Unit 4

Artificial Intelligence Planning

Introduction:-

Planning is required to convert objectives into actions.

 It is decision making task performed by the Robot or computer system to achieve a specific goal.

 Planning is the task of coming up with a sequence of actions that will achieve the goal.

Example :**Goal :** Get distinction or Just clear all the subjects.

Planning:-How many days left for exam, Number of chapters to study, Notes availability etc

 Classical planning environments

◦ Fully observable, deterministic, finite, static (change only happens when the agent acts), and discrete (in time, action, objects)

PLANNING PROBLEM :-

 The Planning Problem is the question that how to go to the next state or the goal state from the current state.

 The planning problem is defined with:

 1.Domain Model: It define the actions along with the objects.

 2.Initial State: It is state where any action is yet to take place.

 3.Goal State:-state which the plan is intended to achieve.

Components of Planning system:-

1.Choose the best rule based upon heuristics.(selection of simple subject it is based on past experience or input from the seniors or lecture attended).

2. Apply this rule to create a new state.(studying of the subject , so as to achieve next stage).

3. Detect when a solution is found.(A Position is reached where he can score more than cut-off marks).

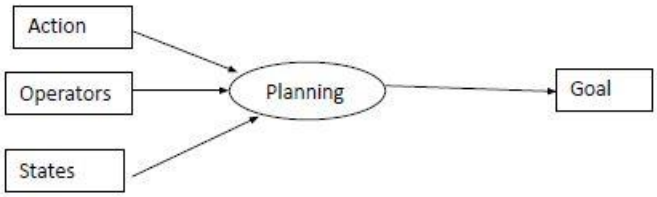
 4. Detect dead ends so that they can be avoided. (Find the difficult topic and avoid them)

 5.Detect when a nearly solved state occurs and use special methods to make it a solved state.-(In position to get more than cut-off marks in all subjects except mathematics and then apply strategies to reach the goal.

Example: A student who want to get an all clear (i.e To score marks above cut-off in all subjects)

Planning Representation:-

Planning Representation Consist of Three components State, Action, and Goal.



1.State:- State is conjunction of positive literals which cannot contain variables and invoke functions. This is representation of facts.

2. **Action:-** These are preconditions that must hold before execution and the effects after execution.

3. **Goal:** It is most specified state .a state is satisfy the given goal if it consists of all the objects required for the goal .

Planning Languages:-

 Planning language should be expressive and every planning language makes use of the representation schema so that the algorithms can be operated on it.

Types of languages

1.Standard Research Institute Problem solver (STRIPS):

 It was developed in 1970's at Stanford for the first intelligent robot.

 It can describe the world(Initial state, Goal state) by providing Objects, Action ,Precondition and effects**.**

 A STRIPS instance is composed of:

1. State:- It is conjunction of positive literals which cannot contain variables and invoke functions.

Example:- At(Home)^Have(Banana)

2. **Goal:-** These are conjunction of literals may contain variables .

Example:- At(Home)^¬Have(Banana)

AT(X)^Sells(X, Banana)

3. **Action**:- These are preconditions that must hold before execution and the effects after execution.

Example:

Action (Fly(p, from,to)

PRECOND: At(p, from) ∧ Plane(p) ∧ Airport(from)

∧ Airport(to)

EFFECT:¬At(p, from) ∧ At(p,to))

2. **Action Description Language (ADL)**

 To overcome the limitation of STRIPS , ADL is used

 It is more expressive language than the STRIPS.

Properties of ADL

1. It allows negative literals.

2.It use quantified variables with the disjunction and the conjunction.

Example:- *¬ Poor ∧ (Famous ∨ Smart)*

3.Conditional Post conditions are allowed.

4.Variables with different types at the same time are allowed and also equality property is available.

Example :- we consider car driving case with strips:

Action(drive(c,from,to))

Pre_condition:at(c,from)^car(c)

Post_condition:at(c,from)^at(c ,to)

In strips post-condition means remove the 1st ‘at’

condition and add the 2nd ‘at’ condition.

With ADL same action is represented as follows.

Action(drive(c,from,to))

Pre_condition:at(c,from)^car(c) ^(from≠to)

Post_condition:at(c,from)^at(c ,to)

3.Planning Domain Description Language(PDDL):

It is standard encoding language for “classical” planning task.

It is superset of STRIPS and ADL.

:Components of a PDDL planning task:

1.**Objects:** Things in the world that interest us.

2.**Predicates**: Properties of objects that we are interested in; can be true or false.

3. **Initial state:** The state of the world that we start in.

4.**Goal specification:** Things that we want to be true.\

5.**Actions/Operators:** Ways of changing the state of the world

Planning tasks specified in PDDL are separated into two files:

1. A **domain file** for predicates and actions.

2. A **problem file** for objects, initial state and goal specification.

Domain files look like this:

(define (domain <domain _name>)

<PDDL code for predicate>

<PDDL code for first action >

<PDDL code for last action>

)

Problem files look like this:

(define (Problem <Problem _name>)

(: domain <domain _name>)

<PDDL code for Objects>

<PDDL code for Initial State>

<PDDL code for Goal >

Blocks-World:-

 The blocks-world problem is known as **Sussman Anomaly.**

 In blocks-world problem, three blocks labeled as 'A', 'B', 'C' are allowed to rest on the flat surface. The given condition is that only one block can be moved at a time to achieve the goal

 It is consisting of followings

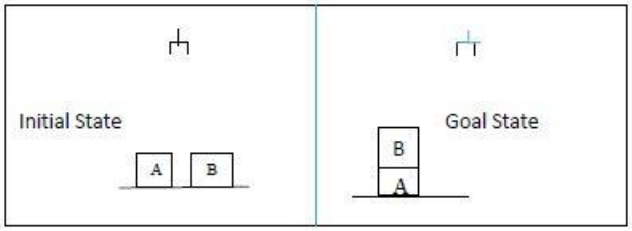
1.A Table

2.Identical blocks with unique letter on them

3.Block are put one to another in stack form.

4.Stack is bulit with a robot arm. The arm can perform operations of lifting a single block at a time and placing it.

Example :



Predicate used to describe states in block world are

1.On(A,table): Block A is on the table.

2. On(B,table): Block B is on the table.

3. On(B,A): Block B is on Block A.

4.Clear (A):Block A has nothing on it.

5. Clear (B):Block B has nothing on it.

6.Holding(B):Robot arm is holding B.

7.Empty-Arm:the arm is not holding anything

State Representation

Initial State: On(A,table)^ On(B,table) ^Clear (A)^ Clear (B) ^Empty-Arm

Goal State: On(A, table)^ On(B,A) ^ Clear (B) ^Empty-Arm

Four Action(Operation) used to describe states in block world are:-

1.UNSTACK(B,A)- To lift block B from A.

2.Stack(B,A):To place block B on A.

3.Lift(B): To lift the Block B from the table.

4.Place(B): To put the block B on the table.

The blocks world is chosen because:-

 it is sufficiently simple and well behaved.

 easily understood

 It provides a good sample environment to study planning:

◦ problems can be broken into nearly distinct subproblems

Goal Stack Planning:-

 Goal Stack Planning is one of the earliest methods in artificial intelligence in which we work backwards from the goal state to the initial state..

 This approach uses a Stack for plan generation. The stack can contain Sub-goal and actions described using predicates.

 The Sub-goals can be solved one by one in any order.

 We start at the goal state and we try fulfilling the preconditions required to achieve the initial state.

 We keep solving these “goals” and “sub-goals” until we finally arrive at the Initial State

The Robot Arm can perform 4 operations:-

 STACK(X,Y) : Stacking Block X on Block Y

 UNSTACK(X,Y) : Picking up Block X which is on top of Block Y

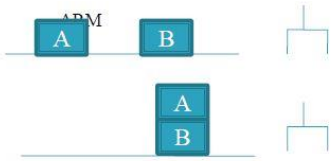
 PICKUP(X) : Picking up Block X which is on top of the table

 PUTDOWN(X) : Put Block X on the table

Operation	Precondition	Action
1.PICKUP(X)	ARM_EMPTY^ ON(X ,TABLE) ^ CLEAR(X)	Holding (X)
2.PUTDOWN(X)	Holding (X)	ARM_EMPTY^ ON(X ,TABLE) ^ CLEAR(X)
3.STACK(X,Y)	Holding (X)^CLEAR(Y)	On(X,Y) ^ CLEAR(X) ^ ARM_EMPTY
4.UNSTACK(X,Y)	On(X,Y) ^CLEAR(X) ^ARM_EMPTY	Holding (X) ^CLEAR(Y)

Initial state:-

Goal state:-



solution:-

:-

Step1: Our goal is ON(A,B)

Explanation: This goal is achieved by operation No.3 so

Step2: **Stack(A,B)**

Step3: Holding (A)^CLEAR(B)- (Precondition)

Explanation: In the step3 Holding(A) is **not true** when we see initial state It is expand by operation No.1

Step4: **PICKUP(A)**

Step5: ARM_EMPTY^ON(A ,TABLE) ^CLEAR(A)- (Precondition) - TRUE

Note :-- hence here step 5 is True so it is also prove the step 3 .

So, when step 3 and 5 true then Step2 **Stack(A,B)** TRUE

And we can achieve our Goal ON(A,B)

The preconditions are highlighted with Red and the actions are marked with **Blue**

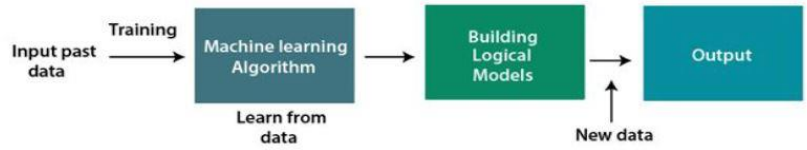
What is machine learning?

Machine learning is a branch of AI which enables machines to learn from past data or experiences without being explicitly programmed.

Machine learning enables a computer system to make predictions or take some decisions using historical data without being explicitly programmed.

Machine Learning Process

 A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it..



Goal of machine Learning:-

 The primary objective of machine learning research is to develop general-purpose algorithms in practical value.

 The main purpose of machine learning is to study and design the algorithms that can be used to produce the predicates from the given dataset.

Machine learning can also be used to:

1. Compete Intelligently
2. Enhance Customer Service

3. Manage our Sales Funnel
4. Detect Fraudulent Activity
5. Predict Journey times
6. Predict how long jobs may take
7. Score our prospects and customers
8. Behavior of our market

Goal of machine Learning:-

1.Image Recognition

2. Speech Recognition

3. Traffic prediction

4. Product recommendations

5. Self-driving cars

6. Email Spam and Malware Filtering

7. Virtual Personal Assistant

8. Online Fraud Detection

9. Stock Market trading

10. Medical Diagnosis

Applications of Machine learning:-

1.Data Collection

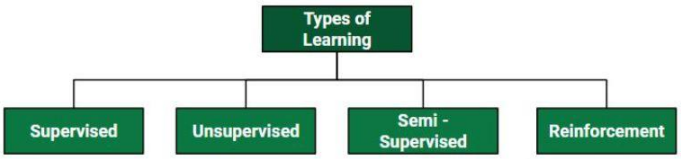
2. Less Amount of Training Data

3. Non-representative Training Data

4. Poor Quality of Data

5. Irrelevant/Unwanted Features

Challenges for Machine Learning:-



1) **Supervised Learning**

Supervised learning is a type of machine learning method in which we provide sample labeled data to the machine learning system in order to train it, and on that basis, it predicts the output.

2) **Unsupervised Learning**

Unsupervised learning is a learning method in which a machine learns without any supervision.

3. **Semi-supervised Learning:**

its working lies between Supervised and Unsupervised techniques. We use these techniques when we are dealing with a data which is a little bit labeled and rest large portion of it is unlabeled. This technique is mostly applicable in case of image data-sets where usually all images are not labeled.

4) **Reinforcement Learning**

It is a feedback-based learning method. in which a learning agent gets a reward for each right action and gets a penalty for each wrong action

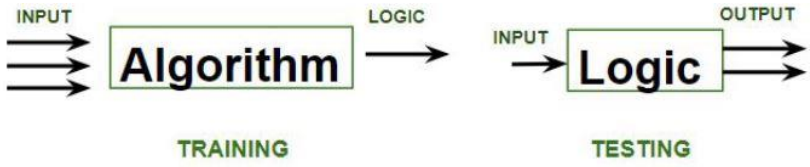
 In Supervised Learning, a machine is trained using '**labeled' data**. Datasets are said to be labeled when they contain both input and output parameters. and on that basis, it predicts the output. This implies that some data is already tagged with the correct answer.

 The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher.

1.Supervised Learning:-

 In Supervised Learning, a machine is trained using ‘labeled’ data. Datasets are said to be labeled when they contain both input and output parameters. and on that basis, it predicts the output. This implies that some data is already tagged with the correct answer.

 The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher.



 Example:

a basket filled with different kinds of fruits. Now the first step is to train the machine with all different fruits one by one like this:

If the shape of the object is rounded and has a depression at the top, is red in color, then it will be labeled as ‐**Apple**.

If the shape of the object is a long curving cylinder having Green-Yellow color, then it will be labeled as ‐**Banana**.

Thus, the machine learns the things from training data(basket containing fruits) and then applies the knowledge to test data(new fruit). Supervised learning classified into two categories of algorithms:

1.Classification: A classification problem is when the output variable is a category, such as “Red” or “blue” or “disease” and “no disease”.

2.Regression: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.

 Supervised learning classified into following categories of algorithms:

Types:-

1. Regression

2. Logistic Regression

3. Classification

4. Naive Bayes Classifiers

5. K-NN (k nearest neighbors)

6. Decision Trees

7. Support Vector Machine

 Unsupervised learning is a learning method in which a machine learns without any supervision.

 The training is provided to the machine with the set of data that has not been labeled, classified.

 No teacher is provided that means no training will be given to the machine. Therefore, the machine is restricted to find the hidden structure in unlabeled data by itself.

 Unsupervised learning is classified into two categories of algorithms:

1.Clustering: A clustering problem is where we want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.

2.Association: An association rule learning problem is where we want to discover rules that describe large portions of our data, such as people that buy X also tend to buy Y.

2.Unsupervised learning

Example: suppose it is given an image having both dogs and cats which it has never seen. So, the machine has no idea about the features of dogs and cats so we can't categorize it as 'dogs and cats. But it can categorize them according to their similarities, patterns, and differences, i.e., we can easily categorize the above picture into two parts. The first may contain all pics having **dogs** in it and the second part may contain all

pics having **cats** in it. Here machine didn't learn anything before, which means no training data .

Types of Unsupervised Learning

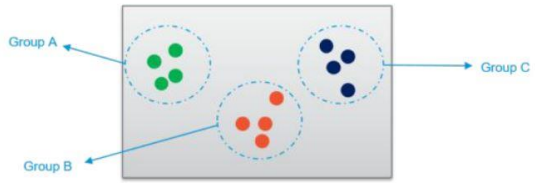
1. Clustering

2. Association

1.Clustering :-

 In this Learning we find patterns in the data that we are working on.

 It may be the shape, size, color etc. which can be used to group data items or create clusters.



Some popular algorithms in Clustering are:

1.Hierarchical Clustering - This algorithm builds clusters based on the similarity between different data points in the dataset.

2.K-Means Clustering -The algorithm creates clusters of different data points which are as homogenous as possible by calculating the centroid of the cluster and making sure that the distance between this centroid and the new data point is as less as possible.

3.K-NN Clustering– This algorithm is also called as a lazy learner because it learns only when the algorithm is given a new data point. It works well with smaller datasets as huge datasets take time to learn.

 2. **Association** :- In this type of Learning we find the dependencies of one data item to another data item and map them such that they help you profit better.

1. Apriori algorithm–

• It is based on **breadth-first search**.

• This Algorithm maps the dependency of one data item with another which can help us understand what data item influences the possibility of something happening to the other data item.

• For example, bread influences the buyer to buy milk and eggs. So that mapping helps increase profits for the store.

2.FP-Growth Algorithm –

•The Frequency Pattern (FP) algorithm finds the count of the pattern that has been repeated, adds that to a table and then finds the most possible item and sets that as the root of the tree.

•This algorithm is faster than Apriori as the support is calculated and checked for increasing iterations rather than creating a rule and checking the support from the dataset.

Supervised vs. Unsupervised Learning

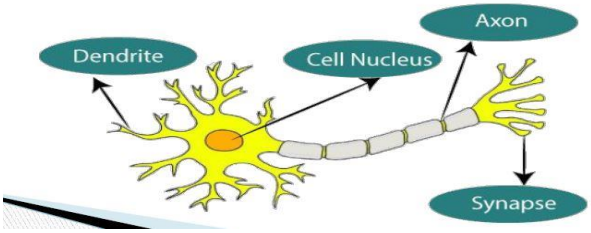
Parameter	Supervised Learning	Unsupervised Learning
Dataset	Labelled	Unlabelled
Method of Learning	Guided learning	The algorithm learns by itself using dataset
Complexity	Simpler method	Computationally complex
Accuracy	More Accurate	Less Accurate

Artificial Neural Network (ANN)?

ANN is information processing paradigm that is inspired by biological nervous system.

Artificial neural networks are built like the human brain, with neuron nodes interconnected like a web.

It is a computational network based on biological neural networks that construct the structure of the human brain.



Artificial Neural Network" is derived from Biological neural networks that develop the structure of a human brain.

The human brain has hundreds of billions of cells called neurons.

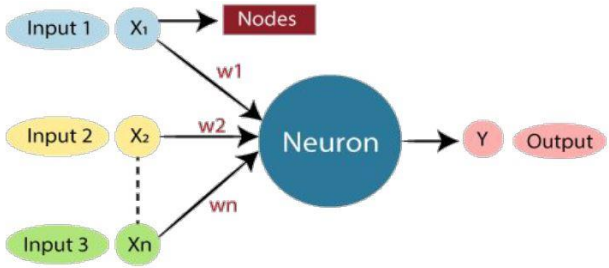
Each neuron is made up of a cell body that is responsible for processing information by carrying information towards (inputs) and away (outputs) from the brain.

Dendrites − They are tree-like branches, responsible for receiving the information from other neurons it is connected to. In other sense, we can say that they are like the ears of neuron.

Soma − It is the cell body of the neuron and is responsible for processing of information, they have received from dendrites.

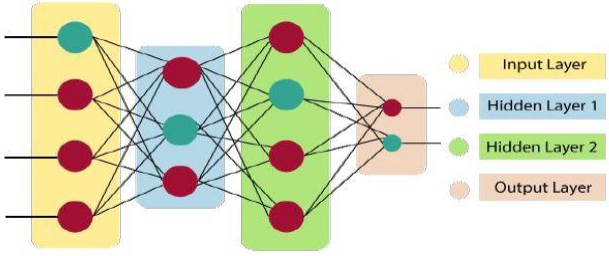
Axon − It is just like a cable through which neurons send the information.

Synapses − It is the connection between the axon and other neuron dendrites. Dendrites from Biological Neural Network represent **inputs** in Artificial Neural Networks, cell nucleus represents **Nodes**, synapse represents **Weights,** and Axon represents **Output.**



Architecture of an artificial neural network:

 Artificial Neural Network primarily consists of three layers:



Input Layer:

 it accepts inputs in several different formats provided by the programmer.

Hidden Layer:

 The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.

Output Layer:

 The input goes through a series of transformations using the hidden layer, which finally results in output that is conveyed using this layer.

 The artificial neural network takes input and computes the weighted sum of the inputs and includes a bias. This computation is represented in the form of a transfer function.

$$\sum_{i=1}^{n} Wi * Xi + b$$

It determines weighted total is passed as an input to an activation function to produce the output. Activation functions choose whether a node should fire or not. Only those who are fired make it to the output layer..

 Neural Network or **artificial neural network** (ANN) are modeled the same as the human brain. The human brain has a mind to think and analyze any task in a particular situation.

 But how can a machine think like that? For the purpose, an artificial brain was designed is known as a neural network.

 The neural network is made up many **perceptions**.

 Backpropagation is a supervised learning algorithm, for training Multi-layer Perceptrons (Artificial Neural Networks).

 The Backpropagation algorithm looks for the minimum value of the error function in weight space using a technique called the delta rule or gradient descent.

 The weights that minimize the error function is then considered to be a solution to the learning problem.

Backpropagation:

 Support vector machines (SVMs) are powerful supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems.

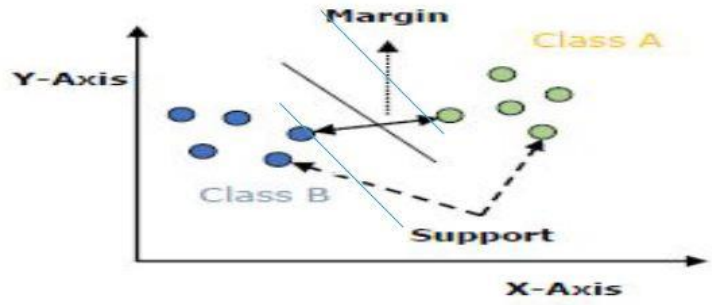
 Support Vectors are simply the co-ordinates of individual observation.

Working of SVM

 An SVM model is basically a representation of different classes in a hyper-plane in multidimensional space.

 The hyper-plane will be generated in an iterative manner by SVM so that the error can be minimized.

 The goal of SVM is to divide the datasets into classes to find a maximum marginal hyper-plane (MMH).



Support Vector Machine

 The followings are important concepts in SVM −

 Support Vectors − Data points that are closest to the hyper-plane is called support vectors. Separating line will be defined with the help of these data points.

 Hyper-plane − It is a decision plane or space which is divided between a set of objects having different classes.

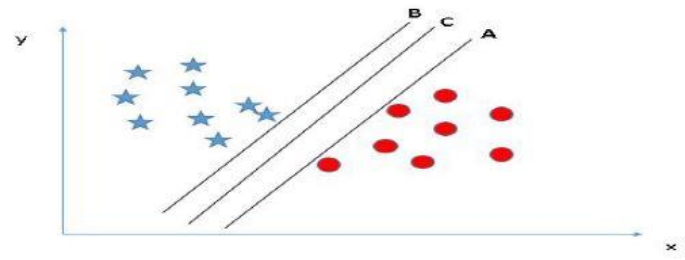
 Margin − It is the gap between two lines on the closet data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors.

 Large margin is considered as a good margin and small margin is considered as a bad margin.

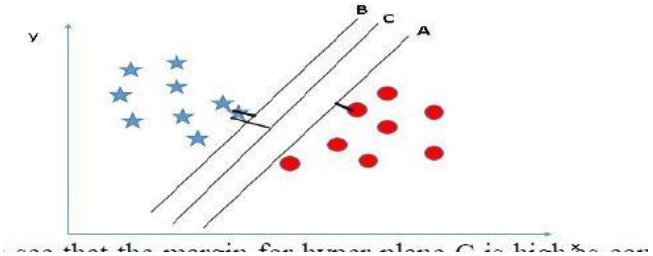
 If we select a hyper-plane having low margin then there is high chance of miss-classification.

 Identify the right hyper-plane

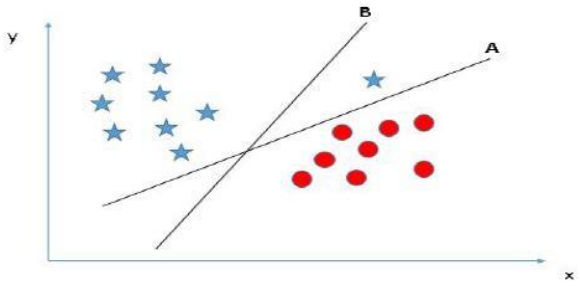
Here, we have three hyper-planes (A, B and C) and all are separating the classes well.Now, How can we identify the right hyper-plane?



In below diagram Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as **Margin**.



Above, we can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C



 In the above example hyper-plane **B** as it has higher margin compared to **A.** Here, hyper-plane B has a classification error and A has classified all correctly.

Therefore, the right hyper-plane is **A.**

 The main goal of SVM is to divide the datasets into classes to find a maximum marginal hyper-plane (MMH) and it can be done in the following two steps −

 First, SVM will generate hyper-planes iteratively that isolate the classes in best way.

 Then, it will choose the hyper-plane that separates the classes correctly.

 Pros of SVM classifiers

 SVM classifiers offers great accuracy and work well with high dimensional space. SVM classifiers basically use a subset of training points hence in result uses very less memory.

 Cons of SVM classifiers

 They have high training time hence in practice not suitable for large datasets. Another disadvantage is that SVM classifiers do not work well with overlapping classes.

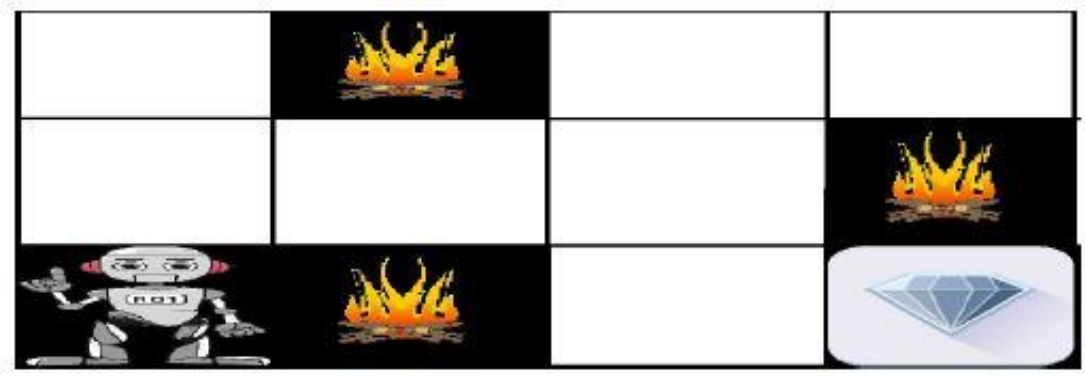
 It is a feedback-based learning method. in which a learning agent gets a reward for each right action and gets a penalty for each wrong action is about taking suitable action to maximize reward in a particular situation.

 It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation.

 Reinforcement learning differs from the supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of training dataset, it is bound to learn from its experience.

Reinforcement learning

 We have an agent and a reward, with many hurdles in between. The agent is supposed to find the best possible path to reach the reward. The following problem explains the problem more easily.



The above image shows robot, diamond and fire. The goal of the robot is to get the reward that is the diamond and avoid the trouble that is fire. The robot learns by trying all the possible paths and then choosing the path which gives him the reward with the least hurdles. Each right step will give the robot a reward and each wrong step will subtract the reward of the robot. The total reward will be calculated when it reaches the final reward that is the diamond.

 Main points in Reinforcement learning –

Input: The input should be an initial state from which the model will start

Output: There are many possible output as there are variety of solution to a particular problem

Training: The training is based upon the input, The model will return a state and the user will decide to reward or punish the model based on its output. The model keeps continues to learn.

 The best solution is decided based on the maximum reward.

Applications of Reinforcement Learning –

 RL can be used in robotics for industrial automation.

 RL can be used in machine learning and data processing

 RL can be used to create training systems that provide custom instruction and materials according to the requirement of students.

10.8 MULTI-AGENT-BASED LEARNING

Let us move our discussion to multi-agent learning. A single agent cannot handle learning in case of complex applications. A team or group of agents possesses the potential to overcome the limitation of single agent and work in co-ordination to accomplish a task. There can be two cases in multi-agent-based learning—one where the agent tries to maximise its own utility and other, where they work in collaboration to achieve some common goal. Consider a manufacturing industry domain. The tasks are split and assigned where each agent works in co-operation to build the end product. This is the case to achieve a common goal. Let us consider one more example. A simple example of multi-agent-based learning is game playing. Assume that in a particular gaming environment, multiple agents are in operation to select the best strategy. Now this can be related with the reinforcement learning, where for each strategy of the agent, some reward is achieved. This is where each agent tries to maximise his own utility function.

10.10 ADAPTIVE LEARNING

No learning method is complete in itself and there is need to select the learning method based on the requirement. Further, there is need to develop a combination of some of the existing methods based on need. *Adaptive machine learning algorithms* are the machine learning models, where the changes in the environment help in selecting the algorithm or learning method. As per the scenario, most suitable algorithm is selected. Moreover, the development of especially fast-adapting algorithms poses many different issues like selection of choices, handling equilibrium states and so on. The adaptive learning solves some of the complex problems for which a single learning method is not enough. This method is even more appropriate when the environment is continuously changing and real-time response is expected.

10.11 LEARNING FOR DECISION-MAKING

What is observed in different learning mechanisms is that with the learnt concepts, the capability to take decisions in increased. Speaking about the supervised or unsupervised methodologies, the decisions taken are not sequential in nature. That is, if the system makes a mistake on one decision, this has no bearing on the subsequent decisions. To cope up with this dynamic situation, there is a need to understand the perspective of decision-making. Another aspect is environment and system learning, which also needs to be looked upon during decision-making. Hence while taking decisions, one specific learning approach may not be suitable. The learning approach is dependent on decision scenario.

10.12 SPEEDUP LEARNING

Speedup learning typically deals with speeding up problem solving by effective use of problem solving experience. Hence, prior problem solving experience is an input for speedup learning.

In this learning,

1. There is no interaction with the environment.
2. New problems cannot be solved.

So, speedup learning accelerates the process based on the previous experiences and prior observations.

The process of speedup learning is depicted in Figure 10.18.

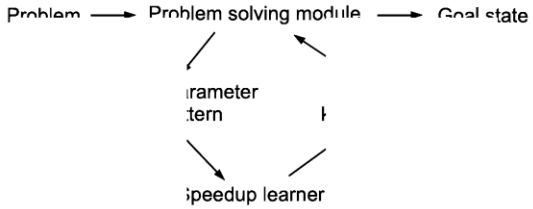


Figure 10.18 Speedup learning modules.

Another dimension to the speedup learning is generalised caching. This is also known as *explanation-based learning*. There are a number of issues with explanation-based learning, as it is implemented and embedded in system to solve real-life problem and is suitable for a particular set of problems, where the sequential processes once developed can be used again and again. But this is not the case in many real-life problems, where dynamic change in environment demands the improvement in the established scenarios and even there is a need to keep on learning based on the new findings.