Algorithm

Step1: select the number K to decide the number of clusters.

Step 2: Select random K points or centroids (It can be other from the input dataset).

Step 3: Assign each data point to their closest centroid, which will from the predefined K clusters.

Step 4: Calculate the variance and place a new centroid of each cluster.

Step 5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step 6: If any reassignment occurs, then go to step-4 else go to finish.

Step 7: The model is ready.

Example: Data Points A(1,2), B(1,4), C(1,0)
D(10,2), E(10,4), F(10,0)
we want to divide these points into K=2 clusters.

1. Choose K=2 (we want 2 clusters)
2. Randomly select 2 initial centroids, for example:
centroid1: (1,2)
centroid 2: (10,2)

! Assign points to closest centroid.

For centroid 1. (1,2)	For centroid 2 (10,2)
Distance to A (1,2) = 0	Distance to A (1,2) = 9
Distance to B (1,4) = 2	Distance to B (1,4) = 9
Distance to C (1,0) = 2	Distance to C (1,0) = 9
Distance to D (10,2) = 9	Distance to D (10,2) = 0
Distance to E (10,4) = 9	Distance to E (10,4) = 2
Distance to F (10,0) = 9	Distance to F (10,0) = 2

Assign each point to the closest centroid
point A, B and C are closer to centroid 1
and points, D, E and F are closer to centroid 2

Now recalculate the new centroid.

New centroid 1 (cluster 1): Mean of points A, B an C

$$= \left(\frac{1+1+1}{3}, \frac{2+4+0}{3}\right) = (1,2)$$

New centroid 2 (cluster 2): Mean of points D, E, F)

$$= \left(\frac{10+10+10}{3}, \frac{2+4+0}{3}\right) = (10,2)$$

Hence Final clusters:

cluster 1 : A (1,2), B (1.4), C (1,0)

cluster 2 : D (10,2), E (10.4), F (10,0)

Q.6.a Explain how a single a layer perceptron learn logical OR operation. Assume weights $w_1 = -0.4$ and $w_2 = 0.3$ and learning rate $\alpha = 0.2$

A single layer perceptron can perform logical OR operations, by appropriately setting weights and biases. The perceptron consists of input nodes (which represent the input values), weights (which are associated with each input), a bias, and an activation function. The truth table of logical OR operator is.

Input		Output
x_1	x_2	T
0	0	0
0	1	1
1	0	1
1	1	1

Given weights are

$w_1 = -0.4$ and

$w_2 = 0.3$

learning rate $\alpha = 0.2$

Activation function (z) = step(z) $\begin{cases} 1 \text{ if } z \geq 0 \\ 0 \text{ if } z < \end{cases}$

for first input (0, 0), actual output (T) = 0

Now, Let bias (b_i) = 0

Compute the weighted sum (z) = $w_1 x_1 + w_2 x_2 + b_i$

$= -0.4 \times 0 + 0.3 \times 0 + 0 = 0$

$\therefore z = 0$

Apply the activation function

Output (y) = step (z) = step (0) = 1

Error (e) = Actual O/p (Target) - Estimated O/P

= 0 - 1 = -1

Now update weight

$\Delta w_1 = \alpha (T-Y) \times x_1 = \alpha e x_1 = 0.21 \ (-1) \times 0 = 0$

$\Delta w_2 = \alpha e x_2 = 0.21 \times (-1) \times 0 = 0$

updated bias $\Delta b_i = \alpha (e) = 0.21 \times (-1) = -0.21$

Now,

updated weights $w_1 = w_1 + \Delta w_1 = -0.4 + 0 = -0.4$

$w_2 = w_2 + \Delta w_2 = 0.3 + 0 = 0.3$

Then

weighted sum $(z) = x_1 w_1 + x_2 w_2 + b_i$

$z = 0 \times -0.4 + 0 \times 0.3 + (-0.21) = -0.21$

O/p (y) = step (z) = step (-0.21) < 0

$= 0$

$\therefore$ error (e) = T - y = 0 - 0 = 0

For input (0, 1), T = 1, $x_1 = 0$, $x_2 = 1$ $b = -0.21$

$z = x_1 w_1 + x_2 w_2 + b_i = 0 \times (-0.4) + 1 \times (0.3) - 0.21$

$\therefore z = 0.09$

$y = \text{step}(z) = \text{step}(0.09) > 0$

$= 1$

$\therefore$ Error (e) = T - y = 1 - 1 = 0

For input (1, 0), T = 1, $x_1 = 1$, $x_2 = 0$, $b_i = -0.21$

$z = x_1 w_1 + x_2 w_2 + b_i = 1 \times (-0.4) + 0 \times 0.3 - 0.21$

$= -0.61$

O/p $y = \text{step}(z) = \text{step}(-0.61) = 0$ ✓

Error (e) = T - y = 1 - 0 = 1

Now,

Update weights

$\Delta w_1 = \alpha e x_1 = 0.21 \times 1 \times 1 = 0.21$

$\Delta w_2 = \alpha e x_2 = 0.21 \times 1 \times 0 = 0$

∴ updated weights are,

$w_1 = w_1 + \Delta w_1 = -0.4 + 0.21 = -0.19$

$w_2 = w_2 + \Delta w_2 = 0.3 + 0 = 0.3$

updated bias $(\Delta b_i) = \alpha e = 0.21 \times 1 = 0.21$

Now weighted sum $(z) = x_1 w_1 + x_2 w_2 + b_i$

$= 1 \times (-0.19) + 0 \times 0.3 + 0.21$

$= 0.02$

Output $(Y) =$ step $(z) =$ step $(0.02) = 1$

Error $(e) = T - Y = 1 - 1 = 0$

For another i/p $(1, 1)$, $x_1 = 1$, $x_2 = 1$, $T = 1$, $b_i = 0.21$

weighted sum $(z) = x_1 w_1 + x_2 w_2 + b_i$

$= 1 \times (-0.4) + 1 \times (0.3) + 0.21$

$= 0.11$

Output $(Y) =$ step $(z) =$ step $(0.11) = 1$

Error $(e) = T - Y = 1 - 1 = 0$

Hence the single layer perceptron learn logical OR operator

Q.6.b What is Expert System? Explain the architecture of expert system in detail

An expert system is a computer program that mimics the decision-making abilities of a human expert. It is designed to solve complex problems

by reasoning through bodies of knowledge, repre-sented mainly as if-then rules rather than conventional procedural code.

It solves the most complex issue as an expert by extracting the knowledge stored in its database (knowledge base). The system helps in decision making for complex problems using both facts and heuristics like a human expert. It is called so because it contains the expert knowledge of a specific domain and can solve an complex problem of that particular domain. These systems are designed for a specific domain such as medicine.

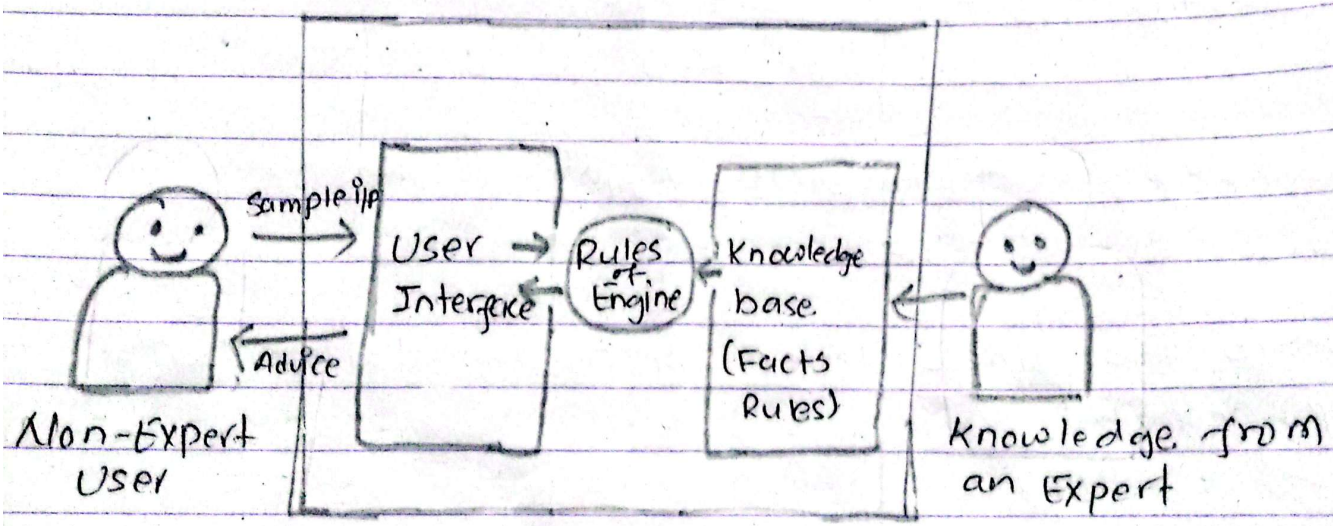


fig: Architecture of Expert system

Components of Expert system

An expert system mainly consists of three components

- User Interface
- Inference Engine (Rules of Engine)
- Knowledge Base

1. User Interface

With the help of user interface, the expert system interacts with the user, takes queries as an input in a readable format, and passes it to the infe-rence engine. After getting the response from the inference engine, it displays the output to the user.

2. Inference Engine (Rules of Engine)

The inference engine is known as brain of expert sy-stem as it is the main processing unit of the expert system. It applies inference rules to the knowledge base to derive a conclusion or deduce new informati-on. It helps in deriving an error-free solution of queries asked by the user. With the help of an inference engine, the system extract the knowledge from the knowledge base.

Inference Engine uses two modes to derive the solutions.

1. Forward chaining: It starts from the known facts and rules, and applies the inference rules to add their conclusion to the known facts.

2. Backward chaining : It is a backward reasoning meth that starts from the goal and work backward to prov the known facts.

3. Knowledge Base
The knowledge base is a type of storage that stores knowledge acquire from the different experts of the particular domain. It is considered as big storage of knowledge. The more the knowledge base, the more precise will be the expert sys -tem. It is similar to database that contains information and rules of a particular domain or subject.

Q. No.) Write short notes on : (Any two)
(a) Bayesian Network
A Bayesian network is a type of graphical model that uses probability to determine the occurrence of an event. It is also known as a belief network or a causal network.

It consists of directed cyclic graph (DCGs) and a table of conditional probabilities to find out the probability of an event happening. It contains nodes and edges, where edges connect the nodes. The graph is acylic - meaning there is no direct path where one node can reach another. The ta -ble of probability, on the other hand, shows the likehood that a random variable will take on certain value.

Bayesian Networks are used across various fields for their ability to more complex relationships & make predictions. For example medicine, Finance, ML weather Forecasting.

Bayesian Networks operate through a process known as Bayesian inference, a statistical reasoning method. Bayesian inference uses Baye's theorem to update the probability of a hypothesis as more evidence is gathered.

(b) Learning by Analogy

Learning by analogy in machine learning is a process that involves using the knowledge or experience gained from solving one problem to solve a different but related problem. It relies on finding similarities between two or more problems and leveraging the solution or strategies from the known problem to address unknown problem.

Analogical reasoning and learning in machine learning can be understood as below

Knowledge Representation:
similarity Assessment
Analogical Mapping
Inference and Generalization

Process of learning by analogy

step 1: Identify the source problem and the target problem

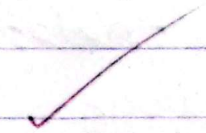
step 2: Analyze the source problem

step 3: Identify the similarities and difference

step 4: Map the knowledge from the source problem to the target problem

step 5: Adapt the solution to the target problem

step 6: Evaluate and refine the solution



Assignment

Artificial Intelligence.

Name: Akriti Poudel
Roll No: 17