Unit -5 : Machine learning

Introduction

What is learning?

→ Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task more efficiently the next time" — Herbert Simon

→ Learning is constructing or modifying representations of what is being experienced — Ryszard Michalski

→ learning is making useful changes in our minds — Marvin Minsky

Machine learning

— ML is a sub field of AI that provides computer the ability to learn and improve its learning from experience without being written rules explicitly.

— The machine start learning with observation or data, in order to look for patterns in data and make better decisions in the future based on the examples that it has been provided.

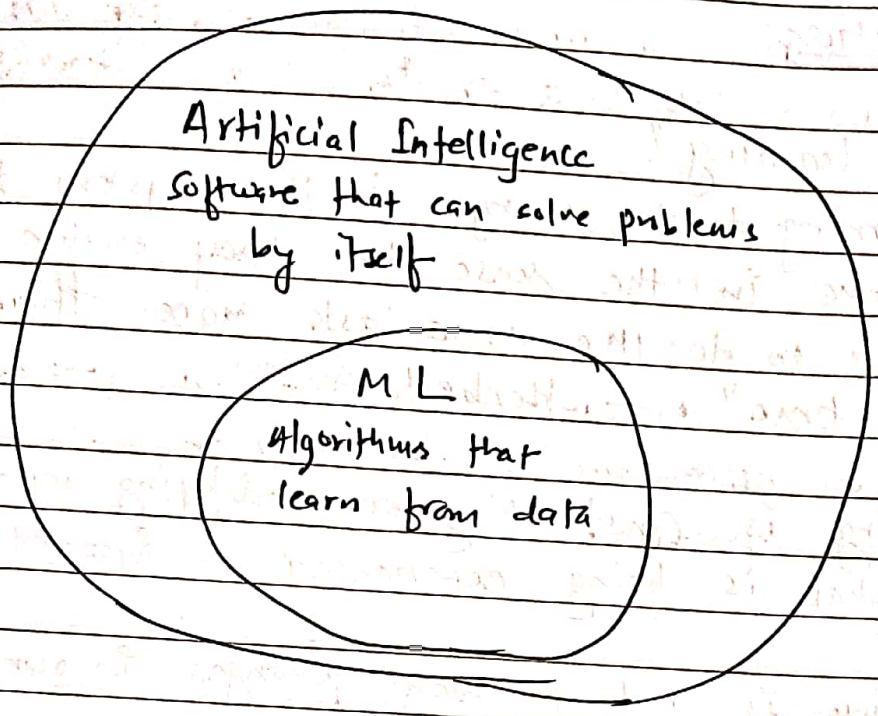
Some of the successful application of ML are:

- Computer vision processing
- Natural language processing
- forecasting (stock trends)
- pattern recognition
- Games
- Expert systems

and so on

Scanned with CamScanner

ML & AI



- AI refers to a very large field of research that encompasses a no. of technique aimed at developing computers that can learn & solve the problems

- ML is the field of AI concerned with learning from data on its own.

Definitions

- A computer program is said to learn from experience E, with respect to some class of task T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.

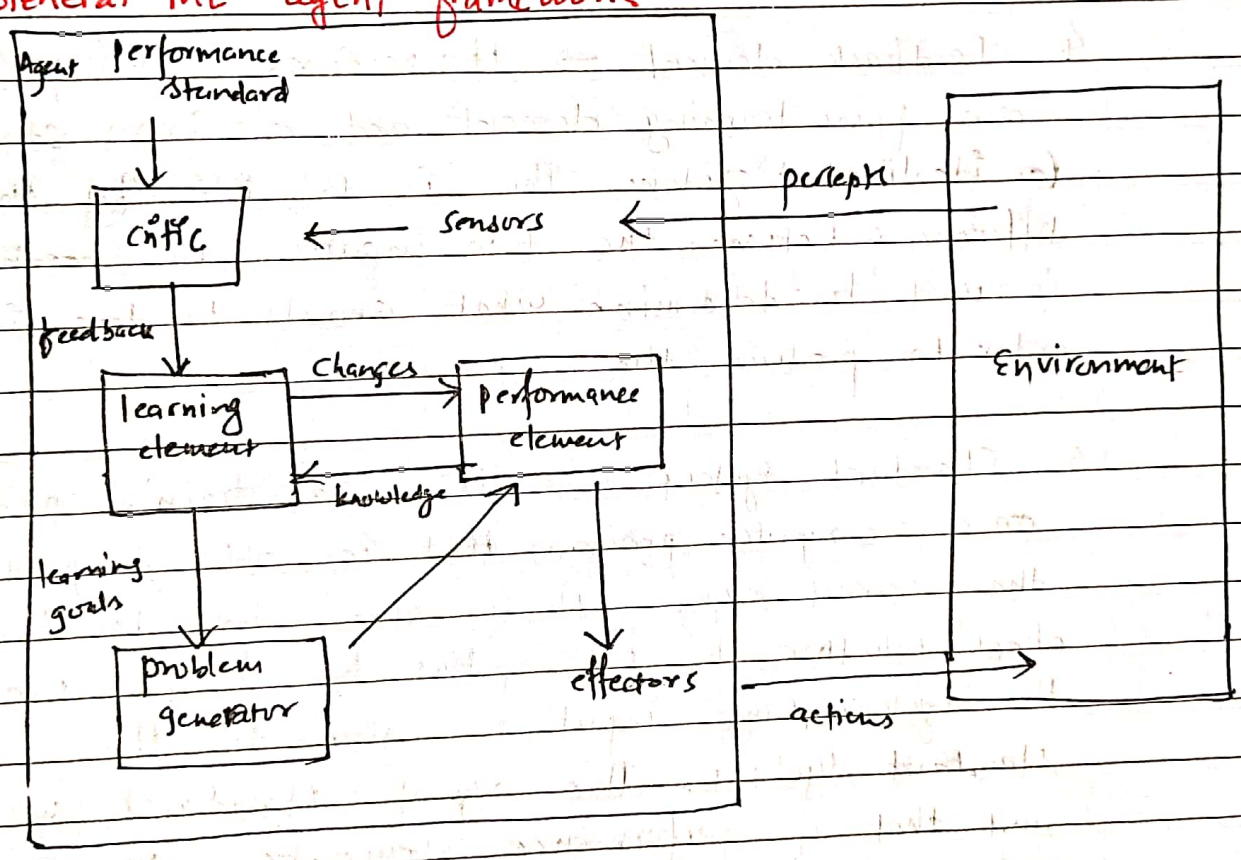
— Tom Mitchell

Scanned with CamScanner

"ML is the science of getting computers to act without being explicitly programmed, but instead letting them learn a few tricks on their own — Danko Nikolic

ML is the science of getting computers to learn and act like humans do, and improve their learning over time in autonomous fashion, by feeding those data and information in the form of observations and real world interactions.

General ML agent framework



1. Learning Element → It receives and processes the input obtained from a person, from reference material like magazines, journals, etc, or from the environment at large.

Scanned with CamScanner

2. Knowledge base → It is similar to database. Initially it may contain some basic knowledge. Thereafter it receives more knowledge which may be new and so be added as it is or it may replace the existing knowledge.

3. Performance element → It uses the updated KB to perform some tasks or solves some problems and produces corresponding o/p.

4. Feedback element → It receives two inputs, one from learning element and one from standard (or idealized) system. This is to identify the differences between the two inputs. The feedback is used to determine what should be done in order to produce the correct o/p.

5. Standard system → It is a trained person or a computer program that is able to produce the correct o/p. ~~It is also~~. In order to check whether the ML system has learned well, the ~~system~~ same input is given to the standard system. The o/p of standard system and that of performance elements are given as input to the feedback element
- Standard system is also called idealized system.

Scanned with CamScanner

Types of Machine Learning

ML can be ~~[illegible]~~ characterized in three types:

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning

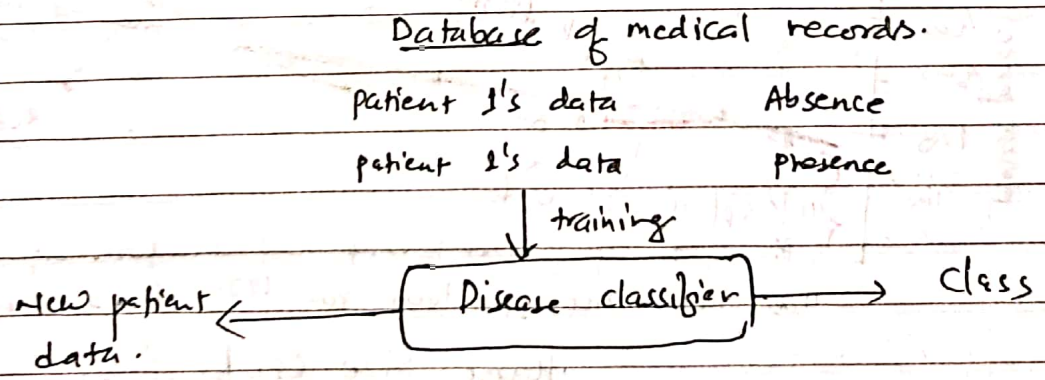
Supervised Learning

In supervised learning, the machine is presented with inputs together with the target (teacher signal) o/ps. Then, the machine ~~[illegible]~~ network tries to produce an o/p as close as possible to the target signal by adjusting the values of internal weights.

~~For example error correction method~~.

- It includes Regression & classification problems.
- Examples: KNN (K-nearest neighbour), ANN (Artificial Neural N/w), Decision Tree etc.

Medical Disease classification



Scanned with CamScanner

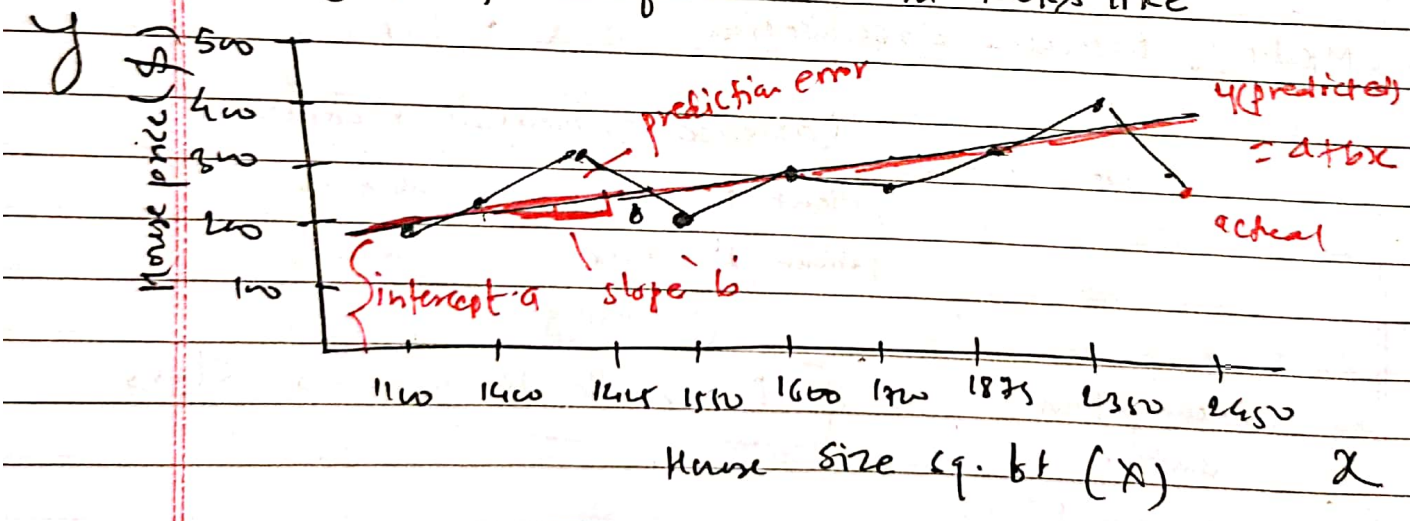
Regression : Example.

Historic data of House price

House size sq. ft (X)	House price $ (Y)
1400	245000
1600	312000
1700	279000
1875	308000
1100	199000
1550	219000
2350	405000
2450	324000
1425	319000
1700	255000

problem : for a new house, with its size (x), what will its price (y) be ?

The graph plot of above data looks like



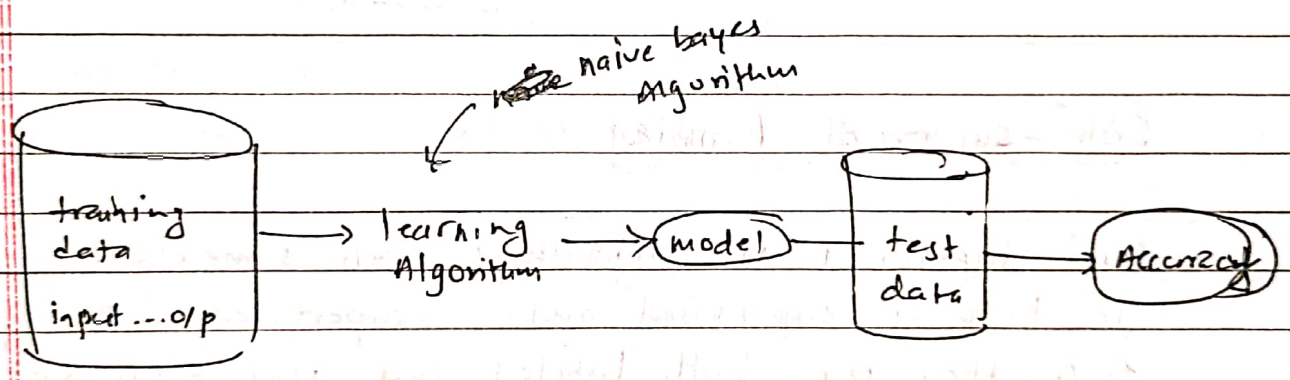
Here we try to fit straight line with the eqn.

We need to find a line with optimal values of a, b (weights) that best fits the historical data by reducing prediction error & improving prediction accuracy.

- So our objective is to find optimal a, b that minimizes the error between actual and predicted values of house price.
- We can find the formula for a & b

$$b = \frac{n\sum xy - \sum x \sum y}{n\sum x^2 - (\sum x)^2}$$

$$a = \bar{y} - b\bar{x}$$



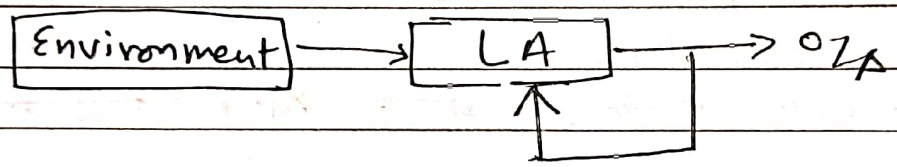
Step 1 : Training Step 2 : testing

Scanned with CamScanner

Unsupervised Learning

- unsupervised ML algorithms are used when the information used to train is neither classified nor ~~lab~~ labeled.

- The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from ~~unlabelled~~ unlabeled data.

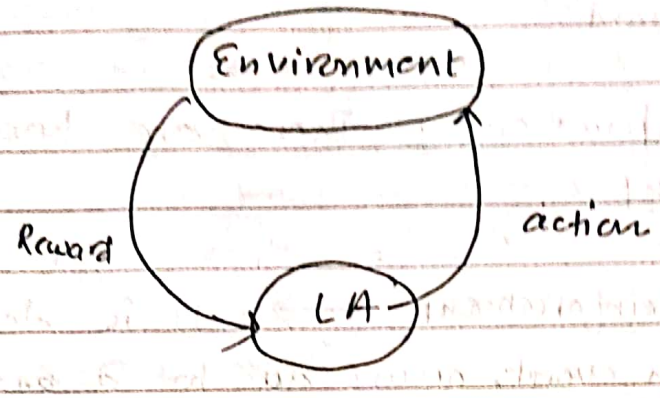


Semi-supervised Learning

Semi-supervised ML algorithms fall somewhere in between supervised and unsupervised learning since they use both labeled and unlabeled data for training - typically a small amount of labeled data and a large amount of unlabeled data.
- The system that uses this method are able to considerably improve learning accuracy.

Scanned with CamScanner

Reinforcement Learning



- It is about taking suitable action to maximize reward in a particular situation.
- The reinforcement agent decides what to do to perform the given task. In the absent of training data set, it is bound to learn from its experience.

Example:

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 situation explains the problem more easily.

Reinforcement learning consists of

Input → Initial state, from which the model will start

output → There are many possible output as there are variety of solutions 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.

Scanned with CamScanner

- The model keeps continues to learn.
- The best solution is decided based on the maximum reward.

Types of Reinforcement: There are two types of reinforcement

1. Positive Reinforcement → It is defined as when an event, occurs due to a particular behavior, increases the strength and the frequency of the behavior.
- In other words it has positive effect on the behavior.

Advantages.

→ Maximizes performance
→ Sustain change for a long period of time.

Disadvantage
→ Too much Reinforcement can lead to overload of states which can diminish the results.

2. Negative Reinforcement → It is defined as strengthening of a behavior because a negative condition is stopped or avoided.

Advantages
→ Increases behavior
→ provide defiance to minimum standard of performance.

Disadvantage
→ It only provides enough to meetup the minimum behavior.

Scanned with CamScanner

Application of Reinforcement Learning

- A variety of different problems can be solved using Reinforcement learning. Because RL agents can learn without expert supervision.
- two of the main application

① Game playing ⇒ Determining the best move

② Control problems
- elevator scheduling
- simulators

Statistical-based Learning : Naive Bayes Model

Bayes Theorem

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

where,

$P(A|B)$ ⇒ posterior probability of A conditioned on B

$P(A)$ ⇒ prior probability of hypothesis A

$P(B|A)$ ⇒ posterior probability of B conditioned on A

$P(B)$ ⇒ prior probability of event B.

Naive Bayes classifier

To classify an instance x, it calculates

$$P(c/x) = P(x/c) * P(c)/P(x)$$

$P(x/c)$ = Likelyhood

$P(c)$ & $P(x)$ = prior probability

Scanned with CamScanner

- These all probability are calculated from training samples using frequency.

- To make classification decision for event x into either c_1 or c_2 (can be extended to c_n) then,

  Calculate $P(c_1/x)$ and $P(c_2/x)$

- If $P(c_1/x) > P(c_2/x)$ then x is assigned to class c_1 otherwise assigned to c_2.

Example

Fruit	Yellow	Sweet	long	total
Orange	350	450	0	650
Banana	400	300	350	400
others	50	100	50	150
Total	800	850	400	1200

Fruit = {yellow, sweet, long}

$$P(\text{yellow}/\text{orange}) = \frac{P(\text{orange}/\text{yellow}) \cdot P(\text{yellow})}{P(\text{orange})}$$

$$= \frac{\frac{350}{800} \times \frac{800}{1200}}{\frac{650}{1200}}$$

$$= 0.5$$

Scanned with CamScanner

$$p(\text{sweet}/\text{orange}) = \frac{p(\text{orange}/\text{sweet}) \cdot p(\text{sweet})}{p(\text{orange})}$$

$$= \frac{\frac{450}{850} \times \frac{850}{1200}}{\frac{650}{1200}}$$

$$= 0.69$$

$$p(\text{long}/\text{orange}) = 0$$

$$p(\text{fruit}/\text{orange}) = 0.5 \times 0.69 \times 0 = 0$$

Similarly,

$$p(\text{yellow}/\text{banana}) = \frac{p(\text{banana}/\text{yellow}) \times p(\text{yellow})}{p(\text{banana})}$$

$$= \frac{\frac{400}{800} \times \frac{800}{1200}}{\frac{400}{1200}}$$

$$= 1$$

$$p(\text{sweet}/\text{banana}) = \frac{p(\text{banana}/\text{sweet}) \times p(\text{sweet})}{p(\text{banana})}$$

$$= \frac{\frac{300}{850} \times \frac{850}{1200}}{\frac{400}{1200}}$$

$$= 0.75$$

Scanned with CamScanner

$$p(\text{long}/\text{banana}) = \frac{p(\text{banana}/\text{long}) \times p(\text{long})}{p(\text{banana})}$$

$$= \frac{\frac{350}{400} \times \frac{400}{1200}}{\frac{400}{1200}}$$

$$= 0.875$$

$\therefore p(\text{fruit}/\text{banana}) = 1 \times 0.75 \times 0.875$
$= 0.65$

Similarly ~~for~~ for, others,

we get,

$p(\text{fruit}/\text{others}) = 0.23 \times 0.66 \times 0.33$
$= 0.072$

$\therefore$ $p(\text{fruit}/\text{banana})$ is greater among all.

$\therefore$ answer is banana //

Scanned with CamScanner

problems with Naive Bayes

Zero frequency problem → when a term is not present in the training set then its probability will be zero.

Attribute independence → Naive Bayes assume the independence between attributes or word during the calculation of conditional probability which may not be true in all cases.

Scanned with CamScanner

Artificial neural networks is non-linear, parallel, distributed, and highly connected network having capability of adaptivity, self organization, fault ~~tolerance~~ tolerance, evidential response and very large scale Integration (VLSI) implementation, which closely resembles with physical nervous system.

- Neural networks are composed of simple elements operating in parallel, inspired by biological nervous system

- We can train neural network to perform a particular functions by adjusting the values of the connection (weights) between elements.

→ ANN is an information processing paradigm that is inspired by the way biological nervous systems, such as brain process information.

→ It is composed of ~~highly~~ large no. of highly interconnected processing elements (neurons) working in union to solve specific problem.

→ An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process.

Scanned with CamScanner

Why Neural Networks?

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either human or other computer techniques.

- A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze.

other advantages:

1. Adaptive learning → An ability to learn how to do tasks based on the data given for training or initial experience.

2. Self organization → An ANN creates own organization or representation of the information it receives during learning time.

3. Real time operation → ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.

4. Fault tolerance via Redundant Information Coding → partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

Scanned with CamScanner

Biological Neuron vs Artificial Neural Network

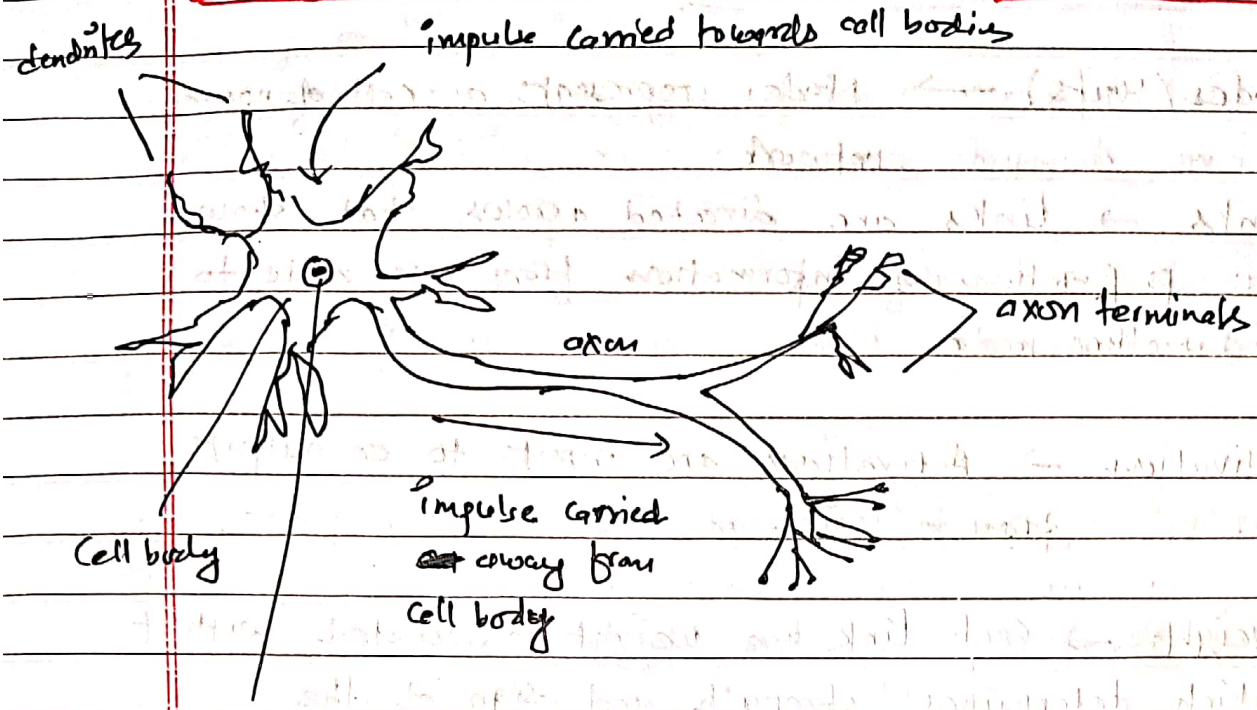
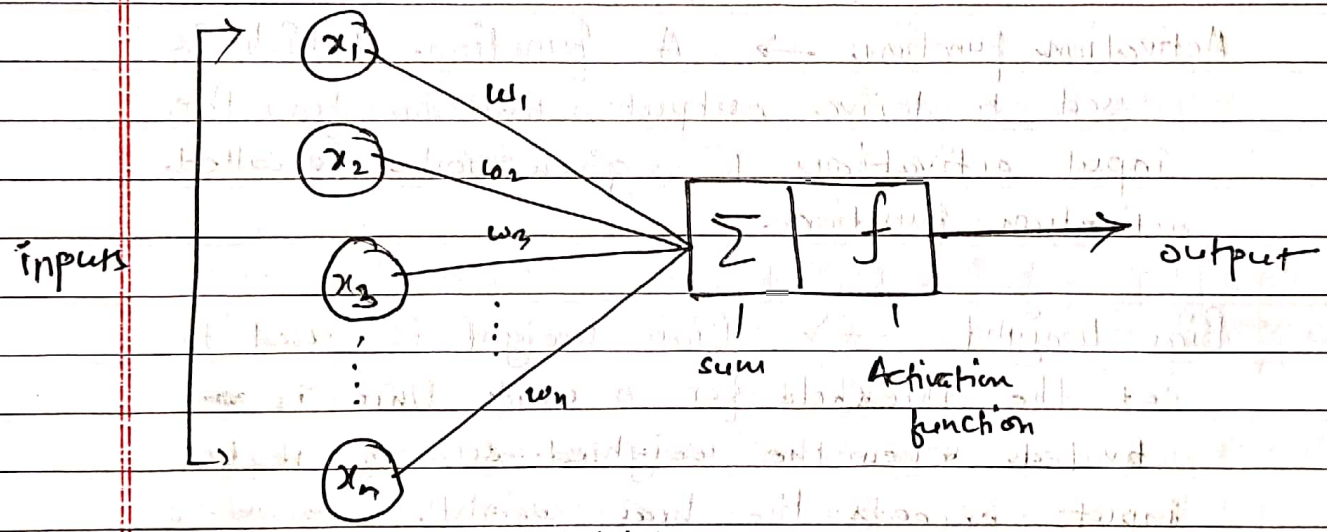


fig: Biological Neuron



Artificial neuron

Scanned with CamScanner

Units of Neural Networks

Nodes (units) → Nodes represent a cell of neural network

Links → links are directed arrows that show propagation of information from one node to another node.

Activation → Activations are inputs to or outputs from a unit.

weight → Each link has weight associated with it which determines strength and sign of the connection.

Activation function → A function which is used to derive output activation from the input activations to a given node is called activation function.

Bias weight → Bias weight is used to set the threshold for a unit. Unit is activated when the weighted sum of real inputs exceeds the bias weight.

note: page 21 figure The units output activation is

$$a_j = g\left(\sum_{i=0}^{n} w_{i,j} a_i\right)$$, where a_i is the output activation of unit i and $w_{i,j}$ is the weight on the link from unit i to this unit.

Scanned with CamScanner

DATE _____ PAGE _____

Simple Model of Neural Network

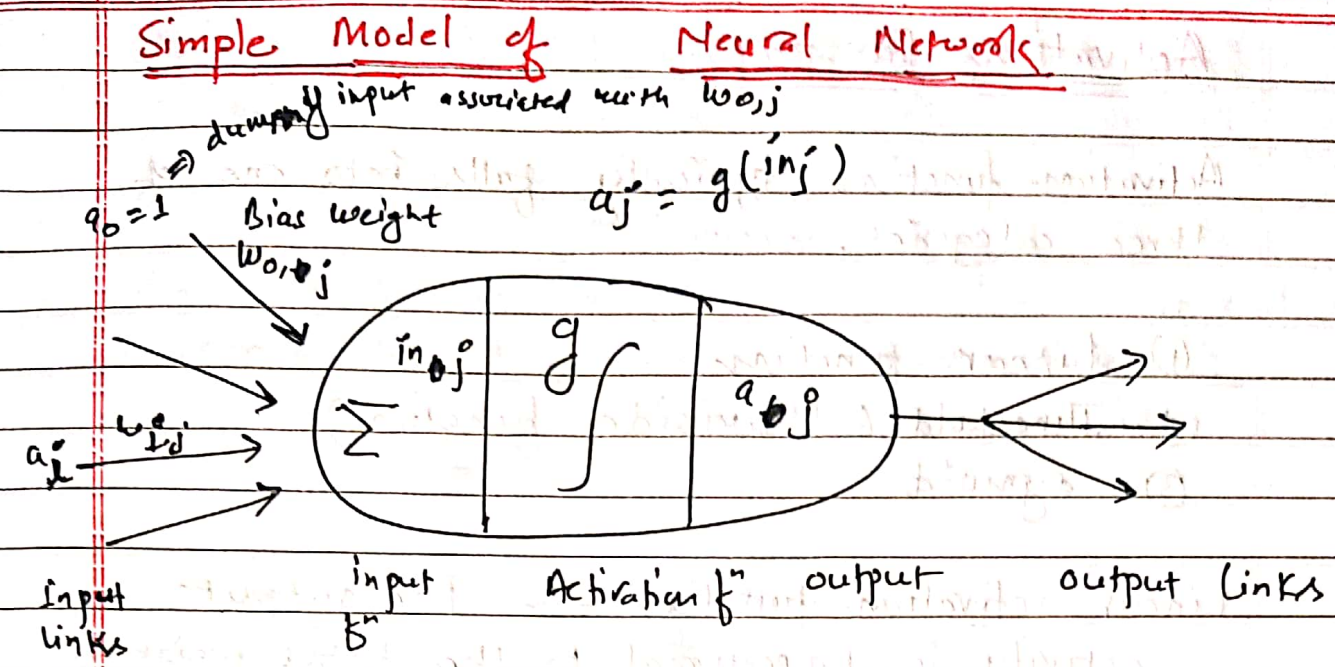


Fig: Simple model of Neural Network

A simple mathematical model of neuron is devised by McCulloch and pit is given in the figure above.

- It fires when a linear combination of its inputs exceeds some threshold.

- A neural network is composed of nodes (units) connected by directed links. A link from unit i to j serve to propagate the activation a_i from i to j. Each link has some numeric weight $w_{i,j}$ associated with it, which determines strength and sign of connection. Each unit j first computes a weighted sum of its inputs:

$$\cancel{in_j = \sum_{j=0}^{n} w_{j,i} a_j} \qquad in_j = \sum_{i=0}^{n} w_{i,j} a_i$$

~~Then it applies~~. Then it applies an activation function g to this sum to derive the output.

$$a_j = g(in_j) = g\left(\sum_{i=0}^{n} w_{i,j} a_i\right)$$

Scanned with CamScanner

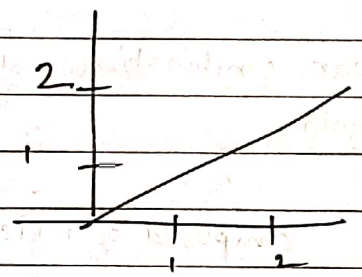
Activation Functions

Activation function typically falls into one of three categories.

1. Linear function
2. Threshold (Heaviside function)
3. sigmoid

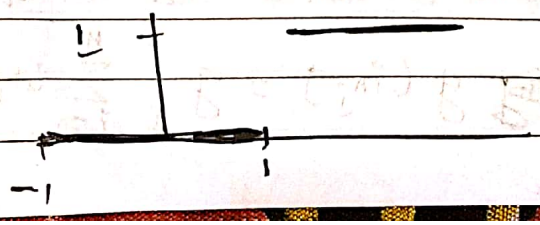
Linear activation function → The output activity is proportional to the total weighted output

$g(x) = Kx + c$, where K and x are Constants



Threeshold activation function → the output are set at one of two levels, depending on whether the total input is greater than or less than some threshold value.

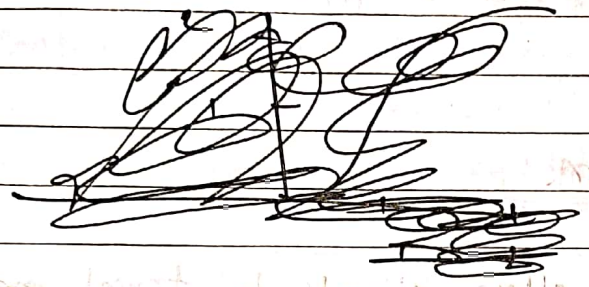
$g(x) = 1$ if $x >= k$ where, k = threshold value
$= 0$ if $x < k$



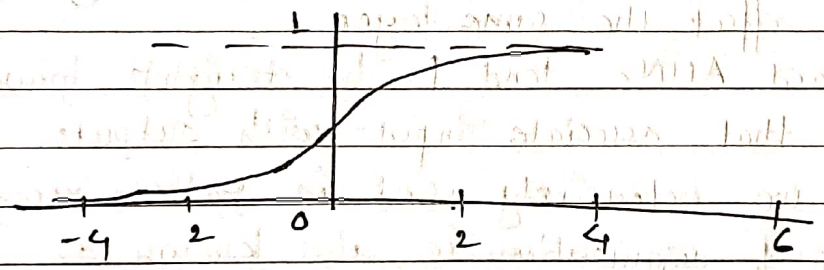
Scanned with CamScanner

Sigmoid activation function → The output varies continuously but not linearly as the input changes

- sigmoid units bear a greater resemblance to real neurons than linear or other threshold units.



$g(x) = 1/(1+e^{-x})$ [x = weighted sum]



piecewise linear function → It is a real function defined on the real numbers or a segment there of, whose graph is composed of straight-line sections.

for example:

$$g(x) = x-3 \quad \text{if } x<0$$
$$= x+1 \quad \text{if } 0<=x<3$$
$$= x+6 \quad \text{if } x>=3$$

Scanned with CamScanner

Ramp Activation function → It is defined as the positive part of its argument.

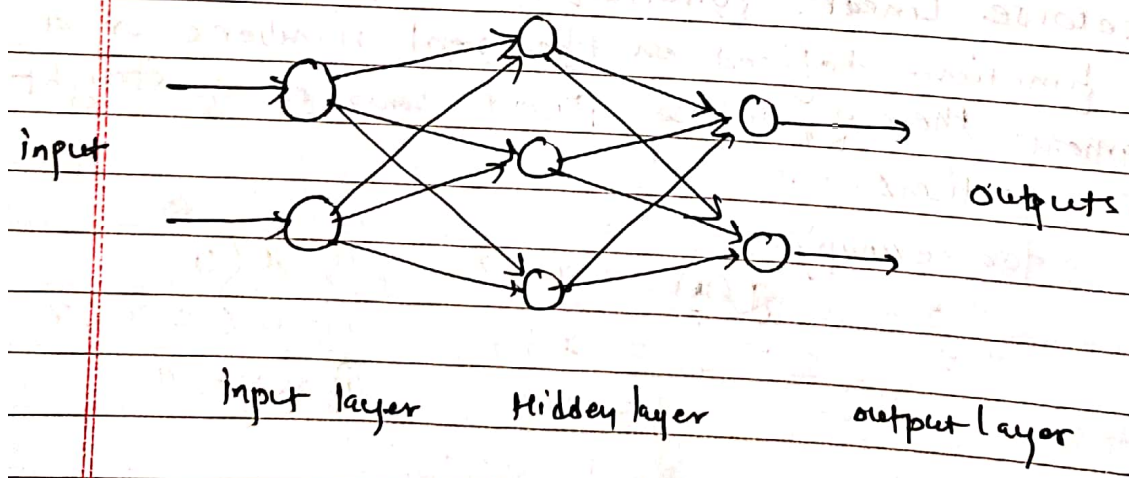
- $f(x) = \max(0, x)$

Types of ANN

Feed-forward Network

Feed-forward ANNs allow signals to travel ~~one~~ one way only; from input to output. There is no feedback (loops) i.e. the output of any layer doesnot affect the same layer.

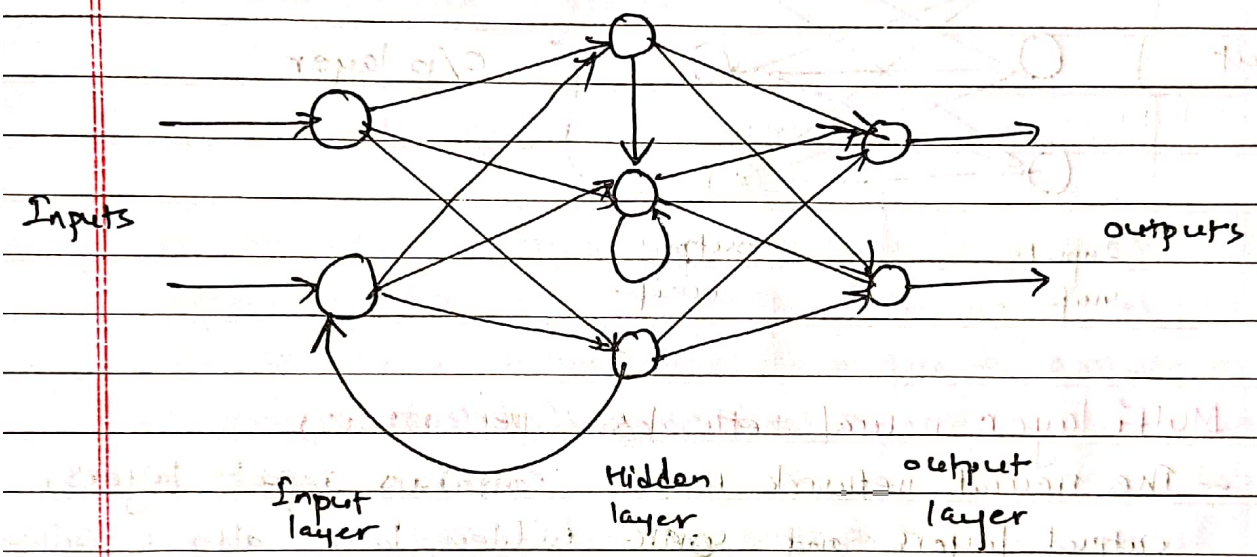
- Feed forward ANNs tend to be straight forward networks that associate input with outputs.
- They are extensively used in pattern recognition.
- This type of organization is also known as bottom-up or top-down.



Scanned with CamScanner

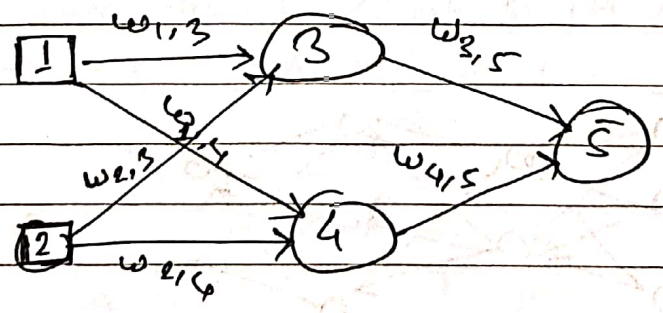
Feed-back networks (Recurrent Networks)

- Feedback networks can have signals travelling in both directions by introducing loops in network.
- Feedback networks are very powerful and can get extremely complicated.
- Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point.
- They remain at the equilibrium point until the input changes and a new equilibrium needs to be found.



Feed forward Neural Network

Example



Here, $a_5 = g(w_{3,5}\, a_3 + w_{4,5}\, a_4)$

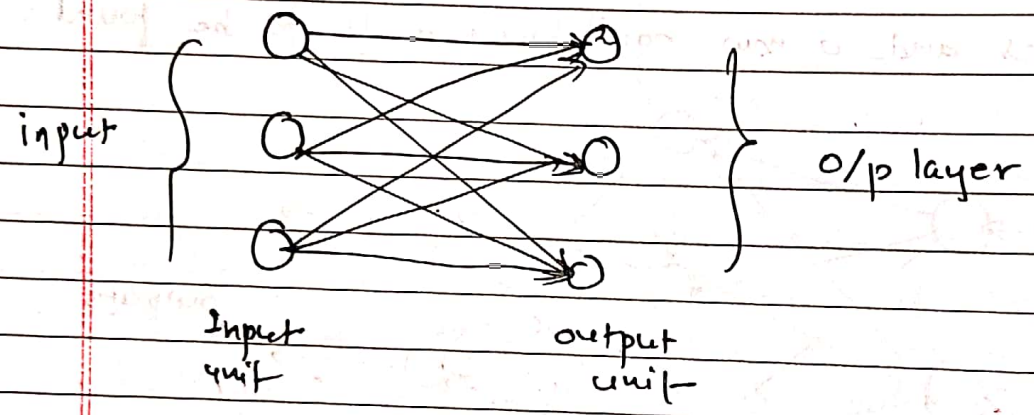
$= g(w_{3,5}\, g(w_{1,3} a_1 + w_{2,3} a_2) + w_{4,5}\, g(w_{1,4} a_1 + w_{2,4} a_2))$

Scanned with CamScanner

Types of Feedforward Neural Network.

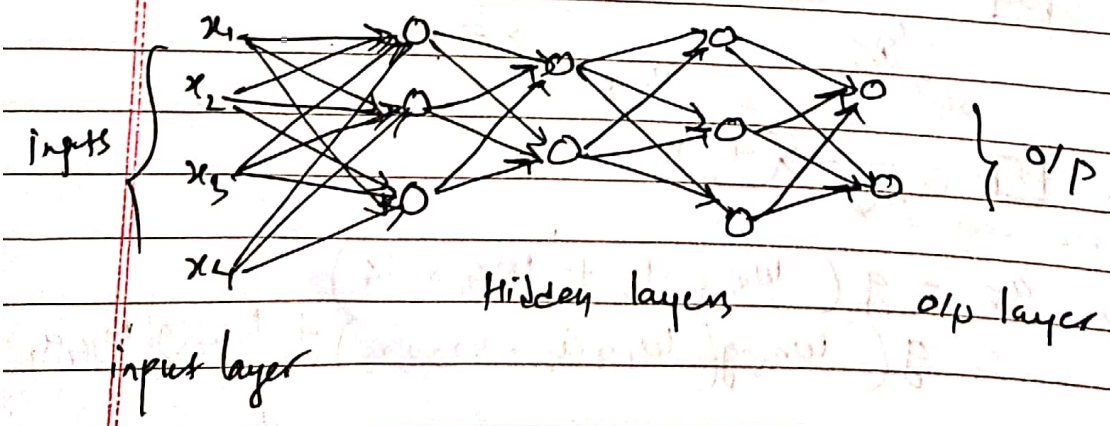
Single-layer neural networks (perceptrons)

A neural network in which all the inputs connected directly to the outputs is called a single layer neural network or perceptron network.



Multi-layer neural networks (perceptrons)

- The neural network which contains input layers, output layers and some hidden layer also is called multilayer neural network. It enlarges the space of hypothesis. Layers of the network are normally fully connected.



Scanned with CamScanner

Learning In Neural Networks:

Learning → one of the powerful features of neural networks.

Learning in Neural networks is carried out by adjusting the connection weights among neurons.

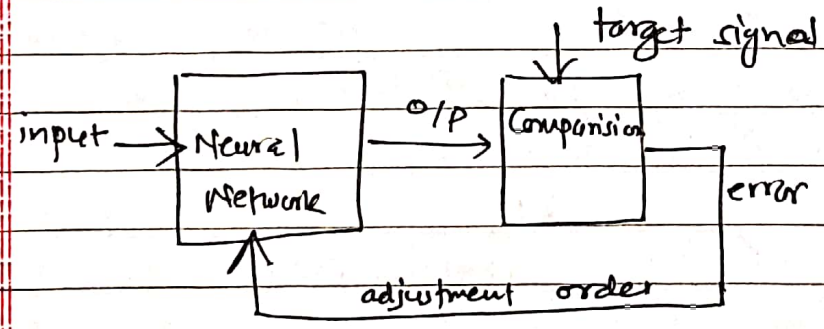
Learning may be classified into two categories:

1. Supervised learning
2. Unsupervised Learning

Supervised learning

In supervised learning, the network is presented with inputs together with the target (teacher signal) outputs. Then the neural network tries to produce an output as close as possible to the target signal by adjusting the values of internal weights. The most common supervised learning method is error correction method.

Error correction method is used for networks which their neurons have discrete output functions. Neural networks are trained with this method in order to reduce the error.

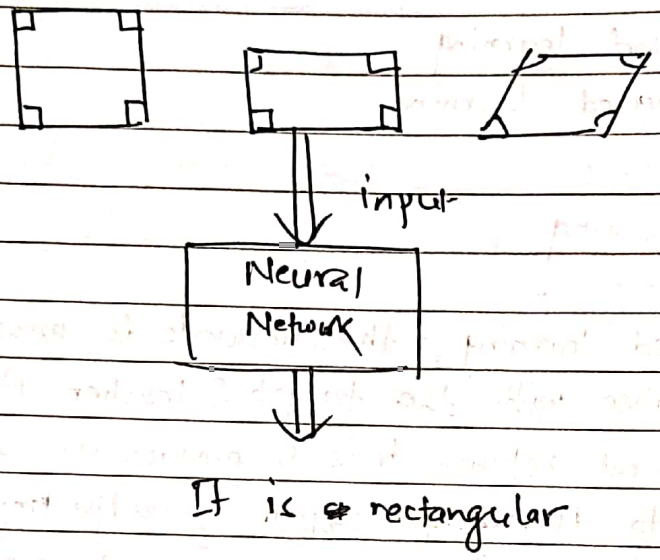


Scanned with CamScanner

Unsupervised learning

In unsupervised learning, there is no teacher (target signal) from outside and the network adjusts its weight in response to only the input patterns.

- A typical example of unsupervised learning is Hebbian learning.



Scanned with CamScanner

Hebbian Learning

→ Hebb's principle can be described as a method of determining how to alter the weights between model neurons. The weight between two neurons increases if the two neurons activate simultaneously - and reduces if they activate separately.

→ Nodes that tend to be either both positive or both negative at the same time have strong positive weights, while those that tend to be opposite have strong negative weights.

Hebb's Algorithm

Step 0 : Initialize all weights to 0

Step 1 : Given a training input, s, with its target output t, set the activations of the input units : $x_i = s_i$

Step 2 : Set the activation of the output unit to the target value $y = t$

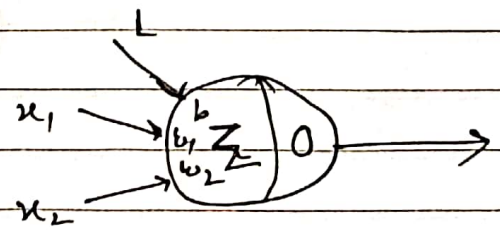
Step 3 : Adjust the weights : $w_i(new) = w_i(old) + x_i t$

Step 4 : Adjust the bias : $b(new) = b(old) + t$

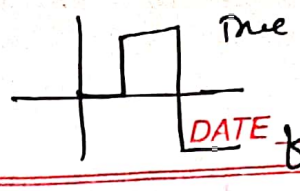
Example : Construct a Hebb Net which performs like an AND function, that is, only when both features are "active" will the data be in the target class.

Training set

x_1	x_2	bias	Target
1	1	1	1
1	-1	1	-1
-1	1	1	-1
-1	-1	1	-1



Scanned with CamScanner



Training first input

Initialize the weights to 0
present the first input:

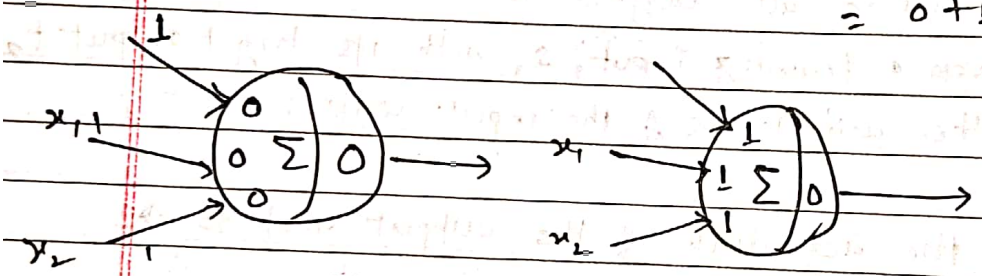
(1 1 1) with a target of 1

update the weights

$w_1(new) = w_1(old) + x_1 t$
$= 0 + 1 \cdot$
$= 1$

$w_2(new) = w_2(old) + x_2 t$
$= 0 + 1 = 1$

$b(new) = b(old) + t$
$= 0 + 1 = 1$



Training 2nd input:

present the second input.

(1, −1, 1) with a target of −1

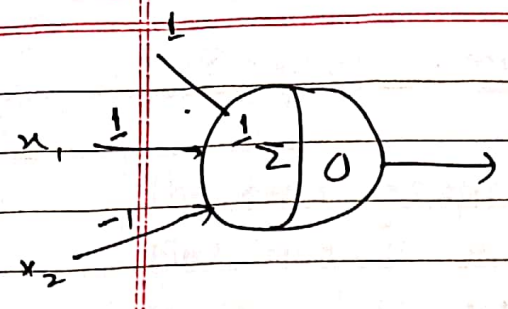
update the weight

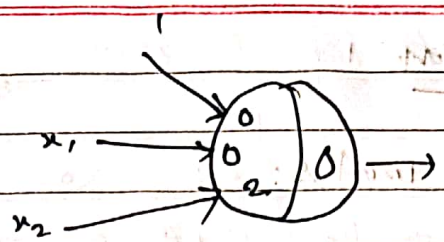
$w_1(new) = w_1(old) + x_1 t$
$= 1 + 1 * (-1) = 0$

$w_2(new) = w_2(old) + x_2 t$
$= 1 + (-1) * (-1)$
$= 2$

$b(new) = b(old) + t$
$= 1 + (-1) = 0$

Scanned with CamScanner





Training third input

present the third input:
(-1, 1, 1) with the target of
-1

update the weights

$w_1(new) = w_1(old) + x_1 t$

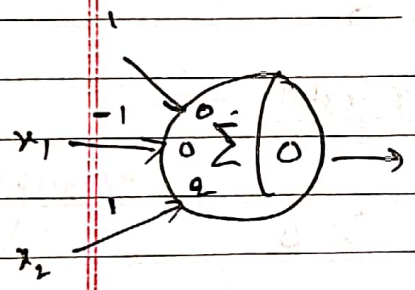
$= 0 + -1 \times -1 = 1$

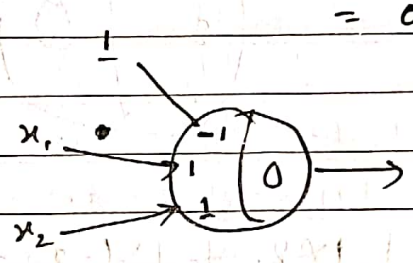
$w_2(new) = w_2(old) + x_2 t$

$= 2 + 1 \times -1 = 1$

$b(new) = b(old) + t$

$= 0 + (-1) = -1$





Training 4th input:

present the 4th input:
(-1, -1, 1) with a target of -1

update the weights

$w_1(new) = w_1(old) + x_1 t$

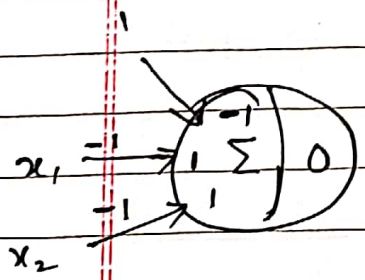
$= 1 + (-1) \times -1 = 2$

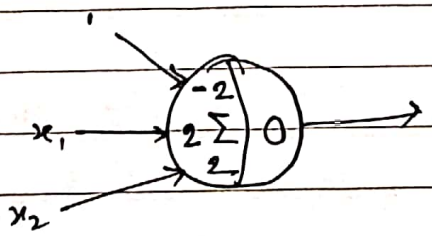
$w_2(new) = w_2(old) + x_2 t$

$= 1 + -1 \times -1 = 2$

$b(new) = b(old) + t$

$= -1 - 1 = -2$



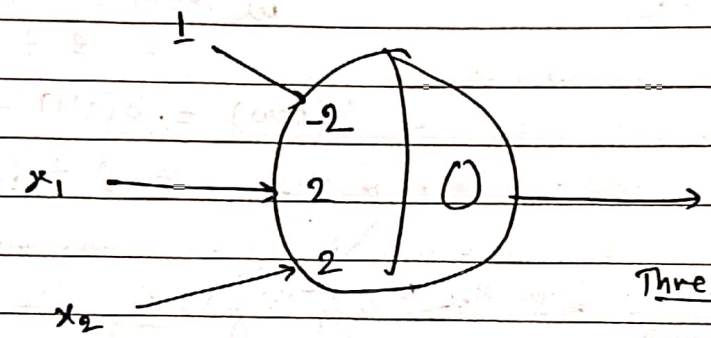


Scanned with CamScanner

Final Neuron

This neuron works:

x_1	x_2	bias	Target
1	1	1	1
1	-1	1	-1
-1	1	1	-1
1	1	1	-1



Threshold = 0

		Remarks
$1 \times 2 + 1 \times 2 + 1 \times (-2) = 2 > 0$	T	(fire)
$(-1) \times 2 + 1 \times 2 + 1 \times (-2) = -2 < 0$	F	doesn't fire
$1 \times 2 + (-1) \times 2 + 1 \times (-2) = -2 < 0$	F	"
$(-1) \times 2 + (-1) \times 2 + 1 \times (-2) = -6 < 0$	F	"

Scanned with CamScanner

Perceptron Learning theory

The term "perceptron" was coined by Frank Rosenblatt in 1962 and is used to describe the connection of simple neurons into networks.

Working Manner

1. Set the weight and thresholds of the neurons to random values.
2. present an input
3. Calculate the output of the neuron
4. Alter the weights to reinforce correct decisions and discourage wrong decisions, hence reducing the error.

So for the network to learn we shall increase the weights on the active inputs when we want the output to be active, and to decrease them when we want the o/p to be inactive.

5. Now present the next input

Perceptron Learning Algorithm

Step 1: Initialize weights and bias to 0. Also initialize learning rate α ($0 < \alpha \leq 1$). For simplicity α is set to 1.

Step 2: Calculate the net output of network

$$y_{in} = b + \sum_{i=1}^{n} x_i w_i$$

n = no. of input neurons.

Scanned with CamScanner

Step 3: Apply Activation function over the net input.

$$y = f(y_{in}) = \begin{cases} 1 & y_{in} > \theta \\ 0 & -\theta \le y_{in} \le \theta \\ -1 & y_{in} < -\theta \end{cases}$$

Step 4: Weight and bias adjustment.

- Compare the value of the actual (calculated) and desired (target) o/p.

if $y \neq t$, then

$$w_i(new) = w_i(old) + \alpha t x_i$$

$$b(new) = b(old) + \alpha t$$

else, we have,

$$w_i(new) = w_i(old)$$

$$b(new) = b(old)$$

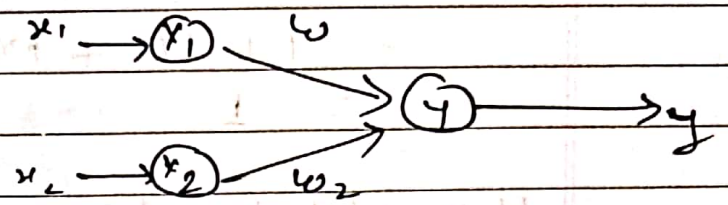
Scanned with CamScanner

Example

Implement AND function using perceptron networks for bipolar inputs and targets.

Soln:

x_1	x_2	t
1	1	1
1	-1	-1
-1	1	-1
-1	-1	-1



Initialize, $w_1 = w_2 = b = 0$ & $\theta = 0$, $\alpha = 1$

For 1st input

$x_1 = 1$, $x_2 = 1$, $t = 1$

Net input $y_{in} = b + w_1 x_1 + w_2 x_2$

$= 0 + 0 + 0 = 0$

o/p y is computed by applying activation over the net input calculated.

$$y = f(y_{in}) = \begin{cases} 1 & y_{in} > 0 \\ 0 & y_{in} = 0 \\ -1 & y_{in} < 0 \end{cases}$$

$f(0) = 0 \neq t$, Hence update weights.

$w_1(\text{new}) = w_1(\text{old}) + \alpha t x_1$

$= 0 + 1 \times 1 \times 1 = 1$

$w_2(\text{new}) = w_2(\text{old}) + \alpha t x_2$

$= 0 + 1 \times 1 \times 1 = 1$

$b(\text{new}) = b(\text{old}) + \alpha t = 0 + 1 \times 1 = 1$

Scanned with CamScanner

For 2nd input

$x_1 = 1,\ x_2 = -1,\ t = 1$

$y_{in} = b + w_1 x_1 + w_2 x_2$
$= 1 + 1 \times 1 + 1 \times (-1)$
$= 1$

$y = f(y_{in})$; $f(1) = 1 \neq t$, Hence update the weight

$w_1(new) = w_1(old) + \alpha t x_1$
$= 1 + 1 \times (-1)(1) = 0$

$w_2(new) = w_2(old) + \alpha t x_2$
$= 1 + 1 \times (-1)(-1) = 2$

$b(new) = b(old) + \alpha t$
$= 1 + 1 \times (-1) = 0$

For III input

$x_1 = -1,\ x_2 = 1,\ t = -1$

$y_{in} = b + x_1 w_1 + x_2 w_2$
$= 0 + (-1) \times 0 + 1 \times 2$
$= 2$

$y = f(y_{in}) = f(2) = 1 \neq t$
Hence update weights

Scanned with CamScanner

$w_1(new) = w_1(old) + \alpha t x_1$

$= 0 + 1 \times (-1) \times (-1) = 1$

$w_2(new) = w_2(old) + \alpha t x_2$

$= 2 + 1 \times -1 \times 1 = 1$

$b(new) = b(old) + \alpha t$

$= 0 + 1 \times (-1) = -1$

For 4th input,

$x_1 = -1, \; x_2 = -1, \; t = 1$

$y_{in} = b + w_1 x_1 + w_2 x_2$

$= -1 + 1 \times (-1) + 1 \times (-1) = -3$

$y = f(y_{in}) = f(-3) = -1 = t$. Hence don't need to update weights.

$\therefore$ Epoch - 1 computed

for epoch-2 repeat the process.

For first input,

$x_1 = 1, \; x_2 = 1, \; t = 1$

$y_{in} = b + w_1 x_1 + w_2 x_2$

$= -1 + 1 \times 1 + 1 \times 1$

$= 1$

$y = f(y_{in}) = f(1) = 1$

Scanned with CamScanner

for 2nd input,

$x_1 = 1, x_2 = -1, t = -1$

$y_{in} = b + w_1 x_1 + w_2 x_2$

$= -1 + 1 \times 1 + 1 \times -1$

$= -1$

$y = f(y_{in}) = f(-1) = -1$

for 3rd input,

$x_1 = -1, x_2 = 1, t = -1$

$y_{in} = b + w_1 x_1 + w_2 x_2$

$= -1 + 1 \times (-1) + 1 \times 1$

$= -1$

$y = f(y_{in}) = f(-1) = -1$

for 4th input,

$x_1 = -1, x_2 = -1, t = -1$

$y_{in} = b + w_1 x_1 + w_2 x_2$

$= -1 + 1 \times (-1) + 1 \times (-1)$

$= -3$

$y = f(y_{in}) = f(-3) = -1$

$\therefore$ Target achieved, the network is trained

Scanned with CamScanner

Delta Rule

→ The delta rule is a gradient descent learning rule for updating the weights of the artificial neurons in a single layer perceptron.

→ It is the special case of the more general back-propagation algorithm.

For a neuron j with activation function $g(x)$ the delta rule for j's ith weight w_{ji} is given by

$$\Delta w_{ji} = \alpha (t_j - y_j) g'(h_j) x_i$$

Where,

α = learning rate

$g(x)$ = neuron's activation function

t_j = target output

h_j = weighted sum of the neuron's input

y_j = actual output

x_i = ith input

It holds,

$$h_j = \sum x_i w_{ji} \text{ and}$$

$$y_i = g(h_j)$$

The delta rule is commonly stated in simplified form for a perceptron with a linear activation function is

$$\nabla w_{ji} = \alpha (t_j - y_j) x_i$$

Scanned with CamScanner

Backpropagation

→ It is a supervised learning method, and is an implementation of the Delta rule.
→ It requires a teacher that knows, or can calculate, the desired output for any given input.
→ It is most useful for feed-forward networks.
→ The term is abbreviation for "Backpropagation of errors".

→ As the algorithm's name implies, the errors propagate backwards from the o/p nodes to the inner nodes. So technically speaking, backpropagation is used to calculate the gradient of the error of the network w.r.t. the network's modifiable weights.

Summary of Backpropagation technique

1. present a training sample to the neural n/w.
2. Compare the n/w's o/p to the desired o/p from that sample. Calculate the error in each o/p neuron.

3. For each neuron, calculate what the o/p should have been, and a scaling factor, how much lower or higher the o/p must be adjusted to match the desired output. This is the local error.
4. Adjust the weights of each neuron to lower the local error.

Scanned with CamScanner

5. Assign "blame" for the local error to neurons at the previous level, giving greater responsibility to neurons connected by stronger weights.

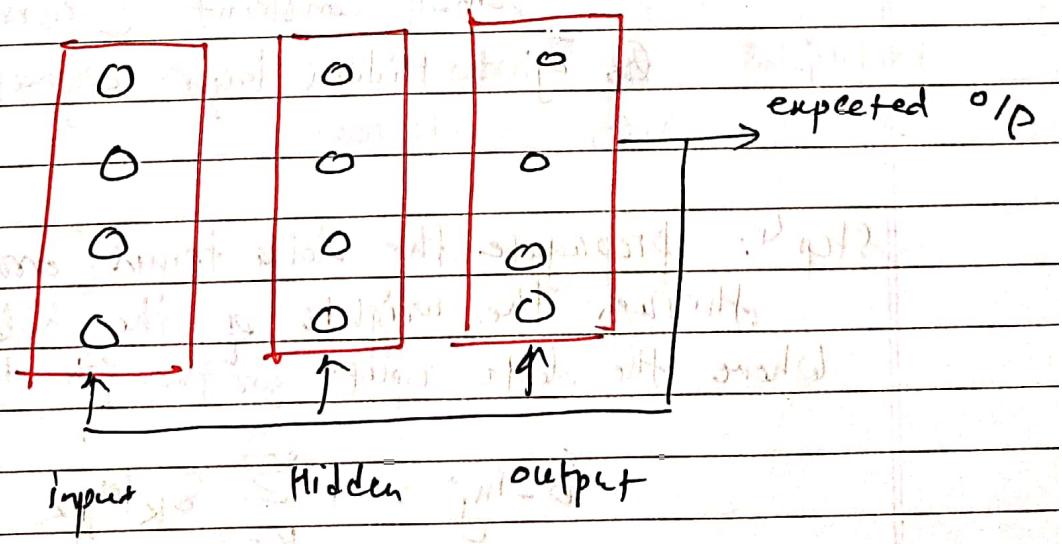
6. Repeat from step 3 on the neurons at the previous level, using each one's "blame" as its error.

Characteristics

* A multi-layered perceptron has three distinctive characteristics.
  - → The n/w contains one or more layers of hidden neurons
  - → The n/w exhibits a high degree of connectivity.
  - → Each neuron has a smooth (differentiable everywhere) nonlinear activation function, the most common is the sigmoidal nonlinearity

$$y_i = \frac{1}{1+e^{s_i}}$$

* The backpropagation algorithm provides a computational efficient method for training multi-layer networks.



Scanned with CamScanner

Algorithm

Step 0: Initialize the weights to small random values.

Step 1: Feed the training sample through the n/w and determine the final o/p.

Step 2: Compute the error for each o/p unit, for unit k it is:

$$\delta_k = (t_k - y_k)\, f'(y_in_k)$$

where,

t_k = Required o/p

y_k = actual o/p

f' = derivative of f

Step 3: Calculate the weight correction term for each o/p unit, for unit k it is:

$$\Delta w_{jk} = \alpha\, \delta_k\, Z_j$$

where,

α = small constant (learning rate

Z_j = Hidden layer signal

Step 4: propagate the delta terms (errors) back through the weights of the hidden units where the delta input for the jth hidden unit is:

$$\delta_in_j = \sum_{k=1}^{n} \delta_k w_{jk}$$

Scanned with CamScanner

Step 5: Calculate the weight correction term for the hidden units:

$$\Delta w_{ij} = \alpha \delta_j x_i$$

Step 6: update the weights:

$$w_{jk}(\text{new}) = w_{jk}(\text{old}) + \Delta w_{jk}$$

Step 7: Test for stopping (max^m cycles, small changes etc)

Note: There are no. of options in the design of a back-propagation system,

(1) Initial weights — best to set inside a small range (say −0.5 to 0.5)

(2) No of cycles → tend to be quite large for backpropagation systems.

(3) No. of neurons in the hidden layer — as few as possible.

Scanned with CamScanner

Extra.

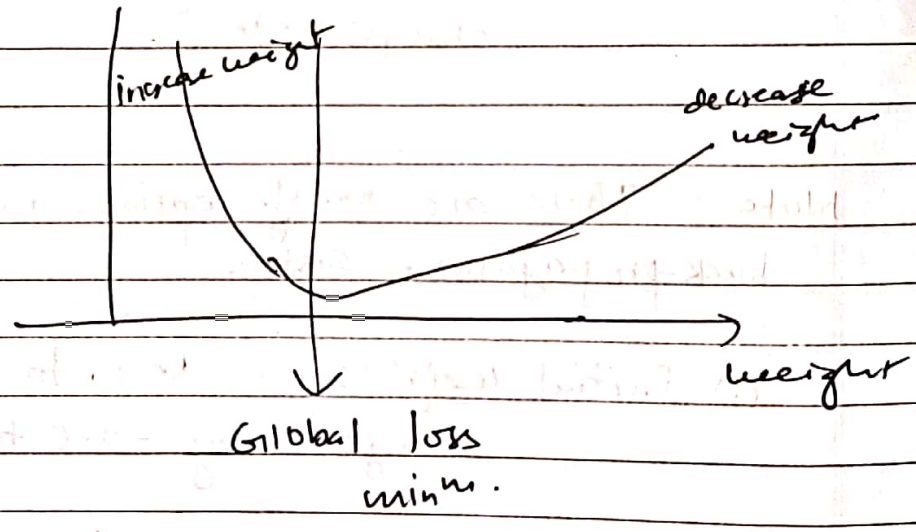
Concept – Backpropagation:

- Supervised learning algorithm used to train neural networks.

Gradient Descent

- Gradient Descent is used ~~to~~ for finding the minimum of a function.

- In Backpropagation we minimize the error function



- performing gradient descent means differentiating the fn.
- The slope will give direction for minm error.

Scanned with CamScanner

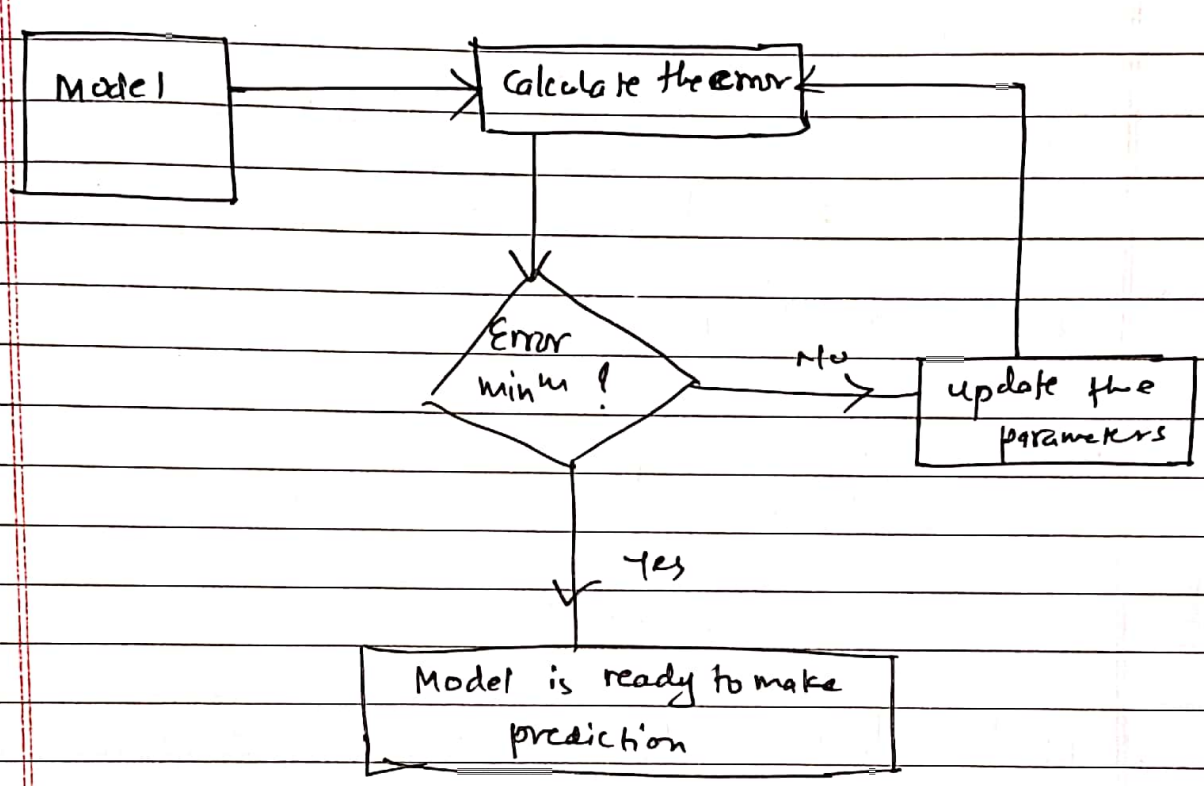


Fig: training process

Scanned with CamScanner

Application of AI

Expert system

An Expert system is a set of program that manipulates encoded knowledge to solve problem in a specialized domain that normally requires human expertise.

It is defined as an automated reasoning that attempts to mimic the performance of human expert.

Characteristics / Features

- Should be able to display the intelligent behavior.
- Should be able to explain the reasoning
- Should be able to draw conclusions from the relationships that are very complex in the nature.
- Should be able to provide much needed portable knowledge.
- Should be able to deal with the certainty.
- Should be easily modified [program's knowledge]

Scanned with CamScanner

Human System	vs Expert System
- Human expert is an individual who has a superior capability understanding of a problem	- Expert system is a computer system which emulates the ability of human expert.
- They are creative - They are adaptive - They posses sensory experie	- The are uninspired. - they need to be told.
- skill and knowledge deteriorate over time. - performance varies with time	- provides permanent expertise & knowledge. - always consistent
- They can deal with common sense	- They need technical knowledge
- Not replacable - susceptible to emotional & psychological factors that can impair decision making.	- replaceable - Not susceptible to emotional factor & provide proper reproducible results always.

Scanned with CamScanner

Block Diagram of Expert system

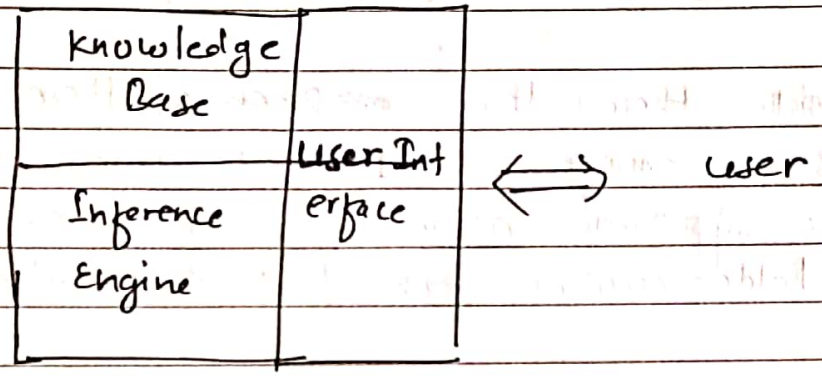


Fig : Block diagram of Expert system

There is currently no such thing as "standard" expert system. Because a variety of techniques are used to create expert systems, they differ as widely as the programmers develop them & the problems they are designed to solve.

- However principle components of most common expert systems are :

1. Knowledge Base → Knowledge Base is the system's knowledge. KB contains both declarative (facts about objects, events & situations) and procedural knowledge (information about course of action).

2. Inference Engine → The control structure or rule interpreter which directs how to use knowledge is called inference engine.

Scanned with CamScanner

3. User Interface → The component of an expert system that communicates with the user is known as user interface.

Note: Beside these three components, there is a working memory - a data structure which stores information about a specific run. It holds current ~~fact~~ fact & knowledge.

Stages of Expert System Development

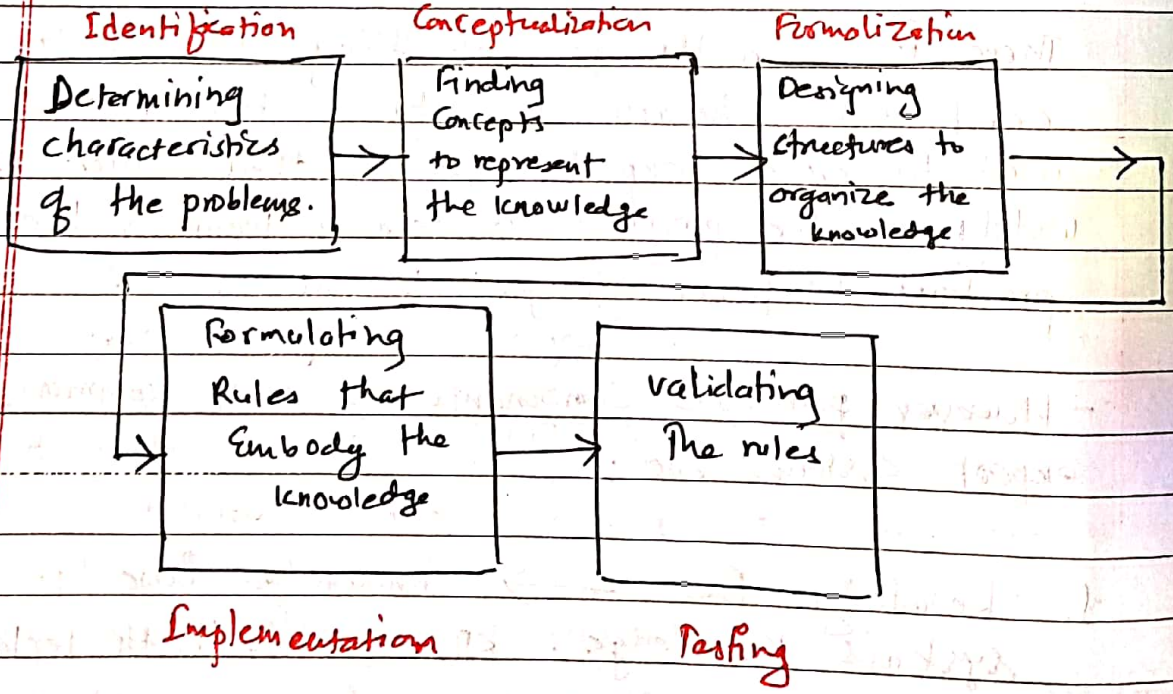


Fig: Different phases of expert System Development.

1. Identification → It is not enough simply to feel that the system would be helpful in certain situation; we must determine the exact nature of the problem and state the precise goals that indicate exactly how we expect the expert system to contribute to the solution.

Scanned with CamScanner

2. Conceptualization → This stage involves analysing the problem further to ensure that its specifics, as well as its generalities are understood.
In this stage the knowledge engineer frequently creates a diagram of the problem to depict graphically the relationships between the objects and processes in the problem domain.
→ It is often helpful at this stage to divide the problem into a series of subproblems and to diagram both the relationships among the pieces of each sub-problem and the relationship among the various sub-problems.

3. Formalization → In this stage, problem is connected to its proposed solution, an expert system, by analyzing the relationships depicted in the conceptualization stage.

During formalization, it is important that the knowledge engineer be familiar with the following:

① The various techniques of KR and heuristic search used in expert systems.
② The expert system "tools" that can greatly expedite the development process. and
③ Other expert systems that may solve similar problems and thus may be adequate to the problem at hand.

4. Implementation → In this stage, the formalized concepts are programmed on to the computer that

Scanned with CamScanner

has been chosen for system development, using the predetermined techniques & tools to implement a "first pass" prototype of the expert system.

- Many scientists actually consider the first prototype to be a "throw-away" system, useful for evaluating progress but hardly usable expert system.

5. Testing → Testing provides opportunities to identify the weakness in the structure and implementation of the system and to make appropriate correction.

Scanned with CamScanner

Natural language processing

DATE _____ PAGE ____

perception and communication are essential components of intelligent behavior. They provide the ability to effectively interact with our environment.

Humans perceive and communicate through their five basic senses of sight, hearing, touch, smell and taste, and their ability to generate meaningful utterances.

- Developing program that understands natural language is a difficult problem. Natural language are large. They contain infinity of different sentences. No matter how many sentences a person has heard or seen, new ones can always be produced.

Also, there is much ambiguity in a natural language. Many words have several meanings and sentences can have different meaning in different contexts. This makes the creation of programs that understands a natural language, one of the most challenging tasks in AI.

Some better known applications of NLP include:

- Voice recognition software → which translates speech into input for word processors or other applications.

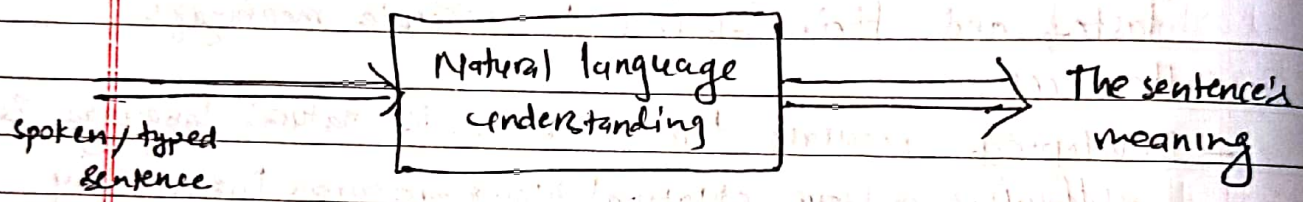
- Text-to-speech synthesizers → which read text aloud for users such as the hearing impaired.

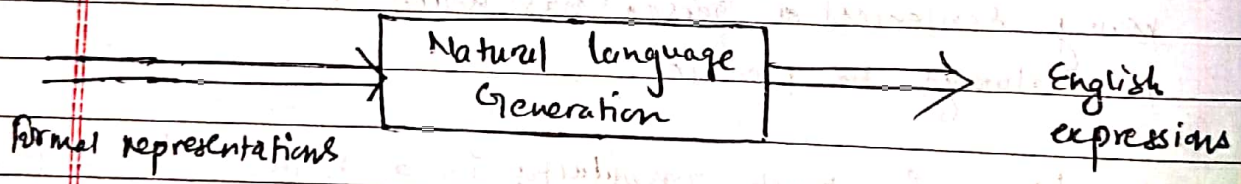
- Grammer and style checkers → which analyze text in an attempt to highlight errors of grammar or usage.

Scanned with CamScanner

Machine translation systems → which automatically renders a document such as a web page in another language.

Computers using natural language as input and/or output





Natural Language Generation:

NLG is the natural language processing task of generating natural language from a machine representation system such as a knowledge base or a logical form.

In a sense, NLG system is like a translator that converts a computer based representation into a natural language representation.

Scanned with CamScanner

The different types of generation techniques can be classified into four main ~~categori~~ categories.

1. Canned (preserved, predetermined) text systems → consitute the simplest approach for single-sentence and multi-sentence text generation. They are trivial to create, but very inflexible.

2. Template Systems → The next level of sophistication rely on the application of pre-defined templates or schemas and are able to support flexible ~~alterat~~ alterations. The template approach is used mainly for multi-sentence generation, particularly in applications whose texts are fairly regular in structure.

3. Phrase-based systems → Employ what can be seen as generalized templates. In such systems a phrasal pattern is first selected to match the top level of the input, and then each part of the pattern is recursively expanded into a more specific phrasal pattern that matches some subportion of the input. At the sentence level, the phrases resemble phrase structure grammar rules and at the discourse level they play the role of text plans.

4. Feature-based systems → which are yet ~~restricted~~ restricted to single-sentence generation, represent each possible minimal alternative of expression by a single feature. Accordingly, each sentence is specified by a unique set of features. In this

Scanned with CamScanner

framework, generation consists in the incremental collection of features appropriate for each portion of the input.
Feature collection itself can either be based on unification or on the traversal of a features selection network
The excessive power of the approach is very high since any distinction in language can be added to the system as a feature.

Many natural language generation systems follow a hybrid approach by combining components that utilize different techniques.

Scanned with CamScanner

Natural language Understanding

DATE _____ PAGE ____

- Developing programs that understand a natural language is a difficult problem. Natural language are large. They contain infinity of ~~sentences~~ different sentences. Many words have several meaning such as can, fly, bear, bank etc, and have different meaning in different contents.

Example: A can of juice I can do it.

This makes the creation of programs that understand a natural language, one of the most challenging tasks in AI.

Three major issues involved in understanding language.

→ A large amount of human knowledge is assumed.
→ language is pattern based, phonemes are components of the words and words make phrases and sentences. phonemes, words and sentences order are not random.
→ language acts are the product of agents (human or m/c)

Levels of Knowledge used in Language Understanding

A language understanding knowledge must have considerable knowledge about the structures of the language including what the words are and how they combine into phrases and sentences. It must also know the meanings of the words and how they contribute to the meanings of the sentence and to

Scanned with CamScanner

the context within which they are being used. The component forms of knowledge needed for an understanding of natural languages are sometimes classified according to the following levels.

1. phonological → Relates sound to the words we recognize.

2. Morphological → This is lexical knowledge which relates to the word construction from basic units called morphemes.
   - A morpheme is the smallest unit of meaning
   eg. friend + ly = friendly

3. Syntactic → This knowledge relates to how words are put together or structured together to form grammatically correct sentences in the language.

4. Semantic → This knowledge is concerned with the meanings of words and phrases and how they combine to form sentence meaning.

5. pragmatic → Relates to use of sentences in different contexts and how the context affects the meaning of the sentence.

6. World → Includes the knowledge of the physical world, the world of human social interaction, and the roles of goals and intensions in communication.

Scanned with CamScanner

Challenges in NLP

NLP is a powerful tool with huge benifits, but there are still a no. of NLP limitations and problems.

1. Contextual words and phrases and homonyms.
2. Synonyms
3. Irony and sarcasm
4. Ambiguity
5. Errors in text or speech
6. Colloquialisms and slangs
7. Domain specific language.
8. low resource languages
9. lack of ~~search~~ research & development.

The most widely used classification is probably the one which divides ambiguity into

1. Lexical ambiguity → word may have more than one interpretation
2. Semantic ambiguity → meaning
3. Syntactic ambiguity → Grammatically structured.
4. Pragmatic ambiguity → context of the phrase results in alternative interpretations of that phrase.

Scanned with CamScanner

Lexeme (Lexicon) & word forms

A lexicon defines the words of a language that a system knows about.
This includes common words and words that are specific to the domain of the application.

- dog and dogs are different forms of the same lexemes.
- Dog and dog catcher are different lexeme

Morphology

- Morphology is the identification, analysis and description of the structure of words.

- Morphological analysis is the process of recognizing the suffixes and prefixes that have been attached to a word.

- We do this by having a table of affixes and trying to match the inputs as:

prefixes + root + suffixes

adjective + ly → adverb
friend + ly → friendly

Scanned with CamScanner

Syntactic Analysis

Syntactic analysis takes an input sentence and produces a representation of its grammatical structure. A grammer describes the valid parts of speech of a language and how to combine them into phrases.

A computer grammer specifies which sentences are in a language and their parse trees.
A parse tree is a hierarchical structure that shows how the grammar applies to the input.
Each level of the tree corresponds to the application of one grammer rule.

It is the starting point for working out the meaning of the whole sentence.

Consider : The dog ate the bone
The bone was eaten by the dog

Understanding the structure (via syntax rule) of the sentences ~~they~~ help ~~us~~ work out that it's the bone that gets eaten not the dog.

A sentence is syntactically ambiguous if there are two or more possible grouping.

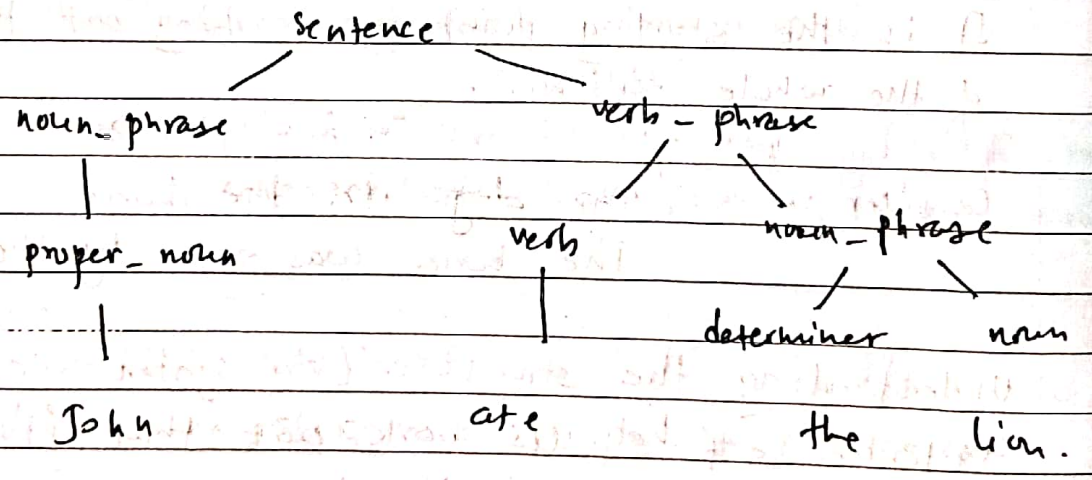
- Syntactic analysis helps determining the meaning of a sentence by working out possible word structure.
- Rules of syntax are specified by writing a grammar for language.
- A parser will check if a sentence is correct according

Scanned with CamScanner

to the grammar. It returns ~~rep~~ a representation (parse tree) of the sentence's structure.

A grammar specifies allowable sentence structures in terms of basic ~~catego~~ categories such as ~~nou~~ noun and verbs.
A given grammar, however, is unlikely to cover all possible grammatical sentences.
Parsing sentence is to help determining their meanings, not just to check that they are ~~cond~~ correct.

A parse trees illustrates the syntactic structure of the sentence.



Semantic Analysis

Semantic analysis is a process of converting the syntactic representation into a meaning representation. This involves following tasks:

- Word sense determination
- Sentence level analysis
- Knowledge representation

Word Sense

Mary had a bat in her office.
bat = a baseball thing
bat = a flying mammal

Sentence level analysis

- once the words are understood, the sentence must be assigned some meaning

I saw an astronomer with a telescope.

~~Knowledge~~ Knowledge Representation

Understanding language requires lots of knowledge.

- predicate logic
- semantic net

Scanned with CamScanner

Parameters in Natural Language Processing

① Auditory input → This basically consists of input devices such as text input, audio input and video inputs.

- Microphone
- MIDI keyboard or other digital musical instrument.

② Segmentation →

① Text segmentation → process of dividing written text into meaningful units, such as words, sentences or topics.

② Word segmentation → Dividing a string of written language into its component words.

③ Word splitting → process of ~~pass~~ parsing concatenated text to infer where word break exists.

Other segmentation problems → process may be required to segment text into segments besides words, including morphemes, paragraphs, topics or discourse turns.

~~different approaches~~

③ Syntax structure

④ Semantic structure

⑤ Pragmatic Analysis → This is high level knowledge which relates to the use of sentences in different contexts and how the context affects the meaning of the sentences.

Scanned with CamScanner

pragmatic - Handling pronouns.

— Handling pronouns such as "he", "she" "it" is not always straight forward.

Example 1

" John buys a new telescope. He sees Mary in the distance. He gets out his telescope. He looks at her through it"

Here, "her" refers to Mary who was not mentioned at all in the previous sentences.

John's Telescope was reffered to as "a new telescope", "His telescope ⇒ "it".

Example 2

" When is the next flight to Sydney?"

" Does it have any seat left?

Here, "it" refers to a particular flight to Sydney, not Sydney itself.

Scanned with CamScanner

Machine Translation

The term machine translation (MT) is used in the sense of translation of one language to another.
The ideal aim of MT system is to produce the best possible translation without human assistance.
Basically every MT system requires programs for translation and automated dictionaries and grammars to support translation.

Pre-editing and post editing in MT

- pre-editing means adjusting the input by marking prefixes, suffixes, ~~clau~~ clause boundaries etc.

- post-editing means improving quality by controlling the vocabulary & o/p of MT should be post edited.

Types of MT

① Rule Based MT (RBMT) → Translates on the basis of grammatical rules. It conducts grammatical analysis of the source language and the target language to generate translated sentence.
However, RBMT requires extensive proofreading, and its heavy dependence on lexicons means that efficiency is achieved after a long period of time.

It is also known as direct translation.

Scanned with CamScanner

Corpora → a collection of written texts.

Direct translation approach needs only a little ~~synat~~ syntactic and semantic analysis.
The analysis of SL texts is oriented to only one TL.

② Statistical Machine translation

- Statistical machine translation automatically learns lexical and structural preferences from corpora.
- Statistical models offer good solution to ambiguity problem.
- They are robust and work well even if there are errors and the presence of new data.

→ SMT works by reffering to statistical models that are based on the analysis of large volumes of bilingual texts. It aims to determine the correspondance between a word from the source language and a word from the target language.
→ A good example is Google translate.
→ Translations are not error-free, dont expect high quality translation.

③ Hybrid Machine Translation (HMT)

HMT, as the term indicates, is a blend of RBMT & SMT. It leverages a translation memory, making it far more effective in terms of quality.
However, even HMT has its share of drawbacks, the greatest of which is the need for expensive editing. Human translators will be required.

Scanned with CamScanner

④ Neural Machine translation (NMT)

NMT depends on neural network models to develop statistical methods for the purpose of translation.

- The primary benifit of NMT is that it provides a single system that can be trained to ~~decipher~~ decipher the source and target text.

- As a result, it doesnot depend on specialized systems that are common to other machine translation system, especially SMT

- To put it simply, NMT is an end-to-end translation system.

⑤ Empirical Machine translation (EMT)

- It uses large amount of raw data in the form of parallel corpora. The raw data consists of texts and their translations.

Example-based MT, analogy based MT, memory-based MT, and ~~case~~ case based MT are the techniques that use empirical approach.

- Basically all these ~~techniq~~ techniques use a corpus ~~data b~~ database of translated examples.

Scanned with CamScanner

⑥ Interlingua approach

Interlingua approach intends to translate SL texts to that of more than one language.

- Translation is from SL to an intermediate form called Interlingua (IL) and then from IL to TL.

Interlingua approach requires complete resolution of all ambiguities in the SL text.

Machine Vision Concept

Machine vision (MV) is the application of computer vision to industry and manufacturing. Whereas computer vision is the general discipline of making computers see (understand what is perceived visually), machine vision, being an engineering discipline, is interested in digital input/output devices and computer networks to control other manufacturing equipment such as robotic arms and equipment to eject defective products.

The machine vision is similar in complexity to voice recognition. The machine vision systems are video cameras, robots or other devices, and computers to visually analyze an operation or activity.

- One of the most common applications of machine vision is the inspection of manufactured goods such as semiconductor chips, automobiles, food & pharmaceuticals

Scanned with CamScanner

The components of machine vision system can vary, but there are several common factors found in most. These elements include:

① Image Sensor → This is also known as vision sensor, which is responsible for analysing images captured by the camera to determine the accuracy of assembly or presence of defects.

② Vision processor → It is the incharge of running machine vision algorithms.

③ Communication Interface → It connects the other parts ~~of~~ in the vision system. It is necessary to facilitate data transmission.
- This interface is usually attached to smart camera or other vision devices.
- Through this, the vision system can link data from the device to particular software that is run in a separate personal computer.

④ Lighting → Every vision system must have a working lighting analysis sequence.

Source of lighting ⇒ LED
Mercury
Xenon
Quartz Halogen
Fluorescent

⑤ Lens → Employed to disperse light ~~is~~ rays.

Scanned with CamScanner

Application areas of Machine vision:

→ Electronic component analysis
→ Signature identification
→ Optical character recognition
→ Handwriting recognition
→ Object recognition
→ pattern recognition
→ Material inspection
→ currency inspection
→ Medical image analysis.

Computer Vision

Computer vision is the science and technology of machines that see, where see in this case means that the machine is able to extract information from an image that is necessary to solve some task.

- Computer vision is concerned with the theory behind Artificial systems that extract information from images.

Applications:

- Controlling processes (eg: an industrial robot or an autonomous vehicle)
- Detecting events (e.g. for visual curveillance or people counting).
- Organizing informations (for indexing databases of images and image sequences).
- Modeling objects or environments (industrial inspections, medical image analysis, or

Scanned with CamScanner

topographical modeling).

- Interaction (computer - human interaction)

- Computer vision is, in some ways, the inverse of computer graphics.
While Computer graphics produces image data from 3D models, computer vision often produces 3D models from image data.

- There is also a trend towards a combination of the two disciplines e.g. as explored in augmented reality.

Sub-domains of Computer vision:

- Scene reconstruction
- event detection
- video tracking
- object recognition
- learning
- indexing
- motion estimation
- and image restoration.

Scanned with CamScanner

Robotics

Robotics is an interdisciplinary branch of engineering and science that include mechanical engineering, electronic engineering, information engineering, computer science, and others.

- Robotics deals with the design, construction, operation and use of robots, as well as computer systems for their control, sensory feedback, and information processing.

A robot generally consists of 5 basic components:

(a) Controller

Every robot is connected to a computer controller, which regulates the components of the arm and keeps them working together.

- The controller also allows the robot to be networked to other systems, so that it may work together with other machines, processes or robots.

Almost all robots are pre-programmed using "teaching" devices or offline s/w programs.

(b) Arm

The arm is the part of the robot that positions the end-effector and sensors to do their pre-programmed business. Many are built to resemble human arms, and have shoulders, elbows, wrists, even fingers. Each joint is said to give

Scanned with CamScanner

the robot 1 degree of freedom.

A simple robot arm with 3 degree of freedom could move in 3 ways: up and down, left and right, forward and backward.

- Most working robots have 6 degree of freedom to allow them to reach any possible point in space within its work envelope.

- The human arm has 7.

(c) Drive

The links are moved into their desired position by the drive.
Typically, a drive is powered by pneumatic or hydraulic pressure or electricity.

(d) End - effector

- The end - effector could be thought of as the "hand" on the end of the robotic arm.
There are many possible end - effectors including a gripper, a vacuum pump, tweezers, scalpel, blowtorch, welder, spray gun or just anything that helps it to do its job.

(e) Sensor

The sensor sends information, in the form of electronic signals back to the controller.
- Robots sensors can detect infrared radiation to see in the dark.
→ computer vision and object recognition is the field of sensor development.

Scanned with CamScanner