What is artificial intelligence? How to find out whether a computer is showing intelligent behavior or not?

Acting humanly: The Turing Test approach

The **Turing Test,** proposed by Alan Turing (1950), was designed to provide a satisfactory operational definition of intelligence. Rather than proposing a long and perhaps controversial list of qualifications required for intelligence, he suggested a test based on in distinguishability from undeniably intelligent entities-human beings. The computer passes the test if a human interrogator, after posing some written questions, cannot tell whether the written responses come from a person or not.
The computer would need to possess the following capabilities:

- **natural language processing** to enable it to communicate successfully in English.
- **knowledge representation** to store what it knows or liears;
- **automated reasoning** to use the stored infornnation to answer questions and to draw new conclusions;
- **machine learning** to adapt to new circumstances and to detect and extrapolate patterns.

Turing's test deliberately avoided direct physical interaction between the interrogator and the computer, because physical simulation of a person is unnecessary for intelligence. However, the so-called **total Turing Test** includes a video signal so that the interrogator can test the subject's perceptual abilities, as well as the opportunity for the interrogator to pass physical objects "through the hatch." To pass the total Turing Test, the computer will need:

- **computer vision** to perceive objects, and
- **robotics** to manipulate objects and move about.

Discuss about the Humans and computers?

Humans and Computers:

	Human Brain	**Computer**
	Typically, neurons are five to six orders of magnitude slower than silicon logic gates	
	Neural events happen in the millisecond (10^{-3} s) range However, the brain makes up for the relatively slow rate of operation of a neuron by having a truly staggering number of neurons (nerve cells) with massive interconnections between them. It is estimated that there are approximately 10 billion neurons in the human cortex, and 60 trillion synapses or connections The net result is that the brain is an enormously efficient structure.	Events in a silicon chip happen in the nanosecond (10^{-9} s) range

	The energetic efficiency of the brain is approximately 10^{-16} joules (J) per operation per second.	The corresponding value for the best computers in use today is about 10^{-6} joules per operation per second.
Computational Units	10^{11} Neurons	1 CPU, 10^{8} gates
Storage Units	10^{11} Neurons 10^{14} Synapses	10^{10} bits RAM 10^{11} bits disk
Cycle Time	10^{-3} sec	10^{-9} sec
Bandwidth	10^{14} bits/sec	10^{10} bits/sec
Memory updates/sec	10^{14}	10^{9}

Describe about the knowledge representation?

Knowledge Representation:

4.1 Introduction

Human beings possess the power of reasoning and using their knowledge of ever to choose actions. Both knowledge and reasoning play an important role in deali with partially observable environments. Unlike reflex agents, knowledge based agen can combine general knowledge with current percepts to infer hidden aspects of t current state prior to selecting actions. In human beings using hidden knowledge an inherent part of decision making. Whether it is in natural language processin or deciding a treatment for an ailment, we all use knowledge in decision makin Consider the knowledge hidden in this statement.

> She threw a stone at the mango and ran to catch it.
>
> What is "it" in the above sentence? Stone or mango? Everyone knows it is mango.

There is ambiguity about "it" in the above statement, but because of prior knowledge; we are able to resolve it. However, representation of this sort of knowledge is no easy, since it imbibes an inherent logic in the way we think, which belies mathematical definition.

Intelligence requires knowledge; knowledge once acquired, must be organized for use. Knowledge base agents work on logic. Therefore, logic is used for representing knowledge. The knowledge of logical agents has to be definite—a statement/proposition/formula has to be either true or false. The limitation of this type of logic is its inability to handle uncertain environments. A good knowledge representation naturally represents the problem domain. Other features of knowledge representation to be borne in mind are:

- It is often voluminous and relevant information or data must be mined from it.
- It is hard to characterize it accurately.
- It is constantly changing.
- It could be erroneous.

Knowledge representation consists of,

(i) **The knowledge base:** This forms the system's intelligence source. Facts and data form the knowledge base.

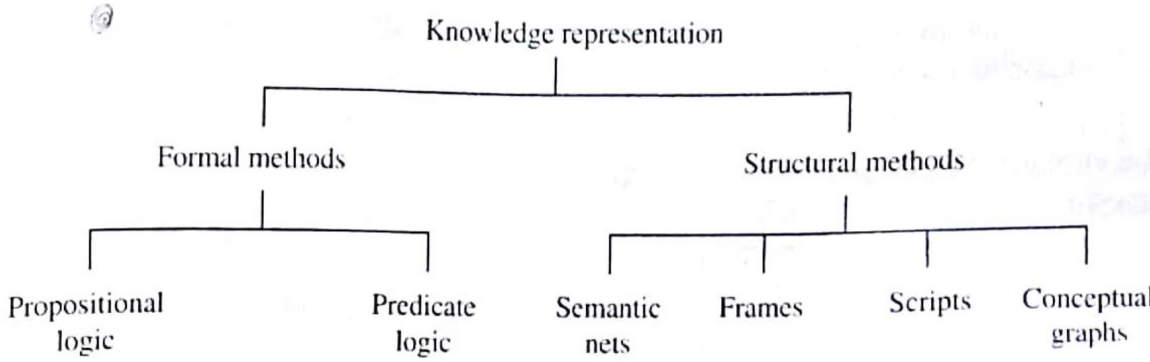

Figure 4.1 *Knowledge representation techniques.*

(ii) **Inference mechanism:** It examines the knowledge base to answer questions, solve problems or make decisions within the problem domain.

Knowledge representation schemes should have the following desirable properties:

- It can be programmed and stored in memory.
- It captures generalization
- It can be modified to correct errors
- It has flexibility to use under different situations.

4.2 Representations and Mappings

There are different ways of representing knowledge in AI programs. In any form of representation we deal with two entities:

- **Facts:** These are truths in their relevant world.
- **Representation of Facts:** This requires a formal structure, which we should be able to manipulate. These two entities can be structured in two levels:
  (i) The *knowledge level* at which facts are described.
  (ii) The *symbol level* where the representation of objects at the knowledge level are defined in terms of symbols that can be manipulated by programs.

The *forward mapping* maps facts to representations and *backward mapping* maps representations to facts. This is shown in Fig. 4.2.

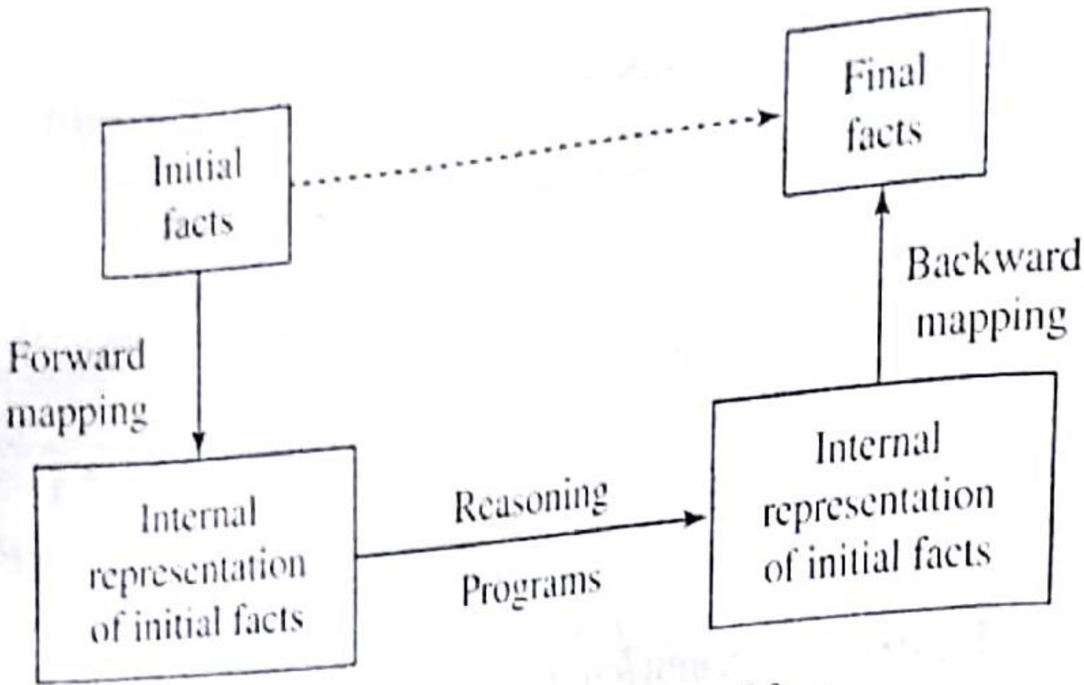

Figure 4.2 *Representation of facts.*

One of the most popular forms of representations is the natural language sentences. A simple example is given below.

Roopa is a girl (sentence)
girl (Roopa) (fact representation)

Suppose we have the fact that all girls have long hair. We can represent it symbolically as

$$\forall \, x{:}girl(x) \rightarrow longhair(x)$$

It translates into, for all x, if x is a girl, x has long hair. Now if we substitute x with Roopa, we conclude

longhair(Roopa)

The backward mapping gives us the sentence "Roopa has long hair".

4.3 Desirable Properties of Knowledge Representation

A good representation should possess the following properties:

(i) **Representational adequacy:** This is the ability to represent all kinds of knowledge needed in that domain.

(ii) **Inferential adequacy:** Ability to manipulate the representational structures in such a way as to derive new structures corresponding to new knowledge inferred from old.

(iii) **Inferential efficiency:** Ability to incorporate into the knowledge structure additional information that can be used to focus the attention of the inference mechanisms in the optimal direction (or rather most promising one)

(iv) **Aquisitional efficiency:** Ability to acquire new information easily.

There is no single system which optimizes all the capabilities. Hence, a number of techniques for knowledge representation exist. Some standard techniques are briefly described below.

4.3.1 Simple Relational Knowledge

Declarative facts are represented as a set of relations similar to that used in database systems. A simple relational database is shown in Table 4.1.

Table 4.1 Relational knowledge

Person	Gender	Height	Weight	Profession
A	Female	5′4″	58 kg	Student
B	Male	5′7″	72 kg	Software engineer
C	Male	6′2″	79 kg	Singer

This form of knowledge representation is simple, but weak in inferential capabilities. For example, if we ask the question, "What is the profession of a male, with a height 5′8″ and weight 69 kg?", the knowledge base may not be able to infer from the data base, unless exact matching of all fields is present in a particular data entry.

But knowledge in this form, can serve as input to more powerful inference engines.

4.3.2 Inheritable Knowledge

The relational knowledge corresponds to a set of attributes and their associated values which together describe the objects of the knowledge base. The basic representation must be augmented with inference mechanisms that operate on the structure of the

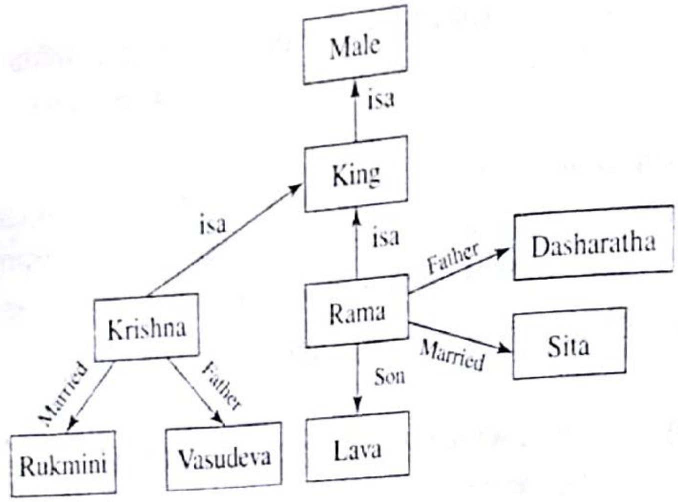

Figure 4.3 *Inheritable knowledge.*

representation. One of the useful forms of inference is *property inheritance* in which elements of specific classes inherit attributes and values from more general classes in which they are included. Objects are organized into classes and classes must be arranged in a generalization hierarchy. An example is shown in Fig. 4.3.

4.3.3 Inferential Knowledge

An example of inference logic is the way we deduced that "Roopa has long hair". This is because "Roopa is a girl" and "girls have long hair". Many procedures are available which reason forward from given facts to conclusions and others which reason backward from desired conclusions to given facts. One of the most commonly used procedures is *resolution* to be dealt later.

4.3.4 Procedural Knowledge

The representations described so for deal with relatively static, declarative facts. Procedural knowledge specifies what to do when. It can be represented in many ways. LISP is the language commonly used for this. The machine uses the knowledge when it executes the code to perform a task.

What are the different learning processes? Explain briefly

Learning Process:

Learning is a very complex process which involves interactions between the brain and the environment. In context of artificial neural networks we define learning (Simon Haykin) as follows:

"Learning is a process by which the free parameters of a neural network are adapted through a process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place". The learning process this involves the following steps:

- The neural network is stimulated (excitatory or inhibitory) by the environment.
- The free parameters of the network, namely the synaptic weights and the biases, undergo a change as a result of this stimulation.
- The response of the neural network to the environment changes as a result of the change in its internal free parameters.

The learning process involves

- Learning paradigms, which determine the manner in which a neural network relates to the environment. Here a neural network represents a set of inter connected neurons.
- Learning law which is a set of well defined rules called the *learning algorithm*. There is no unique algorithm. One law differs from the other in the manner in which the weights and biases are adjusted.

Learning paradigms:

In human beings, learning can take place without a teacher and with a teacher. Similarly in ANN too we have learning with a teacher called supervised learning and learning without a teacher called unsupervised learning.

Supervised Learning:

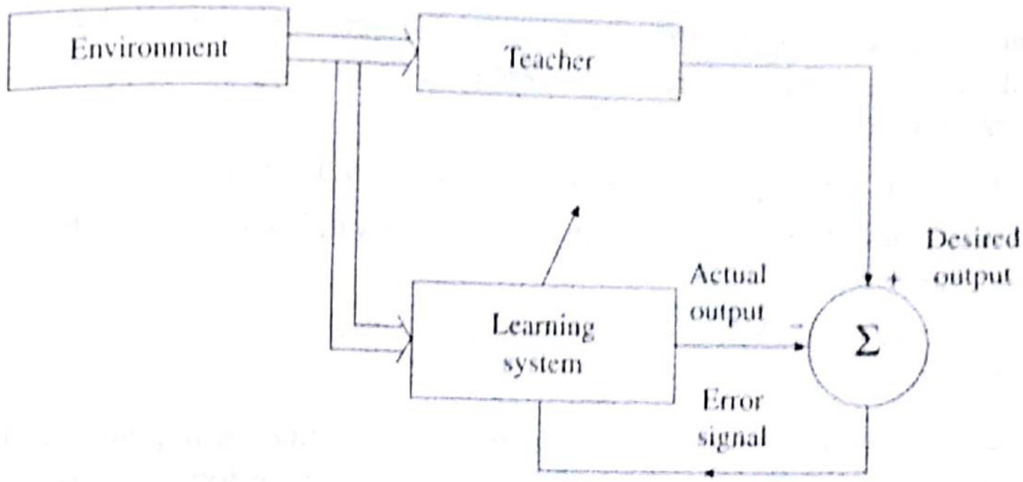

Figure 2.1 *Supervised learning.*

This takes place in the presence of a teacher as shown in Fig. 2.1.

The knowledge is represented in the form of input-output examples, called the *training data*. The training data consists of pairs of input samples, which is in the form of a vector and the corresponding output, which is the desired output. The teacher provides the desired output for a given input vector, by virtue of built-in-knowledge. The network computes the output, called the actual output. The error signal is the difference between the desired output (or response) and the actual output (response). The *network parameters*, namely the *synaptic weights* and *biases*, are adjusted by means of a learning law, under the influence of the error signal and the input vector.

This is done iteratively until the actual response is close to the desired response. At this point we say the network is trained. Thus, the steps involved in supervised learning are:

- Determine the type of training examples and form the training set. This is a set of input vector and the corresponding output vectors.

- In determining the input vector, the input must capture the features descriptive of the object. The number of features should not be too large, but large enough to predict the output.
- Determine the learning law and train the network with the input training data.
- After the training is completed, the network is validated with a subset of the training set.
- The performance is then measured from a test data which is different from the training set.

The mean-square error or the sum of squared errors over the training sample, defined as a function of the system parameters, can be used as a performance measure for the system. For the system to improve performance, the operating point must move towards a point of minimum error. Supervised training is used in tasks such as pattern classification and function approximation.

Unsupervised Learning:

In this class of learning, there is no teacher to oversee the process of learning. In unsupervised learning, we deal with a class of problems to determine how data is organized. The block diagram is shown in Fig. 2.2.

It is similar to the problem of density estimation in statistics. Once the network is tuned to the statistical regularities of the input data, it develops the ability to form internal representations for encoding features of the input. The Self-Organizing Map (SOM) is a topographic organization in which nearby locations in the map represent inputs with similar properties. It is a commonly used unsupervised learning algorithm. The adaptive Resonance Theory (ART) is another algorithm, which lets the user control the degree of similarity between members of the same cluster by means of a user-defined constant called the Vigilance parameter.

Unsupervised learning is often used in pattern recognition tasks.

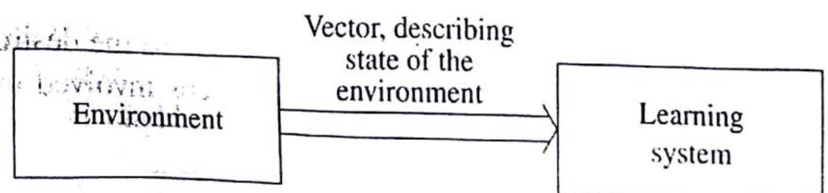

Figure 2.2 *Unsupervised learning.*

Reinforcement Learning:

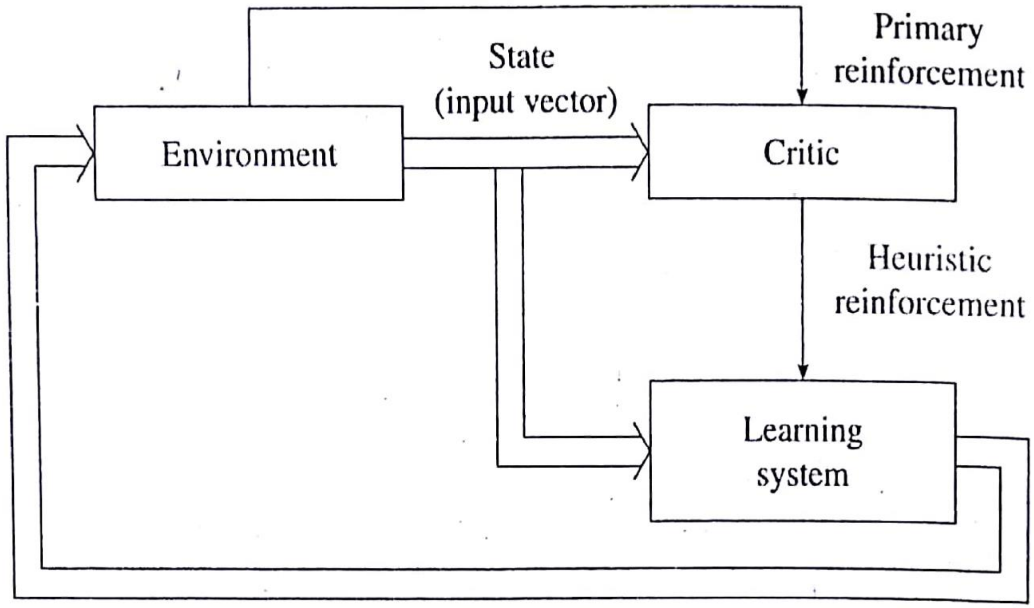

Figure 2.3 *Reinforcement learning.*

This type of learning also belongs to learning without a teacher. The block diagram is shown in Fig. 2.3.

Reinforcement learning is learning what to do, which means how to map situation to actions, so as to maximize a numerical reward signal (or minimize some index of performance). The learner is not told what actions to take (there is no teacher) but must discover which actions yield the maximum reward.

One such learning is shown in Fig 2.3. Here the learning is centered around a critic which converts a primary reinforcement signal received from the environment into a higher quality reinforcement signal called the heuristic reinforcement. The goal of learning is to maximize reward (or minimize a cost function) of actions taken over a sequence of steps, instead of the immediate action. Actions taken earlier in the sequence, may be the best determinants of the system performance. The learning process should discover these actions and feed them back to the environment.

To obtain a better reward the reinforcement agent has to exploit what it already knows and also explore new actions to make better selections. Neither exploitation nor exploration alone can be pursued exclusively. A typical example of reinforcement learning is a chess player-the choice of a move is based on anticipating possible replies and counter replies and by immediate, intuitive judgements of the desirability of particular positions and moves.

2) Discuss about the learning tasks?

Learning Tasks:

1) Pattern Association:

Association forms the basis of a large part of human memory. Association is of two types:

- Auto association, where the neural network is required to store a set of patterns by repeatedly presenting them to the network.
- Hetro association, where a set of input patterns is paired with a set of output patterns.

Auto association uses unsupervised learning laws, where as hetro association requires supervised learning.

Let $\mathbf{x}_k$ be an input vector (pattern) applied to an associative memory and $\mathbf{y}_k$ be the memorized vector (pattern). We can interpret $\mathbf{y}_k$ to be the output vector. The pattern association problem is one of mapping $\mathbf{x}_k$ to $\mathbf{y}_k$.

$$\mathbf{x}_k \rightarrow \mathbf{y}_k; \quad k = 1, 2, \ldots m$$

where m is the number of patterns stored. The input pattern (also called key pattern) acts as a stimulus and determines the storage location of y_k and also holds the key for its retrieval. In auto associative memory $\mathbf{y}_k = \mathbf{x}_k$, so the input and output spaces have same dimensionality. In hetro associative memory $\mathbf{y}_k \neq \mathbf{x}_k$. Therefore, the dimensionality of the output space may not be equal to that of the input space. The relation in associative memory is shown in Fig. 2.10.

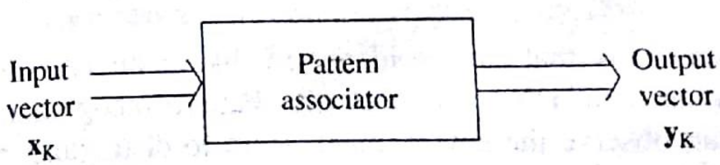

Figure 2.10 *Associative memory.*

During the training, the network is trained to map $\mathbf{x}_k$ with its corresponding $\mathbf{y}_k$. This is called the storage phase. In the recall phase, the memorized pattern ($\mathbf{y}_k$) is retrieved when the network is presented with a noise or distorted version of a key pattern.

The number 'm' of patterns stored provides a measure of the storage capacity of the network. Application of ANN associative memories are:

- Character recognition
- Hand writing recognition
- Noise filtering
- Data compression
- Information retrieval

Associative memories should have following desirable features:

- As large a capacity (m) of stored patterns as possible
- Data to be stored in a robust manner
- Network should be adaptable-addition and deletion of associations must be possible.

2) Pattern Recognition (Classification):

Human beings have highly sophisticated skills for sensing their environment and taking actions according to what they observe. For example,

- Recognizing a face
- reading handwriting
- distinguishing food from smell
- understanding speech

A pattern is an entity that can be identified, like a finger print image, speech signal, word,human face, DNA sequence etc. Pattern recognition is the study of how machines can observe the environment, learn to distinguish between patterns and assign a received pattern to one of a prescribed number of categories or classes. In pattern recognition, the neural network is trained with a set of input patterns and their categories. When a new pattern is presented to the network, the network is able to identify the class to which it belongs to Patterns are viewed as points in a multidimensional (hyperspace) *decision space*, which is divided into regions, each of which is associated with a class. The boundaries in the decision space are determined by the training process.

The process diagram of a pattern recognition system is shown in Fig. 2.11.

Two models of networks are possible.

Model 1:

The machine is made of two parts.

- An unsupervised network for feature extraction.
- A supervised network for classification.

This is shown in Fig. 2.12.

- The input **x** is a point in m-dimensional data space.
- **x** is mapped to a q-dimensional point **y** in feature space; $q < m$. Basically data is compressed by this transformation.
- y is mapped to one of the r classes, in an r-dimensional decision space.

Model 2:

We have a single multilayer feed forward network using supervised learning. Feature extraction is done by the hidden layers of the network.

The various steps in pattern recognition are:

- ***Data acquisition and sensing.***
  - Determine the measurement of physical variables.
  - Issues involved are bandwidth, resolution, sensitivity, distortion, SNR, Latency etc.
- ***Pre-processing***
  - Removal of noise in data
  - Isolation of patterns of interest.

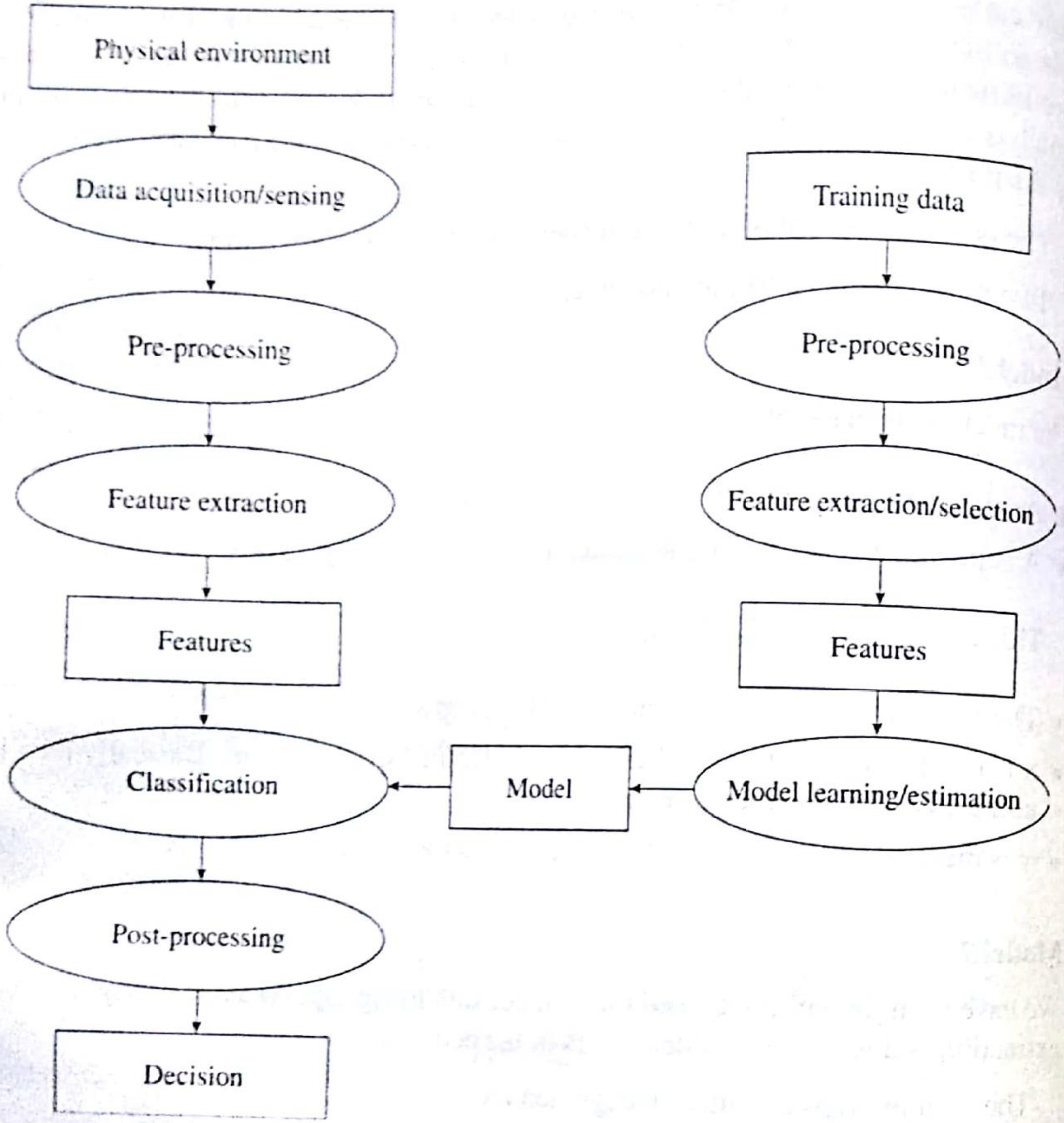

Figure 2.11 *Process diagram of a pattern-recognition system.*

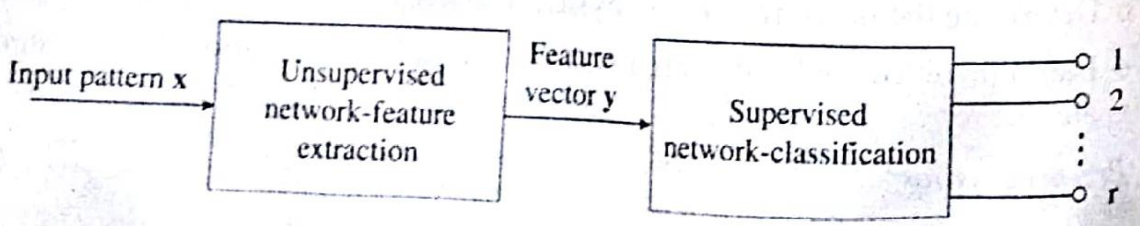

Figure 2.12 *Pattern recognition.*

- *Feature extraction*
  - Determine features which can distinguish patterns of one class from another without ambiguity.
- Learning
  - To map features of pattern groups/classes/categories
- Classification
  - Using features of the pattern presented, and learned models, assign the pattern to a category
- post-processing
  - Evaluation of confidence in decisions
  - Exploitation of context to improve performance
  - Combination of experts.

Some typical applications are:

- Junk mail filtering
- Information extraction
- Biometric recognition
- Medical diagnosis
- Automatic target recognition
- Internet search
- Data mining

3) Function Approximation:

Consider an input-output mapping described by the relationship

$$d = f(\mathbf{x}) \tag{2.20}$$

Here the function $f(\cdot)$ is unknown. We have knowledge about a set of input-output pairs,

$$S = \{(\mathbf{x}_i; \mathbf{d}_i)\}_{i=1}^{N} \tag{2.21}$$

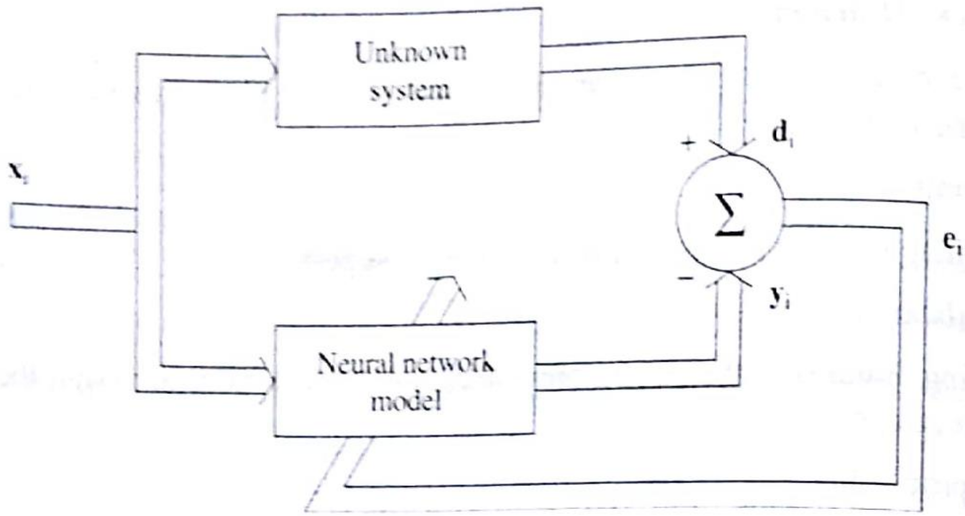

Figure 2.13 *System identification.*

We now design a neural network that approximates the function $f(\cdot)$ to $F(\cdot)$ such that $F(\cdot)$ is close to $f(\cdot)$.

$$||F(\mathbf{x}) - f(\mathbf{x})|| < \epsilon \quad \text{for all } x \tag{2.22}$$

where ϵ is a small number. To make the error small N should be large and the network must have adequate number of free parameters. The function approximation is suitable for application of supervised learning, where $\mathbf{x}_i$ is the input vector and $\mathbf{d}_i$ is the desired response. The function approximation can be applied in system identification. In system identification the mathematical model of the system is unknown. We however have measured data of input $\mathbf{x}_i$ and corresponding output d_i. (This is an $MIMO$ system since $\mathbf{x}_i$ and $\mathbf{d}_i$ are vectors). Now the neural network is trained to minimize the error between the actual response y_i and desired response d_i. The identification problem is depicted in Fig. 2.13.

Reversing the roles of $\mathbf{d}_i$ and $\mathbf{x}_i$, we can have an *inverse system*, where in we determine the vector $\mathbf{x}_i$ which will produce the desired response $\mathbf{d}_i$ as shown in Fig. 2.14.

4) Control Applications:

The use of neural networks in control applications like process control, robotics, aerospace etc has grown rapidly. The basic objective of control is to provide the

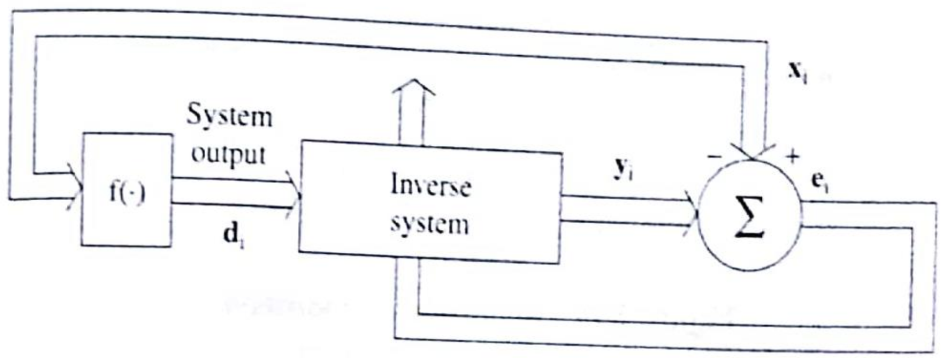

Figure 2.14 *Inverse system.*

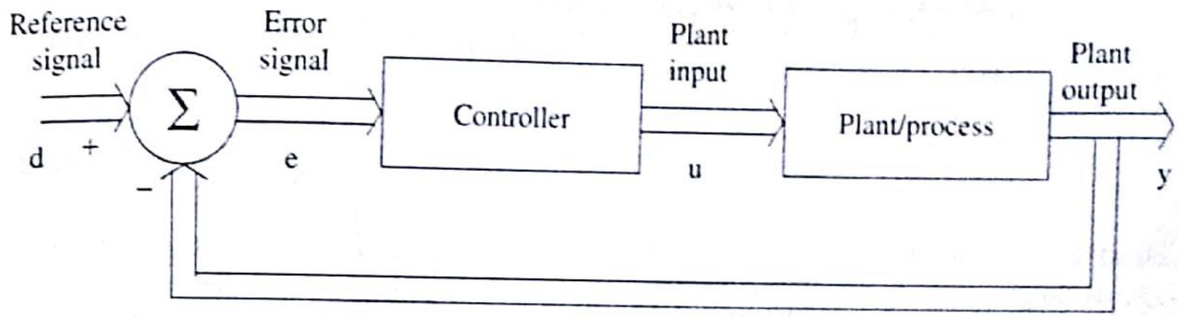

Figure 2.15 *ANN for control application.*

appropriate input signal to a given physical process to yield the desired response. It is similar to reverse identification of the plant and is shown in Fig. 2.15.

The actual plant output **y** is subtracted from the reference signal **d**, which is the desired output, to give the error signal. This error signal is applied to the neural controller to adjust its free parameters with the objective of making **y** track **d**.

Multilayer feed forward networks can be used for this.

5) Filtering:

Filters are used to extract information about a particular quantity of interest from a set of noisy data. There are typical applications, discussed below:

(i) Extraction of information regarding a particular quantity of interest at a discrete time *n*, using data measured up to and including *n*. This is typically referred to as filtering. One such application of ANN is in Blind signal separation. ***Blind signal***

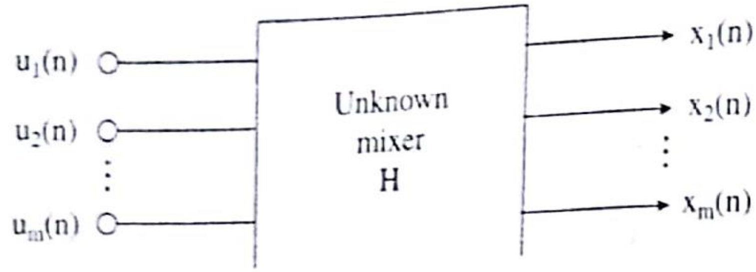

Figure 2.16 *Blind signal separation.*

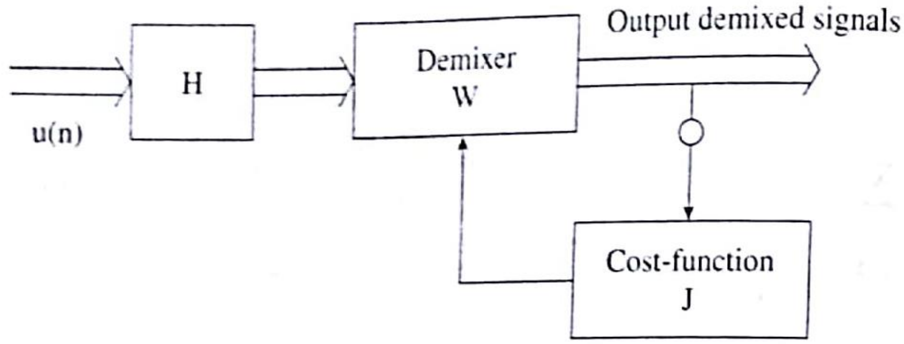

Figure 2.17 *Signal recovery.*

separation is the task of separating signals when they are available in mixed form for observation. The process is termed blind because generally the source signals and mixing procedures are unknown. The source signals may be images, audio signals, biomedical data, financial data etc.

Consider Fig. 2.16. Given the observation vector $\mathbf{x}(n)$, the problem is to recover the original signals $u_1(n), u_2(n), \ldots u_m(n)$. Source signals have to be statistically independent. The determining criterion for separation is a measure of independence, typically represented by a cost function J. J is optimized to yield more or less independent outputs as shown in Fig. 2.17.

(ii) Prediction or forecasting information at time $n + n_0$ based on measurements (observations) available up to time n as shown in Fig. 2.18.

We predict $\mathbf{x}(n)$ based on $x(n-T), x(n-2T), \ldots x(n-mT)$, where T is the sampling period and m is the order of prediction. Error correction based unsupervised learning can be used, outliers in data can affect performance badly. An *outlier* is an observation or data which deviates markedly from other members of the samples in which it occurs. In most large samplings of data, some data points will be further away from the sample mean than what is acceptable. They can indicate faulty

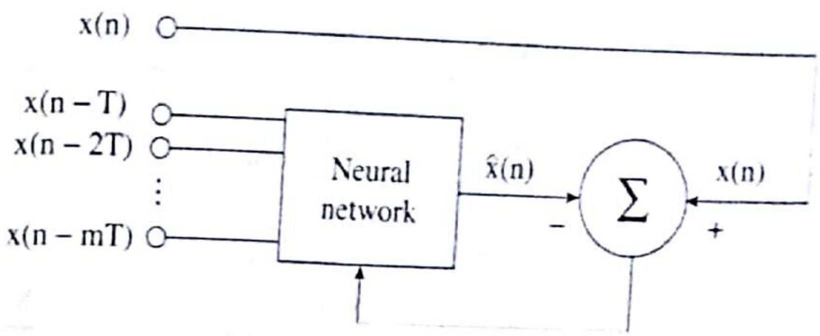

Figure 2.18 *Forecasting.*

data, erroneous procedure or measurement errors. They must be removed before presenting data for the application.

6) Beam Forming:

Beam forming is a signal processing technique used in sensor arrays for directional signal transmission or reception. It is a spatial form of filtering and is used to distinguish between the spatial properties of a target signal and background noise. The device used for this is called the beam former. It has found application in radar, sonar, seismology, wireless communication, speech, acoustics and biomedicine. Adaptive beam forming is used to detect and estimate the signal-of-interest at the output of a sensor array by means of data-adaptive spatial filtering and interference rejection. Two issues are important:

- The *target signal* (which has to be tracked) originates from an unknown direction.
- Interfering signals are not deterministic and not known apriori.

Beam forming using ANN is shown in Fig. 2.19 where a *Generalized Side Lobe Canceller* (GSLC) is used.

Adjustments of the free parameters of the ANN are performed by error correction learning law.

Thus the tasks which can be performed by neural networks are varied and universal in signal-processing systems. Invariably all learning involves mapping of information and therefore involves some form of pattern recognition task or the other.

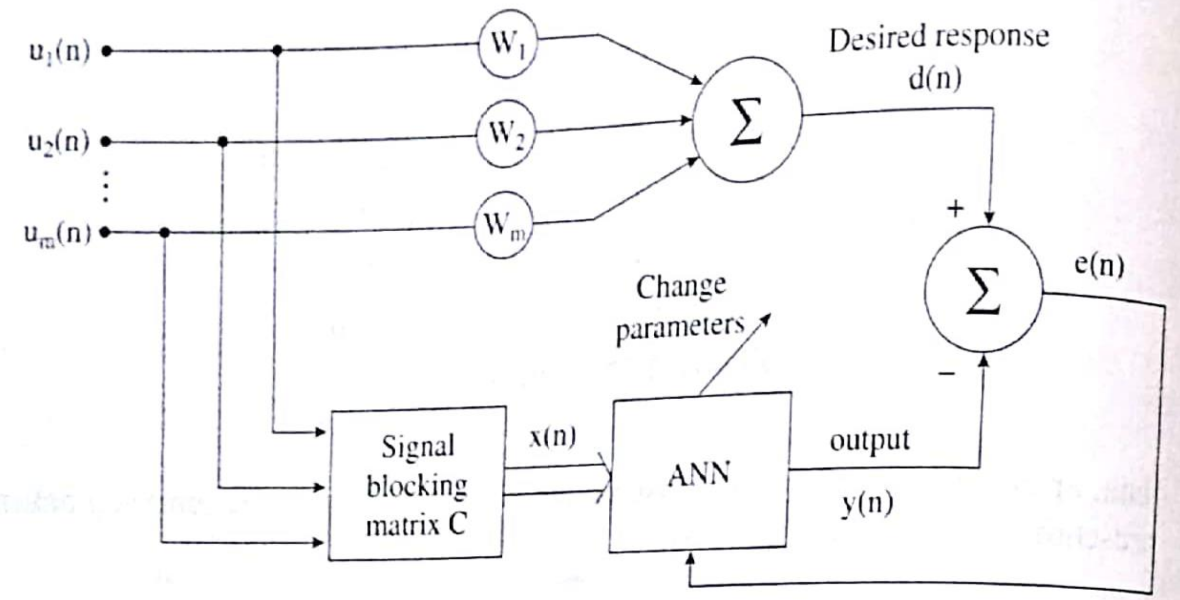

Figure 2.19 *Beam forming.*

Discuss the methods of AI techniques?

Methods of AI Techniques:

Different areas of research and study have branched under the umbrella of Artificial Intelligence. Lets see some of them.

- **Logical AI:** Here knowledge is represented by facts and rules. What a program knows about the world in general, the facts of the specific situation in which it must act, and its goals are all represented by sentences of some mathematical logical language. The program decides what to do by inferring that certain actions are appropriate for achieving its goals. The two most popular logics used are the propositional logic and the first order predicate logic.

- **Search methods:** Many applications require searching through a large input space to locate a relevant piece of information. AI search programs examine large numbers of possibilities, in a structured fashion, to arrive at the goal state. Examples are moves in a chess game, locating an ancestor in the family tree or inferences by a theorem proving program. Search methods are continually evolving to make them more effective, efficient and optimal, in various domains.

...nition is the scientific discipline whose more effective, efficient and ...

- **Pattern recognition:** Pattern recognition is the scientific discipline whose g is to classify objects into a number of classes or categories. Depending on application, these objects could be images, signals or any other measurement t needs to be classified. Pattern recognition is an integral part of most machi intelligence systems used for decision making.

- **Inference:** Inference is the process where in given a set of facts and rules, oth facts can be derived from them. Inference is a powerful tool for drawing inferenc from the knowledge base. Mathematical logic is used for inference.

- **Expert systems:** These are AI programs that achieve expert-level competence problem solving, in task areas, by using knowledge about specific tasks in a specif domain. They are also called *knowledge-based systems*. Often, the term expe systems is reserved for programs whose knowledge base contains the knowledg used by human experts, in contrast to knowledge gathered from textbooks or nor experts. The area of human intellect to be captured in an expert system is called th *task domain*. *Task* refers to some goal-oriented, problem-solving activity. *Domai* refers to the area within which the task is being performed. Typical tasks ar diagnosis, planning, scheduling, configuration and design. Some applications o expert systems are medical diagnosis, trouble shooting, financial decision making process monitoring and control, etc.

- **Epistemology:** This is the study of the different kinds of knowledge required for problem solving.

- **Ontology:** Ontological analysis clarifies the structure of knowledge. Given a domain, its ontology forms the heart of any system of knowledge representa- tion for that domain. Without ontologies, or the conceptualizations that underlie knowledge, there cannot be a vocabulary for representing knowledge. Secondly, ontologies enable knowledge sharing by providing a common platform for knowl- edge representation. Consider a simple example to illustrate the importance of knowledge representation. Suppose we were to ask, what is the date today, we can have three different answers : It is 5^{th} September. 05/09/2010 (if we are in India we write the date first and then the month) or 09/05/2010 (the month first and then the date!). Clearly non uniformity in the representation can lead to confusion. Ontology is the branch which precisely deals with this.

- **Genetic programming:** This deals with generating solutions in a given problem domain, to optimize a given function. The search is carried out from one iteration to another by modifying the solution , using techniques of mutation and cross-over, from genetics. Hence, the name.
- **Heuristics:** It is a branch of AI dealing with experience–based techniques for problem solving, learning and discovery. Heuristic methods and techniques is all about quickly and efficiently discovering an optimal solution for a specific problem. It uses a certain 'rule of thumb'(from experience) to make an educated guess, an informed decision or simply using common sense.