# Chapter- 6 Learning System

## Learning:

- Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task (or task drawn from the same population) more effectively the next time.
- Learning denotes changes in a system that enables the system to do the same task more efficiently next time.
- Learning is an important feature of "Intelligence".
- Learning is constructing or modifying representations of what is being experienced.

**Definition:**

A computer program is said to be learn from experience E with respect to some class of tasks T and performance measure P, if it's performance at tasks in T, as measured by P, improves with experience E.

**Given:** A task T

A performance measure P.

Some experience E with the task.

**Goal:** Generalized the experience in a way that allows improving your performance on the task.

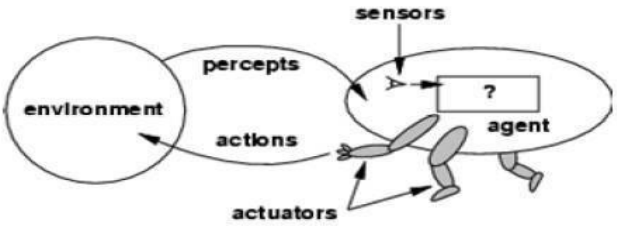
Why do you require Machine Learning?

- Understand and improve efficiency of human learning.
- Discover new things or structure that is unknown.
- Fill in skeletal or incomplete specification about a domain.

## 1.2 Learning Agents

An agent is an entity is capable of perceiving and does actions.

An agent can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.

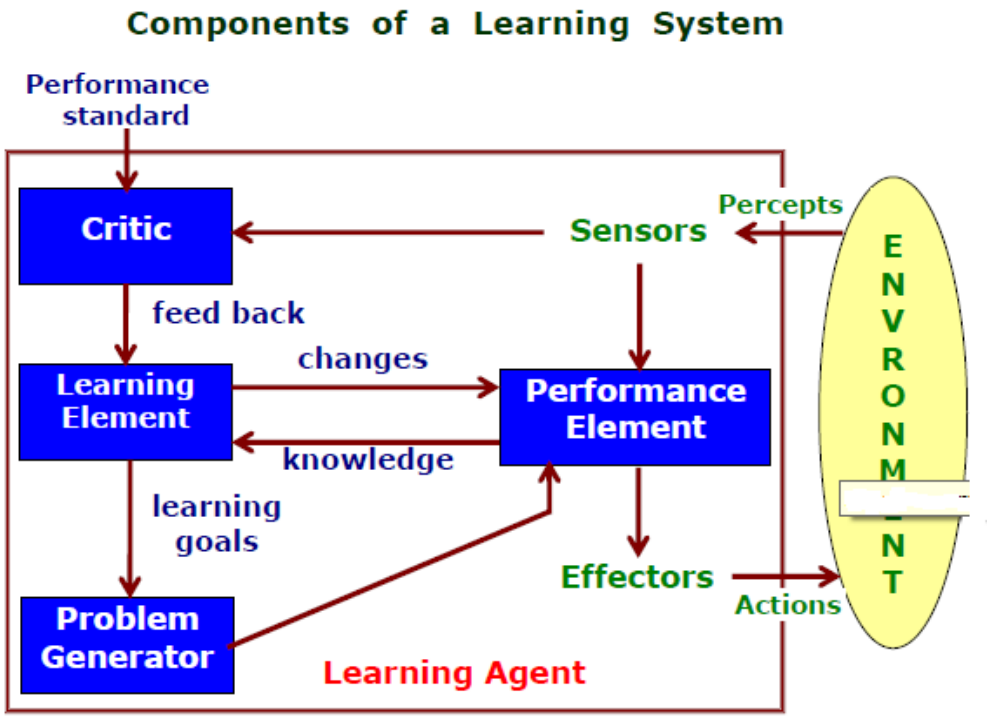



| Environment / Agent | Sensors | Actuators |
|---|---|---|
| Human agent | Eyes, ears, etc | Leg, hands, mouth |
| Robotic agent | Cameras, IR range finders | motors |
| Software agent | Key stroke, File contents | Displays to screen, write fi |



In computer science an agent is a software agent that assists users and acts in performing computer related tasks.

**Components of a Learning system:**




1. **Performance Element:** The performance element is the agent itself that acts in the world . It takes in percepts and decides on external actions.
2. **Learning Elements:** It is responsible for making improvements. Takes knowledge about performance element and some feedback determines how to determine performance element.

3. **Critic:** Tells the Learning element how agent is doing (success or failure) by comparing with a fixed standard of performance.
4. **Problem Generator:** Suggests problems or actions that will generate new examples or experiences that will aid in training the system further.

## Types of Learning

### 1. Rote learning

Rote learning technique avoids understanding the inner complexities but focuses on memorizing the material so that it can be recalled by the learner exactly the way it was read or heard.

- **Learning by Memorization** which avoids understanding the inner complexities the subject that is being learned; Rote learning instead focuses on memorizing the material so that it can be recalled by the learner exactly the way it was read or heard.
- **Learning something by Repeating** over and over and over again; saying the same thing and trying to remember how to say it; it does not help us to understand; it helps us to remember, like we learn a poem, or a song, or something like that by rote learning.

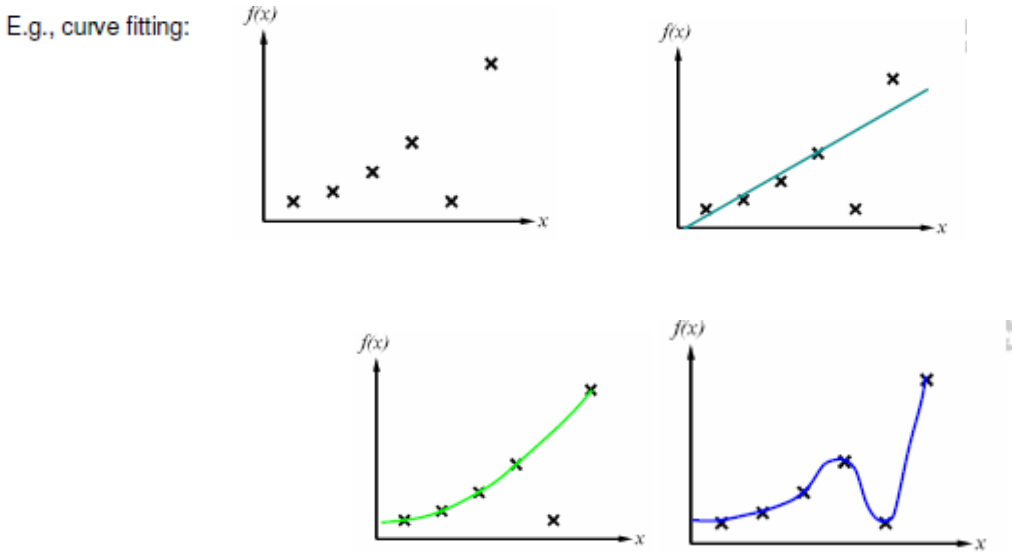
### 2. Inductive Learning

- A process of learning by example.
- The system tries to induce a general rule from a set of observed instances. The learning method extracts rules and patterns out of massive data sets.
- This involves the process of *learning by example* -- where a system tries to induce a general rule from a set of observed instances.
- This involves classification -- assigning, to a particular input, the name of a class to which it belongs. Classification is important to many problem solving tasks.
- A learning system has to be capable of evolving its own class descriptions:
  - Initial class definitions may not be adequate.
  - The world may not be well understood or rapidly changing.

- The task of constructing class definitions is called *induction* or *concept learning*.

- Three methods are used in inductive learning
  1. Winston's Learning program
  2. Version Trees
  3. Decision trees

**Inductive learning Example: Curve fitting**



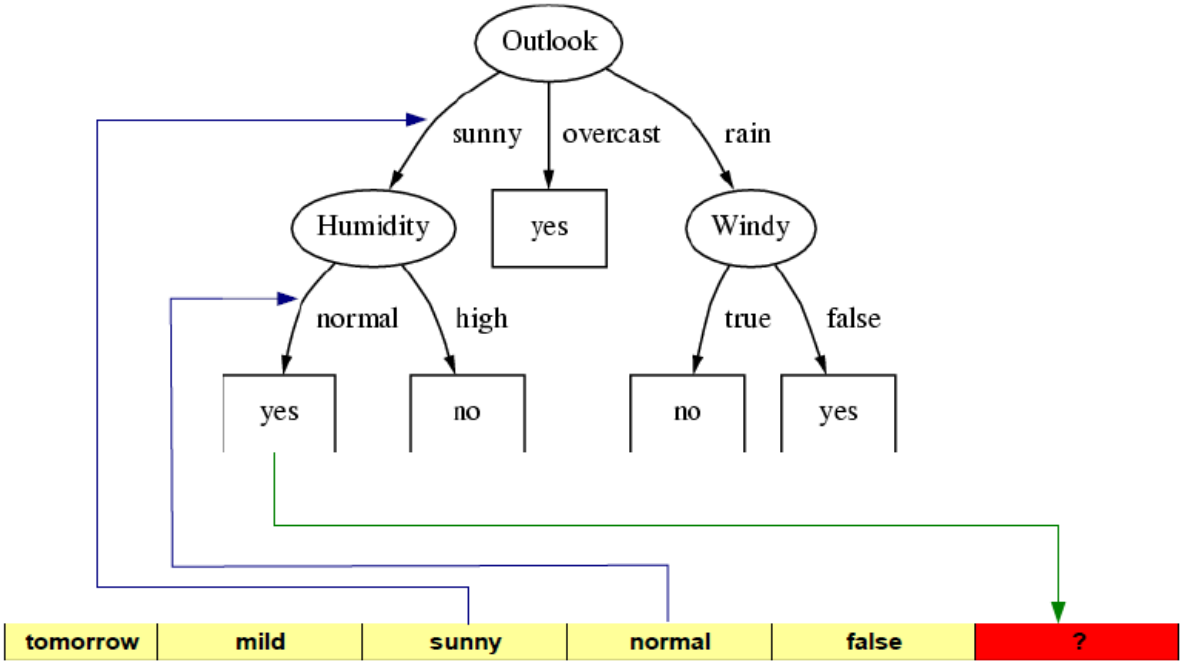

## 3. Decision tree

- Decision tree is a powerful tool of classification and prediction.
- It represents rules that are easily expressed and use to retrieve useful information
- A decision tree consist of
  - **Nodes:** test for the value of certain attribute.
  - **Edges:** correspond to the outcome of a test.
  - **Leaves:** terminal nodes that predict the outcome.

# A Sample Task

| Day | Temperature | Outlook | Humidity | Windy | Play Golf? |
|---|---|---|---|---|---|
| 07-05 | hot | sunny | high | false | no |
| 07-06 | hot | sunny | high | true | no |
| 07-07 | hot | overcast | high | false | yes |
| 07-09 | cool | rain | normal | false | yes |
| 07-10 | cool | overcast | normal | true | yes |
| 07-12 | mild | sunny | high | false | no |
| 07-14 | cool | sunny | normal | false | yes |
| 07-15 | mild | rain | normal | false | yes |
| 07-20 | mild | sunny | normal | true | yes |
| 07-21 | mild | overcast | high | true | yes |
| 07-22 | hot | overcast | normal | false | yes |
| 07-23 | mild | rain | high | true | no |

# Decision Tree Learning

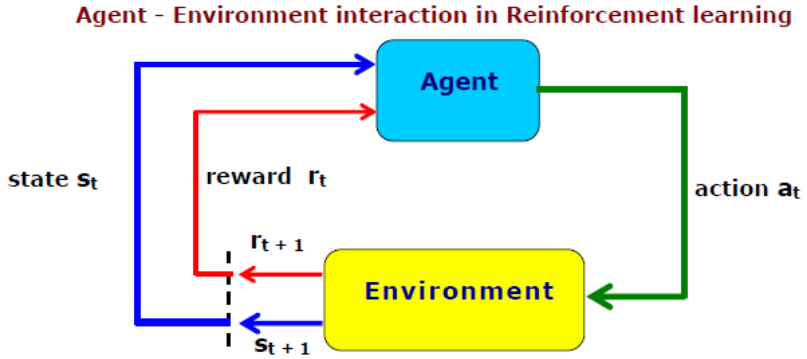


## 4. Reinforcement Learning

- Reinforcement learning refers to a class of problem in machine learning which postulates an agent exploring an environment.

- the agent perceives it's current state and takes actions.
- The environment in return, provides a reward positive or negative.
- The algorithm attempt to find a policy for maximizing cumulative reward for the agent over the course of the problem.

- In other word Reinforcement is: "A computational approach to learning whereby an agent tries to maximize the total amount of reward it receives when interacting with a complex, uncertain environment".
- It is a way of programming agents by reward and punishment without needing to specify how task is to be achieved.




‡ The agent and environment interact in a sequence of discrete time steps, $t = 0, 1, 2, 3, 4, \ldots$

‡ At each discrete time $t$, the agent (learning system) observes state $s_t \in S$ and chooses action $a_t \in A$

‡ Then agent receives an immediate numerical reward $r_{t+1} \in \Re$ and the state changes to $s_{t+1}$

‡ At each time step, the agent implements a mapping from states to probabilities of selecting each possible action.

‡ The agent's goal, is to maximize the total amount of reward it receives over the long run.

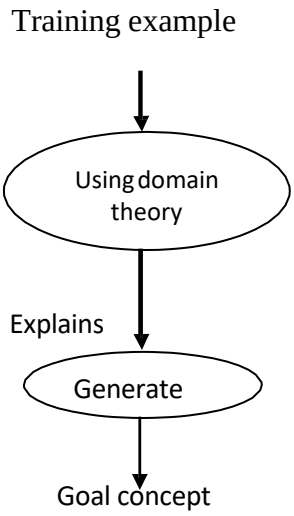
## 5. Explanation based Learning (EBL)

- Human learning is accomplished by examining particular situations and rating them to the background knowledge in the form of general principles. This kind of learning is called "Explanation Based Learning (EBL)".
- The system attempts to learn from single example' X' by explaining why X is an example of target concept.
- EBL system accepts an example (i.e., a training example) and explain what it learns from the example. The EBL system takes only the relevant concepts of the training. This explanation is translated into particular form that a problem solving can understand. The explanation is generalized so that it can be used to solve other problem.
- An EBL accepts 4 kinds of inputs

a. **A training Example:** what the learning sees in the world.
b. **A goal concept:** High level description of what the program is supposed to learn.
c. **Operational criterion:** description of which concepts are usable.
d. **A domain theory:** a set of rules that describes relationships between object and action in a domain.

- EBL has two steps:

a. **Explanation:** the domain theory is used to prune away all unimportant aspects of the training example with respect to the goal concept.
b. **Generalization:** the explanation is generalized as far as possible while still describing the goal concept.




**Example:**

o/p: Lender (x,y)→ relative (x,y) ∧ Rich (y)

relative (x,y) ⟵ uncle (y,x)

rich(y) ⟵ Ceo(y,b) ∧

Bank(B)

rich (y) ⟵ Own (y,H) ∧

House(H) Input