Dated:

ARTIFICIAL INTELLIGENCE:

Before learning about "ai" we must understand the meaning of word "Intelligence".

Intelligence: "The ability to learn and solve problems. And also learn from past failure experiences and correct them"

Basic terms on which intellegence is composed are;

* Reasoning
* Learning
* Problem solving
* Perception
* Linguistic Intellegence

After overviewing the definition of Intellegence we can easily conclude that artificial intellegence is that branch of computer science field which deals to give computer intellegence so that it can solving normal living problem by itself.

Importance of Artificial Intellegence:

- To create systems that exhibit intellegent behaviour with the capability to learn demonstrate, explain and advise its users.

- Helping machines find solutions to complex problems like humans do and applying them as algorithms in a computer-friendly manner.

General analytical tasks, including finding patterns in data, that have been performed by software for many years can also be performed more effectively using AI.

Use cases are proliferating as AI's potential is understood. We describe 31 core use cases across eight sectors: Assetmanagenment, healthcare insurance, law and compilance, manufaturing, retail, transport and utilities.

Dated:

Approaches of Artificial Intellegence:

There are total four approaches of Artificial Intellegence and that are as follows;

* Acting humanly (The turing approach):

This approach was designed by Alan Turing. The ideology behind this approach is that a computer passes the test if a human interrogator, after asking some written responses come from a human or from a computer. Afterwards we'll study this approach in detail.

* Thinking humanly (cognitive modeling approach):

The idea behind this approach is to determine whether the computer thinks like a human.

* Thinking rationally (The laws of thought approach):

The idea behind this approach is to determine whether the computer thinks rationally i.e logical reasoning etc.

* Acting rationally (The rational agent approach):

The idea behind this approach is to determine whether the computer acts rationally

Difference between Artificial Intelligence and Machine Learning:

Many times we get confused between AI and ML but Machine Learning, a fundamental concept of AI research since the field inception, is the study of computer algorithms that improve automatically through experience. The mathematical analysis of machine Learning and algorithms and their performance is the branch of theoretical computer science known as a computational learning theory.

Dated:

Expert Systems:

Exper systems (ES) are one of the prominent research domains of AI. It is introduced by the researchers at Stanford University. The ES are the computer applications developed to solve complex problems in a particular domain, at the level of extra ordinary human intellegence and expertise.

Characterstics:

* High performance
* Reliable
* Understandable
* Highly responsive

Capabilities:

* Advising
* Instructing and assisting human in decision making
* Demonstrating
* Deriving a solution
* Diagnosing.
* Explaining
* Interpreting input
* Predicting results.
* Justifying the conclusion
* suggesting alternative options to a problem.

Incapabilities:

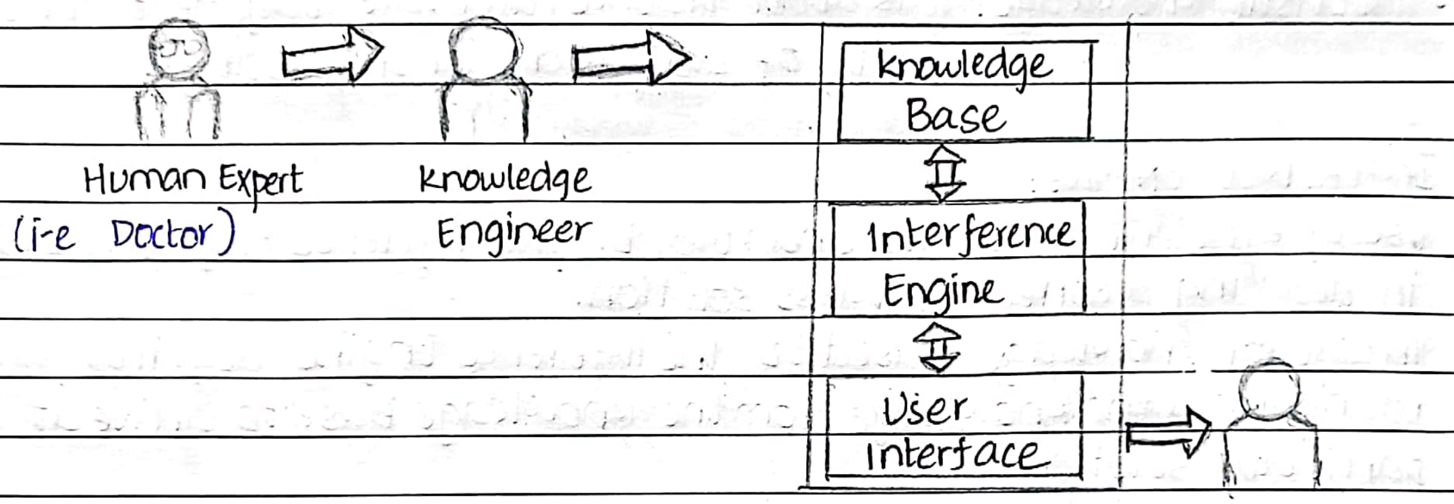
* Substituting human desicion makers
* Possessing human capabilities
* Producing accurate output for inadequate knowledge base.
* Refining their own knowledge.

Dated:

Basic component of Expert Systems:

* Knowledge Base
* Interference Engine
* User Interface
* Explanation facility.

Ps: Mujhe pata hai bht buri bni hai :-)



Knowledge Base:

It contain domain-specific and high quality knowledge. knowledge is required to exihbit intellegence. The success of any ES majorly depends upon the collection of highly accurate and precise knowledge.

Let us know that what is the basic definition of "knowledge" in terms of Computer.

Knowledge: The data is collection of facts. The information is organized as data and facts about the task domain. Data information, and past experience combined together are termed as knowledge.

Dated:

Components of knowledge Base:

The knowledge base of an ES is a store of both factual and heuristic knowledge. What are those?

- Fatual knowledge: It is the information widely accepted by the knowledge Engineers and scholars in task domain
- Heuristic knowledge: It is about practice, accurate judgement, One's ability of evaluation, and guessing.

INFERENCE ENGINE:

Use of efficient procedures and rules by the Inference Engine is essential in deducing a correct, flawless solution.

In case of knowledge-based ES, the Inference Engine acquires and manipulates the knowledge from the knowledge base to arrive at a particular solution.

To recommend a solution the Inference Engine uses following strategies:

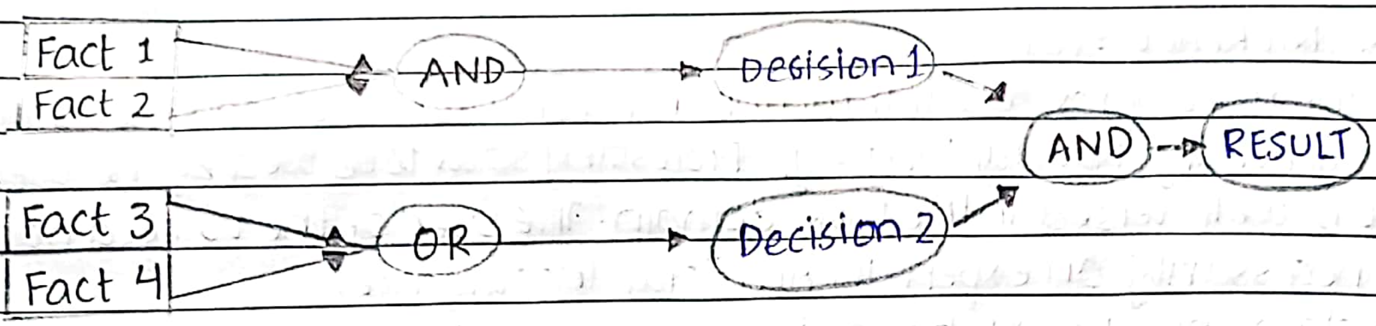
- Data Driven Approach aka FORWARD chaining.
- Goal Driven Approach aka Backward Chaining.

FORWARD CHAINING:

It is a strategy of an expert system to answer the question "What can happen next?"

Here, the Inference Engine follows the chain of conditions and derivations and finally deduces the outcome. It considers all the facts and rules and sorts them before concluding to a solution.

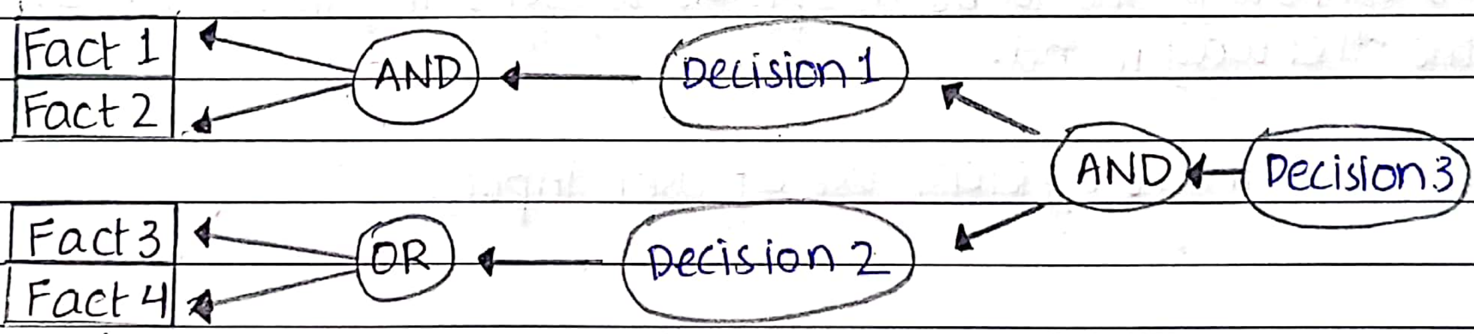
Dated:



This strategy is followed for working on conclusion, result, or effect.
For example,
prediction of share market status as an effect of changes in interest rates.

BACKWARD CHAINING:
With this strategy, an expert system finds out the answer to the question,
"Why this happened?"

On the basis of what has already happened, the Inference Engine tries to find out which conditions could have happened in the past for this result. This strategy is followed for finding out cause or reason.
For example,
Diagnosis of blood cancer in humans.



Scanned with CamScanner

Dated:

USER INTERFACE : (UI)

User interface provides interaction between user of ES and the ES itself. It is generally Natural Language Processing so as to be used by the user who is well-versed in the task domain. The user of the ES need not be necessarily an expert in Artificial Intelligence.
It explains how the ES has arrived at a particular recommendation. The explanation may appear in the following forms.

- Natural language displayed on screen.
- Verbal narrations in natural language.
- Listening of rule numbers displayed on the screen.

The user interface makes it easy to trace the creadibility of the deductions.

Requirements of Efficient ES User Interface:

- It should help users to accomplish their goals in shortest possible way
- It should be designed to work for user's existing or desired work practices.
- Its technology should be adaptable to user's requirements; not the other way round.
- It should make efficient use of user input.

Dated:

Expert Systems Limitations:

No technology can offer easy and complete solution. Large systems are costly, require significant development time, and computer resources ESs have their limitations which include;

- Limitations of technology
- Difficult knowledge acquisition
- ES are difficult to maintain
- High development costs.

Applications of Expert system:

Design Domain: Camera lens design
Automobile design

Medical Domain: Diagonosis systems to deduce cause of disease from observed data, conduction medical operations on humans.

Monitoring systems: comparing data continuosly with observed system or with prescribed behaviour such as leakage monitoring in long petroleum pipeline.

Process control system: controlling a physical process based on monitoring.

knowledge Domain: Finding out faults in vehicles, computers.

Finance /commerce: Detection of possible frauds, suspicious transaction stock market trading, Airline scheduling, cargo scheduling.

The Turing Test

In the past the **Turing test** (proposed by Alan Turing in 1950) has served as a benchmark in measuring progress in the field of artificial intelligence. Today the significance of the Turing test has faded although it remains an important part of the artificial intelligence folklore. Turing's proposal was to allow a human, whom we call the interrogator, to communicate with a test subject by means of a typewriter system without being told whether the test subject was a human or a machine. In this environment, a machine would be declared to behave intelligently if the interrogator was not able to distinguish it from a human. Turing predicted that by the year 2000 machines would have a 30 percent chance of passing a five-minute Turing test—a conjecture that turned out to be surprisingly accurate.

The Origins of Artificial Intelligence

The quest to build machines that mimic human behavior has a long history, but many would agree that the modern field of artificial intelligence had its origins in 1950. This was the year that Alan Turing published the article "Computing Machinery and Intelligence" in which he proposed that machines could be programmed to exhibit intelligent behavior. The name of the field—*artificial intelligence*—was coined a few years later in the now legendary proposal written by John McCarthy who suggested that a "study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College" to explore "the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it."

One reason that the Turing test is no longer considered to be a meaningful measure of intelligence is that an eerie appearance of intelligence can be **produced** with relative ease. A well-known example arose as a result of the program DOCTOR (a version of the more general system called ELIZA) developed by Joseph Weizenbaum in the mid-1960s. This interactive program was designed to project the image of a Rogerian analyst conducting a psychological interview; the computer played the role of the analyst while the user played the patient. Internally, all that DOCTOR did was restructure the statements made by the patient according to some well-defined rules and direct them back to the patient. For example, in response to the statement "I am tired today," DOCTOR might have replied with "Why do you think you're tired today?" If DOCTOR was unable to recognize the sentence structure, it merely responded with something like "Go on" or "That's very interesting."

Weizenbaum's purpose in developing DOCTOR dealt with the study of natural language communication. The subject of psychotherapy merely provided an environment in which the program could "communicate." To Weizenbaum's dismay, however, several psychologists proposed using the program for actual psychotherapy. (The Rogerian thesis is that the patient, not the analyst, should lead the discussion during the therapeutic session, and thus, they argued, a computer could possibly conduct a discussion as well as a therapist could.) Moreover, DOCTOR projected the image of comprehension so strongly that many who "communicated" with it became subservient to the machine's question-and-answer dialogue. In a sense, DOCTOR passed the Turing test. The result was that ethical, as well as technical, issues were raised, and Weizenbaum became an advocate for maintaining human dignity in a world of advancing technology.

More recent examples of Turing test "successes" include Internet viruses that carry on "intelligent" dialogs with a human victim in order to trick the human into dropping his or her malware guard. Moreover, phenomena similar to Turing tests occur in the context of computer games such as chess-playing programs. Although these programs select moves merely by applying brute-force techniques (similar to those we will discuss in Section 11.3), humans competing against the computer often experience the sensation that the machine possesses creativity and even a personality. Similar sensations occur in robotics where machines have been built with physical attributes that project intelligent characteristics. Examples include toy robot dogs that project adorable personalities merely by tilting their heads or lifting their ears in response to a sound.

Questions & Exercises

1. Identify several types of "intelligent" actions that might be made by an agent.
2. A plant placed in a dark room with a single light source grows toward the light. Is this an intelligent response? Does the plant possess intelligence? What, then, is your definition of intelligence?
3. Suppose a vending machine is designed to dispense various products depending on which button is pressed. Would you say that such a machine is "aware" of which button is pressed? What, then, is your definition of awareness?

If a machine passes the Turing test, would you agree that it is intelligent? If not, would you agree that it appears to be intelligent?

Suppose you used a chat room to chat with someone over the Internet (or used Instant Messenger) and carried on a meaningful coherent conversation for ten minutes. If later you found out that you had conversed with a machine, would you conclude that the machine was intelligent? Why or why not?

Strong AI Versus Weak AI

The conjecture that machines can be programmed to exhibit intelligent behavior is known as **weak AI** and is accepted, to varying degrees, by a wide audience today. However, the conjecture that machines can be programmed to possess intelligence and, in fact, consciousness, which is known as **strong AI**, is widely debated. Opponents of strong AI argue that a machine is inherently different from a human and thus can never feel love, tell right from wrong, and think about itself in the same way that a human does. However, proponents of strong AI argue that the human mind is constructed from small components that individually are not human and are not conscious but, when combined, are. Why, they argue, would the same phenomenon not be possible with machines?

The problem in resolving the strong AI debate is that such attributes as intelligence and consciousness are internal characteristics that cannot be identified directly. As Alan Turing pointed out, we credit other humans with intelligence because they behave intelligently—even though we cannot observe their internal mental states. Are we, then, prepared to grant the same latitude to a machine if it exhibits the external characteristics of consciousness? Why or why not?

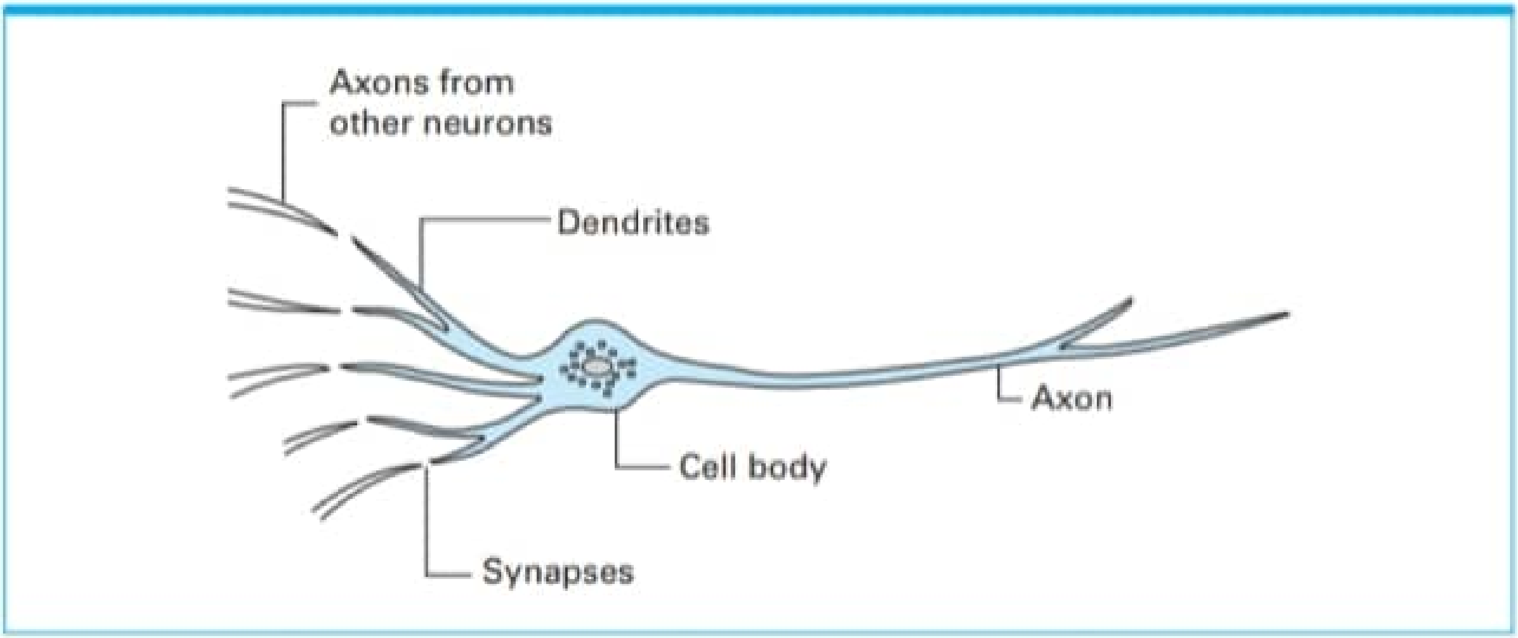
11.5 Artificial Neural Networks

With all the progress that has been made in artificial intelligence, many problems in the field continue to tax the abilities of computers using traditional algorithmic approaches. Sequences of instructions do not seem capable of perceiving and reasoning at levels comparable to those of the human mind. For this reason, many researchers are turning to approaches that leverage phenomena observed in nature. One such approach is genetic algorithms presented in the previous section. Another approach is the artificial neural network.

Basic Properties

Artificial neural networks provide a computer processing model that mimics networks of neurons in living biological systems. A biological neuron is a single cell with input tentacles called dendrites and an output tentacle called the axon (Figure 11.15). The signals transmitted via a cell's axon reflect whether the cell is in an inhibited or excited state. This state is determined by the combination of signals received by the cell's dendrites. These dendrites pick up signals from the axons of other cells across small gaps known as synapses. Research suggests that

Figure 11.15 A neuron in a living biological system



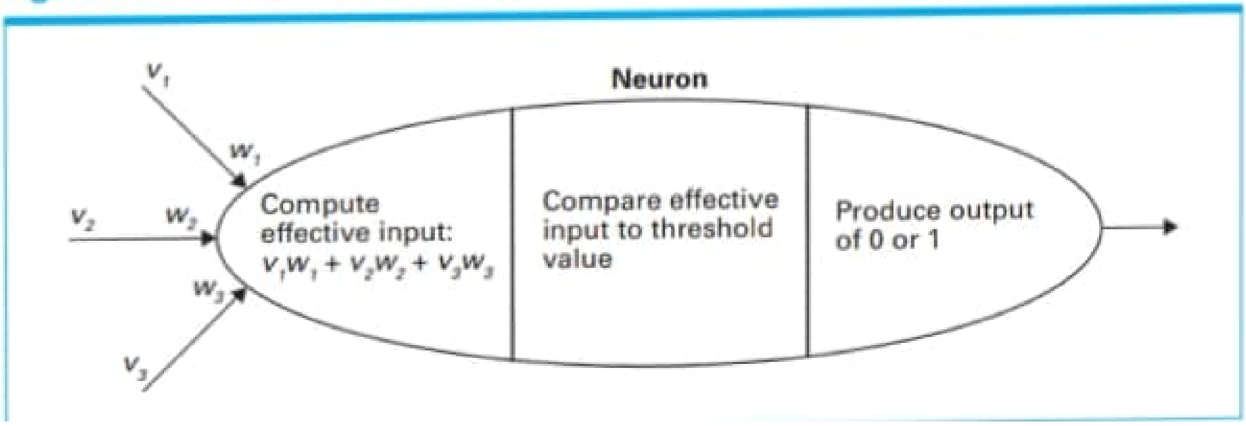
the conductivity across a single synapse is controlled by the chemical composition of the synapse. That is, whether the particular input signal will have an exciting or inhibiting effect on the neuron is determined by the chemical composition of the synapse. Thus it is believed that a biological neural network learns by adjusting these chemical connections between neurons.

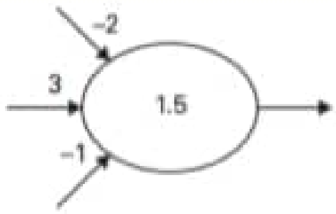
A neuron in an artificial neural network is a software unit that mimics this basic understanding of a biological neuron. It produces an output of 1 or 0, depending on whether its effective input exceeds a given value, which is called the neuron's **threshold** value. This effective input is a weighted sum of the actual inputs, as represented in Figure 11.16. In this figure, a neuron is represented with an oval and connections between neurons are represented with arrows. The values obtained from the axons of other neurons (denoted by v_1, v_2, and v_3) are used as inputs to the depicted neuron. In addition to these values, each connection is associated with a **weight** (denoted by w_1, w_2, and w_3). The neuron receiving these input values multiplies each by the associated weight for the connection and then adds these products to form the effective input ($v_1w_1 + v_2w_2 + v_3w_3$). If this sum exceeds the neuron's threshold value, the neuron produces an output of 1 (simulating an excited state); otherwise the neuron produces a 0 as its output (simulating an inhibited state).

Following the lead of Figure 11.16, we adopt the convention of representing neurons as circles. Where each input connects to a neuron, we record the weight associated with that input. Finally, we write the neuron's threshold value in the middle of the circle. As an example, Figure 11.17 represents a neuron with a threshold value of 1.5 and weights of −2, 3, and −1 associated with each of its input connections. Therefore if the neuron receives the inputs 1, 1, and 0, its effective input is $(1)(-2) + (1)(3) + (0)(-1) = 1$, and thus its output is 0. But, if the neuron receives 0, 1, and 1, its effective input is $(0)(-2) + (1)(3) + (1)(-1) = 2$, which exceeds the threshold value. The neuron's output will thus be 1.

The fact that a weight can be positive or negative means that the corresponding input can have either an inhibiting or exciting effect on the receiving neuron. (If the weight is negative, then a 1 at that input position reduces the weighted sum and thus tends to hold the effective input below the threshold value. In contrast, a positive weight causes the associated input to have an increasing effect on the weighted sum and thus increase the chances of that sum exceeding the threshold value.) Moreover, the actual size of the weight controls the degree to which the corresponding input is allowed to inhibit or excite the receiving neuron. Consequently, by adjusting the values of the weights throughout an artificial neural network, we can program the network to respond to different inputs in a predetermined manner.

Figure 11.16 The activities within a neuron





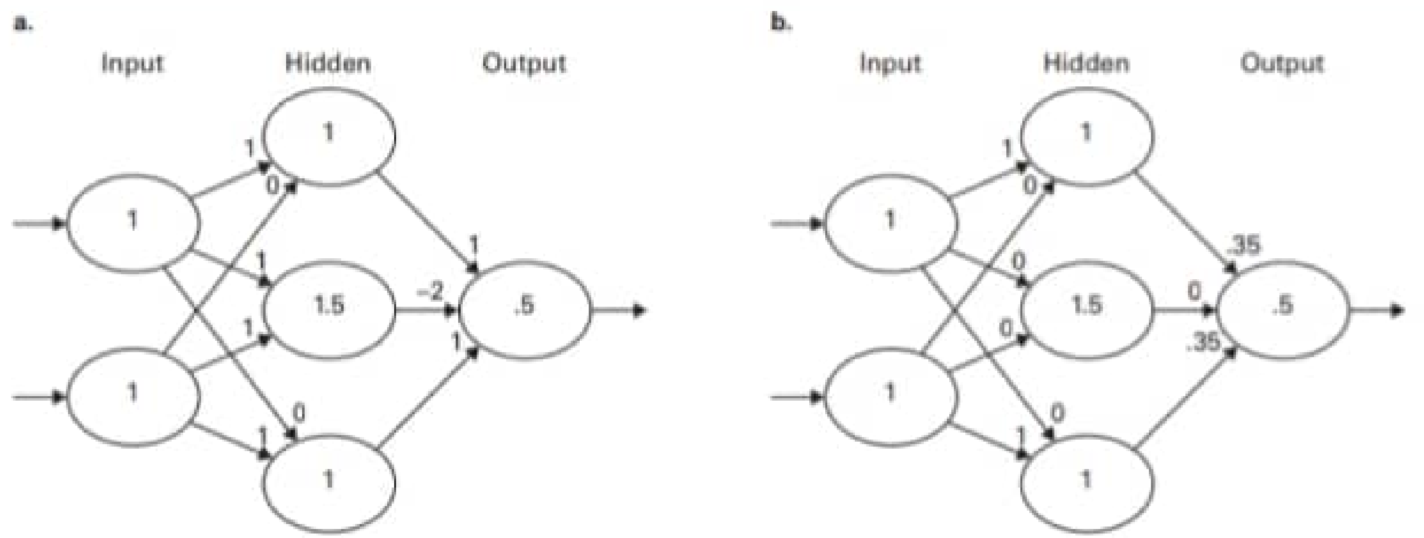
Artificial neural networks are typically arranged in a topology of several layers. The input neurons are in the first layer and the output neurons are in the last. Additional layers of neurons (called hidden layers) may be included between the input and output layers. Each neuron of one layer is interconnected with every neuron in the subsequent layer. As an example, the simple network presented in Figure 11.18a is programmed to produce an output of 1 if its two inputs differ and an output of 0 otherwise. If, however, we change the weights to those shown in Figure 11.18b, we obtain a network that responds with a 1 if both of its inputs are 1s and with a 0 otherwise.

We should note that the network configuration in Figure 11.18 is far more simplistic than an actual biological network. A human brain contains approximately 10^{11} neurons with about 10^{4} synapses per neuron. Indeed, the dendrites of a biological neuron are so numerous that they appear more like a fibrous mesh than the individual tentacles represented in Figure 11.15.

Training Artificial Neural Networks

An important feature of artificial neural networks is that they are not programmed in the traditional sense but instead are trained. That is, a programmer does not determine the values of the weights needed to solve a particular problem and then "plug" those values into the network. Instead, an artificial neural

Figure 11.18 A neural network with two different programs



network learns the proper weight values via supervised training (Section 11.4) involving a repetitive process in which inputs from the training set are applied to the network and then the weights are adjusted by small increments so that the network's performance approaches the desired behavior.

It is interesting to note how genetic algorithm techniques have been applied to the task of training artificial neural networks. In particular, to train a neural network, a number of sets of weights for the network can be randomly generated—each set of which will serve as a chromosome for the genetic algorithm. Then, in a step-by-step process, the network can be assigned the weights represented by each chromosone and tested over a variety of inputs. The chromosones producing fewest errors during this testing process can then be given a greater probabilty of being selected as parents for the next generation. In numerous experiments this appoarch has ultimately led to a successful set of weights.

Let us consider an example in which training an artificial neural network to solve a problem has been successful and perhaps more productive than trying to provide a solution by means of traditional programming techniques. The problem is one that might be faced by a robot when trying to understand its environment via the information it receives from its video camera. Suppose, for example, that the robot must distinguish between the walls of a room, which are white, and the floor, which is black. At first glance, this would appear to be an easy task: Simply classify the white pixels as part of a wall and the black pixels at part of the floor. However, as the robot looks in different directions or moves around in the room, various lighting conditions can cause the wall to appear gray in some cases whereas in other cases the floor may appear gray. Thus, the robot needs to learn to distinguish between walls and floor under a wide variety of lighting conditions.

To accomplish this, we could build an artificial neural network whose inputs consist of values indicating the color characteristics of an individual pixel in the image as well as a value indicating the overall brightness of the entire image. We could then train the network by providing it with numerous examples of pixels representing parts of walls and floors under various lighting conditions.

The results of training an artificial neural network using these techniques are represented in Figure 11.19. The first column represents the original images; the next depicts the robot's interpretation. Note that although the walls in the top original are rather dark, the robot has correctly identified most of the associated pixels as white wall pixels, yet the floor in the lower image has still been correctly identifed. (The ball in the images was part of a more extensive experiment.) You will also notice that the robot's image processing system is not perfect. The neural network has mistakenly identified some of the wall pixels as floor pixels (and some of the floor pixels as wall pixels). These are examples of realities that often must be accommodated in the application of a theory. In th[illegible] case, the errors can be corrected by programming the rob[illegible]ivid[illegible] floor pixels that appear among a multitude of wall pixels [illegible]

Beyond simple learning problems (such as the classi[illegible]), a[illegible]ficial neural networks have been used to learn sophisticated intelligent beha[illegible]ior, as testified by the ALVINN project cited in the previous section. Indeed, ALVINN was an artificial neural network whose composition was surprisingly simple (Figure 11.20). Its input was obtained from a 30 by 32 array of sensors,

Figure 11.19 Results of using a neural network to classify pixels in an image (Inspired by www.actapress.com)

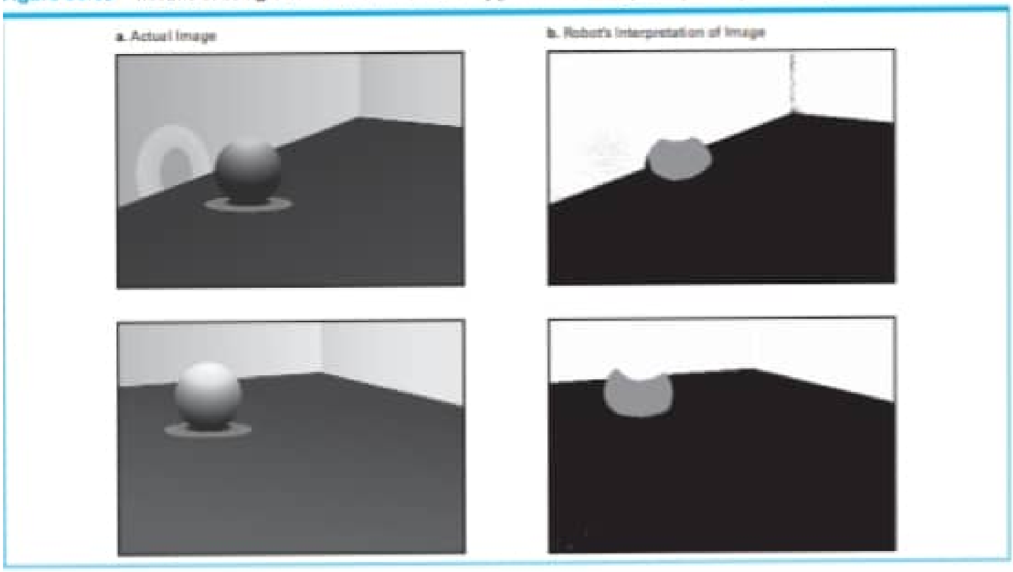
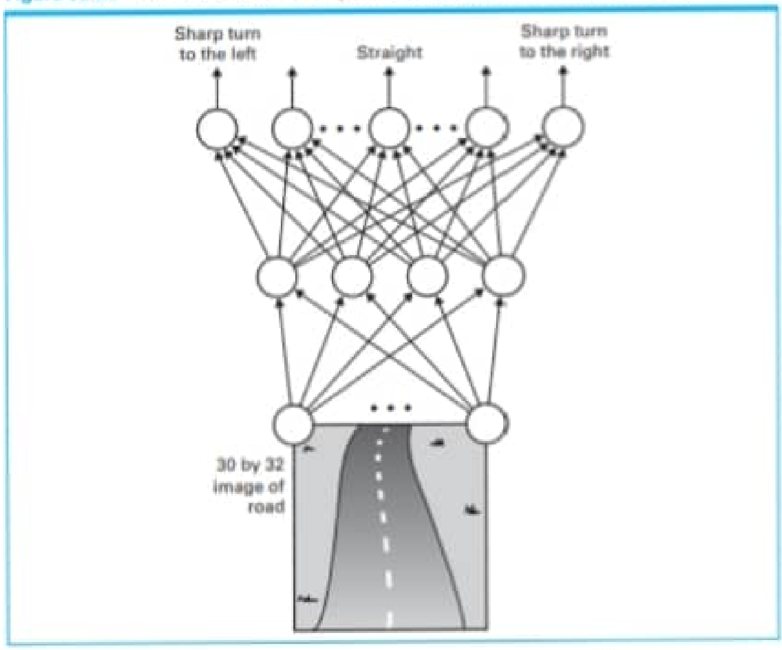


Figure 11.20 The structure of ALVINN (Autonomous Land Vehicle in a Neural Net)



each of which observed a unique portion of the video image of the road ahead and reported its findings to each of four neurons on a hidden layer. (Thus, each of these four neurons had 960 inputs.) The output of each of these four neurons was connected to each of thirty output neurons, whose outputs indicated the direction to steer. Excited neurons at one end of the thirty neuron row indicated a sharp turn to the left, while excited neurons at the other end indicated a sharp turn to the right.

ALVINN was trained by "watching" a human drive while it made its own steering decisions, comparing its decisions to those of the human, and making slight modifications to its weights to bring its decisions closer to those of the human. There was, however, an interesting side issue. Although ALVINN learned to steer following this simple technique, ALVINN did not learn how to recover from mistakes. Thus, the data collected from the human was artificially enriched to include recovery situations as well. (One approach to this recovery training that was initially considered was to have the human swerve the vehicle so that ALVINN could watch the human recover and thus learn how to recover on its own. But unless ALVINN was disabled while the human performed the initial swerve procedure, ALVINN learned to swerve as well as to recover—an obviously undesirable trait.)