Unit-5 : Machine Learning

- Ankit Pangeni

* Introduction to Machine Learning

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

The machine starts learning with observations or data, in order to look for patterns in data & make better decisions in the future based on the examples it has been provided.

Some successful applications of machine learning are:
- Computer vision processing
- Natural language processing
- Forecasting, pattern recognition, games, expert systems and so on.

Traditional programming vs machine learning

Data → Computer → output
Program → Computer

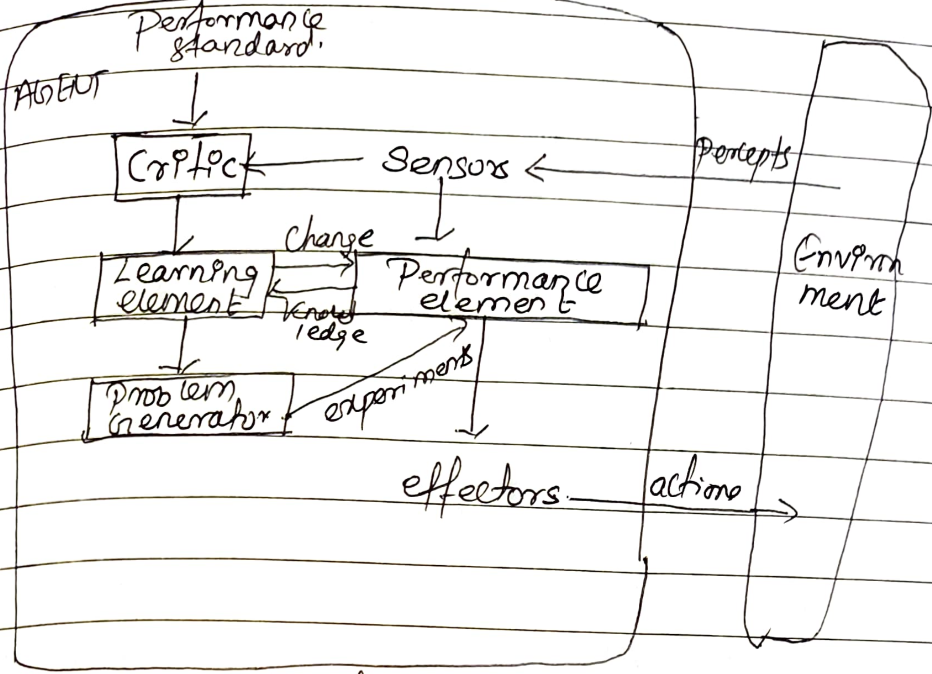
fig: Traditional programming

Data → Computer → Program
output → Computer

fig: Machine learning

* Concept of Learning (General Machine Learning agent Framework or architecture)

The following fig. represents the general steps involved in machine learning agent. It learns from inputs & corrects its decision based on the feedback provided by critic



It consists of following component

1. Learning element : Receives & processes the input obtained from a person, from reference material like magazines, journals, etc or from the environment at large.

2. Knowledge base : Somewhat similar to the database. Initially it may contain some basic knowledge, but new knowledge can be added as well as existing can be replaced.

3. Performance element : Uses the updated knowledge base to perform some tasks or solves some problems & produces output.

4. Feedback element : Receives two inputs, one from learning element & one from standard system. It is used to determine what should be done to produce the correct output.

5. Standard system : It is a well programmed system that is able to produce the correct output.

* Types of Machine learning

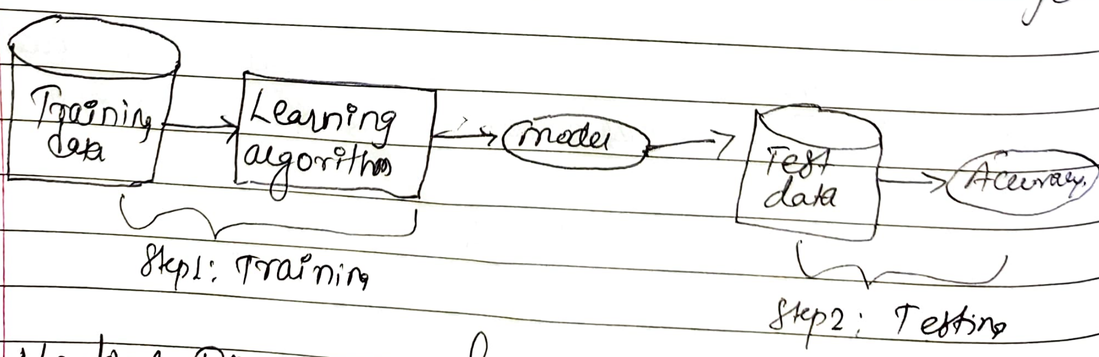
(I) Supervised Learning:

Here, the system is supplied with a set of training examples consisting of inputs & corresponding outputs, & is required to discover the relation or mapping between them

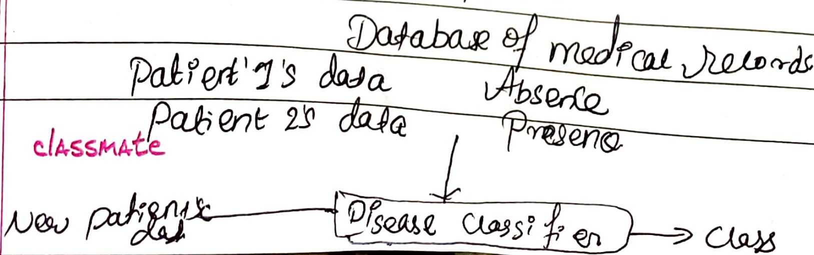
The machine is presented with inputs together with the target outputs. Then, the machine network tries to produce an output as close as possible to the target signal by adjusting the values of internal weights. Supervised learning can take what it has learnt in the past & apply that to a new data using labelled example to predict future patterns & events.

Classification is an example of supervised learning. It is a two step process:

- Model construction: describing a set of predetermined classes
- Model usage:- for classifying future or unknown objects



Ex: Medical Disease Classification



DATE

(2) Unsupervised Learning

Unsupervised learning algorithms are used when the information used to train is neither classified nor labeled. It studies how systems can infer a function to describe a hidden structure from unlabelled data. 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 data.

Examples: Clustering the data into similar group, predict new target market

The main difference between supervised & unsupervised learning is that Supervised learning relies on labelled input & output training data, whereas unsupervised learning processes unlabelled or raw data

(3) Semi-Supervised Learning

They fall somewhere in between supervised & unsupervised learning, since they use both labeled & unlabeled data for training - typically a small amount of labeled data & a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, Semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it & learn from it. Otherwise, acquiring unlabeled data generally doesn't require additional resources. Ex: classification of market after searching information from google.

DATE

(4) Reinforcement Learning

It is a type of dynamic learning that trains agent using reward & punishment. In this learning, agent learns by interacting with environment & consists of three components: Agent (learner), Environment (Agent interacts with) and Action (what agent can do). It is employed by various software & machines to find the best possible behaviour or path it should take in a specific situation.

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

Types of reinforcement:

- Positive: When an event occurs due to a particular behaviour, increases the strength & frequency of the behaviour. It has a positive effect on the behaviour

Advantage:- Maximizes performance, sustain change for long period of time.

Disadvantage:- Overhead of states can diminish the results.

- Negative:- Strengthening of a behaviour because a negative condition is stopped or avoided.

Advantage:- Increases behaviour.

Disadvantage:- Only provides enough to meet up a minimum behaviour

classmate

PAGE

Statistical based learning : Naive Bayes Model

Basics:-

- Probability of an event X : $P(X) = \frac{\text{No. of times } X \text{ occurs}}{\text{Total no. of events in Sample space}}$

- Joint probability :

$P(A \cap B) = P(A) \times P(B)$ if events are independent

$P(A \cap B) = P(A/B) * P(B)$

$\left[P(A/B) = \frac{P(A \cap B)}{P(B)} \right.$ i.e. $\left. P(A \cap B) = P(A/B) \cdot P(B) \right]$

- Bayes Theorem: $P(A/B) = \frac{P(B/A) \cdot P(A)}{P(B)}$

Naive Baye's Learning / classifier:

It is a classification technique based on Baye's theorem. It assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. ex: a fruit may be considered to be an apple, if it is red & about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple & that's why it is known as 'Naive'

Bayes theorem provides a way of calculating posterior probability $P(C/X)$ from $P(C)$, $P(X)$ & $P(X/C)$

$$P(C/X) = \frac{P(X/C) \cdot P(C)}{P(X)}$$

$P(C/X)$ → posterior prob.

$P(X/C)$ → likelihood

$P(C)$ → class posterior prob.

$P(X)$ → predictor prior prob.

DATE

$$P(c/x) = P(x_1/c) \times P(x_2/c) \times \cdots \times P(x_n/c) \times P(c)$$

Applications of Naive Baye's Model

→ Real time prediction
→ Text classification
→ Recommendation system.

Problems with Naive Bay
- Zero frequency problem
- Attribute independence

Ex: From the given data set decide whether to play golf or not in today's condition.

today (Sunny, Hot, Normal, False)

S.N	Outlook	Temperature	Humidity	Wind	Play golf
1.	Rainy	Hot	High	False	NO
2.	Rainy	Hot	High	True	No
3.	Overcast	Hot	High	False	Yes
4.	Sunny	Mild	High	False	Yes
5.	Sunny	Cool	Normal	False	yes
6.	Sunny	Cool	Normal	True	No
7.	Overcast	Cool	Normal	True	yes
8.	Rainy	Mild	High	False	NO
9.	Rainy	Cool	Normal	False	yes
10.	Sunny	Mild	Normal	False	Yes
11.	Rainy	Mild	Normal	True	yes
12.	Overcast	Mild	High	True	Yes
13.	Overcast	Hot	Normal	False	Yes
14.	Sunny	Mild	High	False	NO

8/10/24

For outlook

	Yes	No	P(yes)	P(No)
Sunny	3	2	3/9	2/5
Overcast	4	0	4/9	0
Rainy	2	3	2/9	3/5
Total	9	5		

For Temperature.

	Yes	No	p(yes)	P(NO)
Hot	2	2	2/9	2/5
Cool	3	1	3/9	1/5
Mild	4	2	4/9	2/5
Total	9	5		

For Humidity

	Yes	No	P(yes)	p(No)
High	3	4	3/9	4/5
Normal	6	1	6/9	1/5
Total	9	5		

For wind

	Yes	No	p(yes)	p(NO)
True	3	2	3/9	2/5
False	6	3	6/9	3/5
Total	9	5		

Now,

$$P\left(\frac{Yes}{Today}\right) = P\left(\frac{Sunny}{Yes}\right) \times P\left(\frac{HOT}{Yes}\right) \times P\left(\frac{Normal}{Yes}\right) \times P\left(\frac{False}{Yes}\right)$$

$$= \frac{3}{9} \times \frac{2}{9} \times \frac{6}{9} \times \frac{6}{9} \times \frac{9}{14}$$

$$= 0.0212$$

$$P\left(\frac{No}{Today}\right) = P\left(\frac{Sunny}{No}\right) \times P\left(\frac{HOT}{No}\right) \times P\left(\frac{Normal}{No}\right) \times P\left(\frac{False}{No}\right)$$

$$= \frac{2}{5} \times \frac{2}{5} \times \frac{1}{5} \times \frac{3}{5} \times \frac{5}{14}$$

$$= 0.0069$$

Since, sum of probabilities must be equal to 1, so we have to normalize.

$$P\left(\frac{Yes}{Today}\right) = \frac{P\left(\frac{Yes}{Today}\right)}{P\left(\frac{Yes}{Today}\right) + P\left(\frac{No}{Today}\right)}$$

$$= \frac{0.0212}{0.0212 + 0.0069} = 0.7544$$

$$P\left(\frac{NO}{Today}\right) = \frac{0.0069}{0.0069 + 0.0212} = 0.2456$$

- Since, $P\left(\frac{Yes}{Today}\right) > P\left(\frac{No}{Today}\right)$. So, they can play golf in today's condition

DATE

* Learning by Genetic Algorithms

Genetic Algorithms (GAs) are adaptive heuristic search algorithm that belongs to the larger part of evolutionary algorithms. It is based on the idea of neural selection & genetics. The historical data are provided to find out better solution or simply GA's are used to generate high quality solutions for optimization & search problem.

GA's simulate the process of natural selection which means those species who can adopt to change their environment are able to survive & reproduce next generation (fittest survival). Each generation consists of a population of individuals & each individual represents a point in search space & possible solutions which are represented as string of characters/ integers/ floats/ bits.

Operations in Genetic Algorithm

i) Reproduction / selection → This operator provides the performance to the individuals with good fitness score and allow them to pass their genes to the successive generation. To select the parents, the techniques are round wheel, tournament selection, rank selection, etc.

ii) Crossover → This represents mating between two individuals. Two individual are selected using selection operators & crossover sites are chosen randomly then

DATE

genes at the crossover sites are exchanged thus it creates completely new offsprings.

Ex: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

→ | 0 | 1 | 2 | 3 | 4 | 5 | 7 | 5 | 8 |

| 5 | 8 | 9 | 4 | 2 | 3 | 5 | 7 | 5 | 8 |

The popular crossovers are one-point crossover, multipoint crossover & uniform crossover.

iii) Mutation → The key idea is to insert random genes in offspring to maintain the diversity in the population to avoid the premature convergence.

Ex: | 0 | 0 | 1 | 1 | 0 | 1 | → | 0 | 0 | 1 | 0 | 0 | 1 |

Before mutation   After mutation

iv) Fitness function → It is the function which takes a candidate solution to the problem as input & produce output, determines how good or fit candidate is.

Genetic Algorithm

→ Randomly initialize population
→ Determine fitness of population
→ until convergence repeat

DATE

→ Select parents from population
→ Crossover & generate new population
→ perform mutation on new population
→ Calculate fitness for new population.

Applications of GA's

→ Recurrance Neural Networks
→ Mutation Testing
→ Learning Fuzzy rule.

* Difference between supervised & unsupervised learning

Supervised Learning	Unsupervised learning
i) Supervised learning algorithms are trained using labelled data.	i) Unsupervised learning algo. are trained using unlabeled data.
ii) Supervised learning model predicts the output.	ii) Unsupervised learning model finds the hidden paterns in data.
iii) Input data is provided to the model along with output.	iii) Only input data is provided to the model.
iv) It needs supervision to train the model.	iv) It does not need any supervision to train the model.
v) Supervised learning model produces an accurate result.	v) Unsupervised learning model may give less accurate result as compared to supervised learning.

DATE

* Learning with Neural Networks

Introduction

* Artificial Neural Network (ANN):-

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

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous system. We can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements.

The goal of ANN is to solve problems in the same way that human mind does. ANN are widely used in computer vision, speech recognition, medical diagnosis, face recognition, signature verification, etc.

* Biological Neural Networks (BNN) vs Artificial Neural Networks (ANN)

Criteria	BNN	ANN
i) Processing	Massively parallel, slow but superior than ANN	Massively parallel, fast but inferior than BNN
ii) Size	10^{11} neurons and 10^{15} interconnections.	10^2 to 10^4 nodes (mainly depends on type of application & network designer)
iii) Learning	They can tolerate ambiguity.	Very precise, structured & formatted data is required to tolerate ambiguity
iv) Fault tolerance	Performance degrades with even partial damage.	It is capable of robust performance, hence has the potential to be fault tolerant
v) Storage Capacity	Stores the information in the synapse.	Store the information in continuous memory locations.

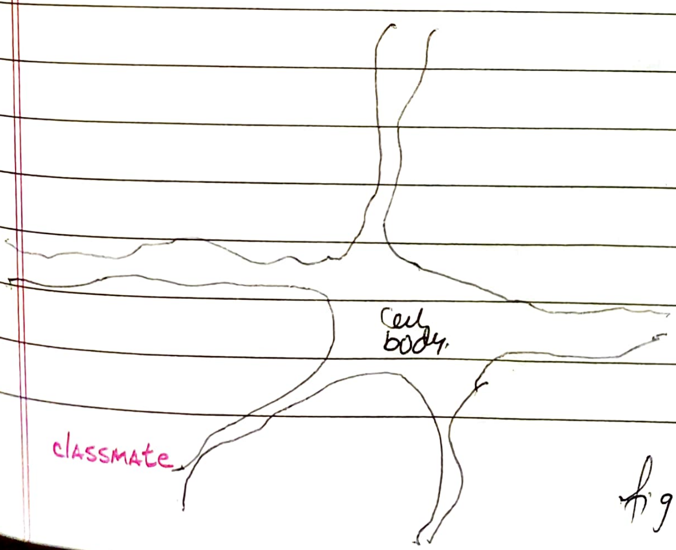
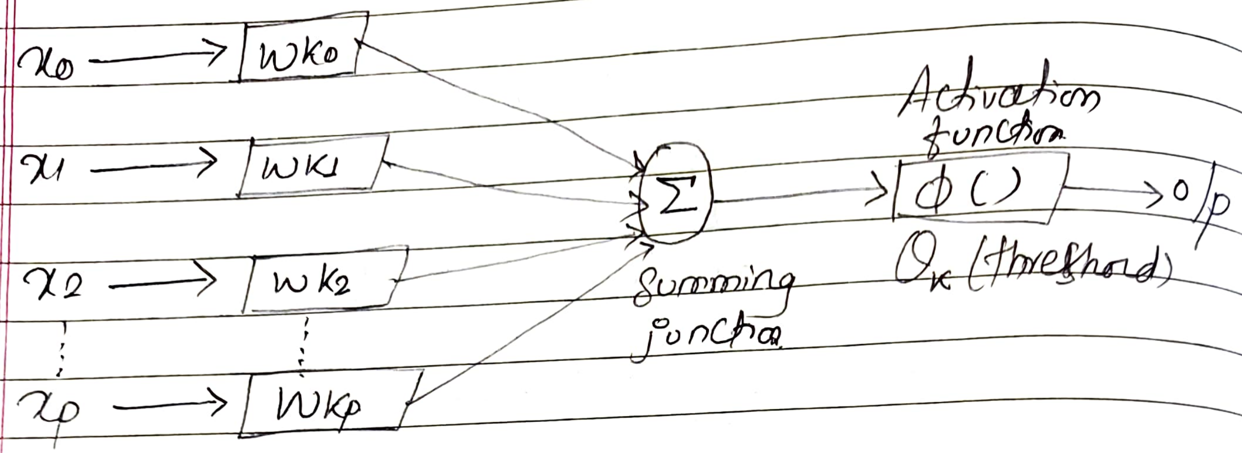


fig: Biological neuron

DATE

* Mathematical Model of ANN.



Input symbols weights fig: Mathematical model of ANN.

Components:

- Inputs: The numerical value $x_0, x_1, x_2, \ldots x_p$ are the inputs to neuron. Each link has a weight associated with it which determines strength & sign of the connection
- Activation function: Used to derive output activation from the input activations to a given node.
- Bias weight: It is used to set the threshold for a unit. Unit is activated when the weighted sum of real inputs exceeds the bias weight

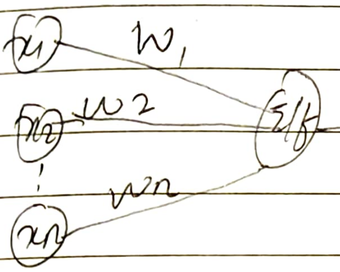
The calculation for a single neuron goes as follows

Each unit first computes a weighted sum of its inputs

$$in_i = \sum_{j=0}^{n} W_{j,i} \, x_j$$

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

$$O_i = g(in_i) = g\left(\sum_{j=0}^{n} W_{j,i} \, x_j\right)$$



output

$$f\left(b + \sum_{i=1}^{n} x_i w_i\right)$$

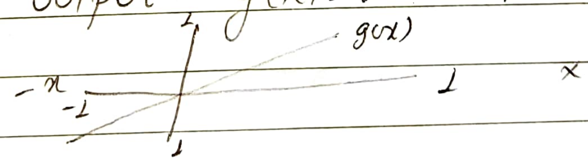
bias function threshold

* Types of Activation Functions:

Activation function typically falls into one of three categories:

- **Linear activation function**

  Here, the output activity is proportional to the total weighted output. $g(x) = kx + c$, where k and x are constant.

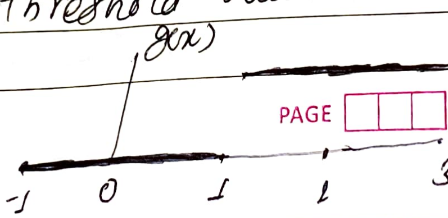


- **Threshold activation function**

  Here, the output are set at one of the two levels, depending on whether the input is greater than or less than some threshold value.

  $g(x) = 1$ if $x \geq K$

  $g(x) = 0$ if $x < K$



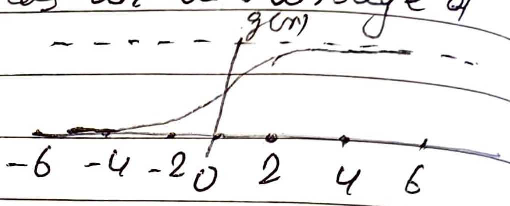
PAGE

classmate

- ## Sigmoid activation functions

Here the output varies continuously but not linearly as the input changes. Sigmoid units bear a greater resemblance to real neurons than do linear or threshold units. It has an advantage of differentiable.

$$g(x) = \frac{1}{1+e^{-x}}$$



Types of ANN

- ## Feed-forward networks

Feed-forward ANNs allow signals to travel one way only ; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organization is also referred to as bottom-up or top-down.

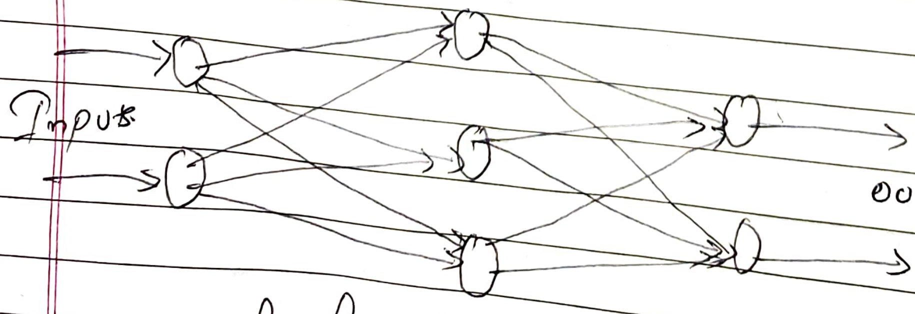
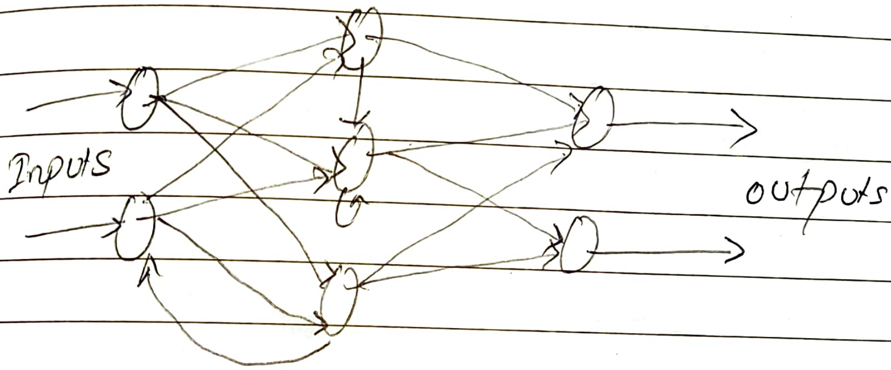


fig: feed-forward networks

classmate

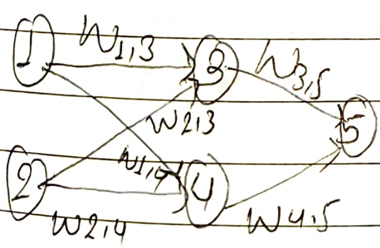
- Feedback networks (Recurrent networks)

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



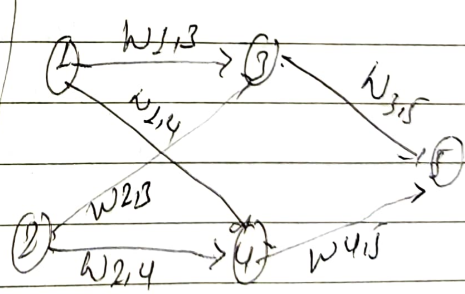
Example

Feed forward:-



$$g_3 = g(W_{1,3} a_1 + W_{2,3} a_2)$$
$$g_4 = g(W_{1,4} a_1 + W_{2,4} a_2)$$
$$g_5 = g(W_{4,5} a_2 + W_{3,5} a_3)$$

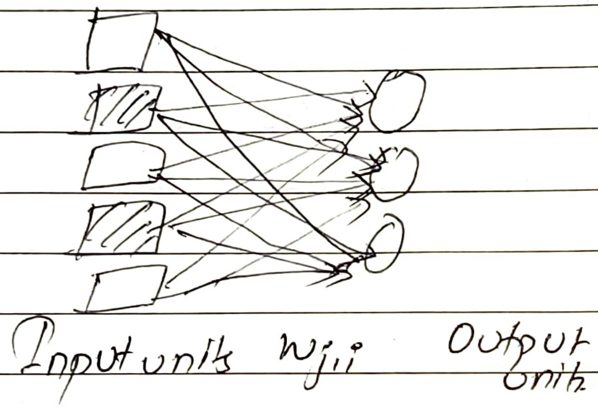
Feed backward



$$g_5 = g(W_{3,5} a_3 + W_{4,5} a_4)$$
$$= g(W_{3,5}\, g(W_{1,3} a_1 + W_{2,3} a_2) + g(W_{1,4} a_1 + W_{2,4} a_2))$$

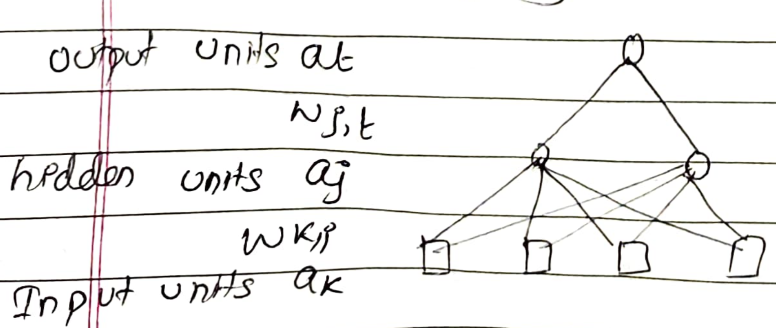
- Single-layer neural network (perception)

A feed forward neural network in which all the inputs connected directly to the outputs is called a single layer neural network, or a perception network. Since each output unit is independent of the others each weight affects only one of the outputs.



- Multilayer neural networks (perceptrons)

A feed forward neural network which contains input layers, output layers and some hidden layers also is called multilayer nn. The advantage of adding hidden layers is that it enlarge the space of hypothess. Layers of this network are normally fully connected



DATE

Application of Artificial Neural Networks

- **Speech Recognition.**

Speech occupies a prominent role in human-human interaction. Therefore, it is natural for people to expect speech interfaces with computers. ANN has made great progress in this field. Following ANNs have been used for speech recognition: Multilayer networks, multilayer networks with recurrent connections, kohonen self-organizing feature map.

- **Character Recognition**

It is an interesting problem which falls under the general area of pattern recognition. Many neural networks have been developed for automatic recognition of handwritten characters, either letters or digits. Multilayer neural networks such as Backpropagation neural networks, neocognition ANNs have been used for character recognition.

- **Signature verification.**

Signatures are one of the most useful ways to authorize & authenticate a person in legal transactions. Signature verification technique is a non-vision based technique. The trained neural network will classify the signature as being genuine or forged under the verification stage.

- **Human face recognition**

It is a typical task because of the characterization of "non-face" images. However, if a neural network is well trained, then it can be divided into two classes namely images having faces & images that don't have faces.

DATE

First, all input images must be processed. Then the dimensionality of that image must be reduced. And at last it must be classified using neural network training algorithm.

* Learning by Training ANN.

Learning means to adopt the changes in itself when there is change in environment. In complex system of ANN, after learning it changes its internal structure based on the information passing through it.

Learning is important in ANN because of some changes occuring in environment then ANN must change its inputs, outputs, activation function & weight. To change the internal structure of ANN. There are various methods (i.e. learning rules) which are as follows.

1) Hebbian Learning

It is a general principle that states that the synaptic efficiency between two neurons should increase if the two neurons are 'simultaneously' active, and decrease if not.

Hebb's Law can be represented in the form of two rules. It provides the basis for learning without a teacher.

DATE

- If two neurons on either side of a connection are activated synchronously, then the weight of that connection is increased.

- If two neurons on either side of a connection are activated asynchronously, then the weight of that connection is decreased.

Hebb's Algorithm:-

Step 1: Initialize all weights and bias to 0.
Step 2: Given a training input s, with its target output t, set the activations of the input units: $x_i = s_i$
Step 3: Set the activation of the output unit to the target value : $y = t$
Step 4: Adjust the weights. $w_i(new) = w_i(old) + x_i y$.
Step 5: Adjust the bias (just like the weights): $b(new) = b(old) + y$

Pseudocode for Hebb's net (supervised)

Initialization: for i=1 to n; { b=0, wi=0 }
For each of the training samples s:t do
{ /* s is the input pattern, t is target output to sample */
 for i=1 to n { xi=si } /* set to input units */
 y=t /* set y to the target */
 for i=1 to n {
 wi = wi + xi * y } /* update weights */
 b = b + xi * y /* update bias */
}

DATE

2) Perceptron Learning:

Perceptron learning is used in supervised learning rule to classify the data in two classes. It consists of a single neuron with arbitary value 1 or 0 depending on threshold & also consist of bias weight. Training patterns are presented to the networks inputs; the output is computed.

Variables used:

- $y = f(x)$ denotes the output for an input vector x.
- $D = (x_1, d_1), (x_2, d_2), ----, (x_n, d_n)$ is the training set of n samples where $x_j \rightarrow$ input vector

$d_j \rightarrow$ desired output for x_j

Algorithm:

Step-1: Initialize weights and threshold

Step-2: For each example j in our training set D, perform following steps 3 & 4 over the input x_j and desired output d_j

Step-3: Calculate the actual output as:

$$y(t) = g[w_0(t)x_0(t) + w_1(t)x_1(t) + ---- + w_n(t)x_n(t)]$$

i.e. $y(t) = g[\sum w_i(t) \cdot x_i(t)]$

Step-4: Update the weights as

$$w_i(t+1) = w_i(t) + \alpha\left(d(t) - y(t)\right)x_i(t)$$

where,

$0 \leq \alpha \leq 1$ (learning rate)

DATE

3) Back-propagation Learning

It is a supervised learning method, and is an implementation of Delta rule. It requires a teacher that can calculate the desired output for any given input. It is most useful for feed-forward networks. Back propagation requires that the activation function used by the artificial neurons (or nodes) is differentiable.

Algorithm:

Step-1: Randomly choose the initial weight.

Step-2: Compute output (y) from neural network.

$$y = g(\Sigma x_i w_i) \text{ where } g(x) = \frac{1}{1+e^{-x}} \text{ is sigmoid function}$$

Step-3: Compute error $\delta = t - y$; where t is targeted output & y is actual output

Step-4: Propagate this δ error back to layer of neural network and compute error at each layer as:

$$\delta_i = \Sigma w_{ij} \delta_j$$

Step-5: Update the weight as

$$w_{ij}(new) = w_{ij}(old) + \alpha \delta_j \frac{d(y_i)}{de} x_i$$

where,

$\frac{d(y_i)}{de}$ is derivative of y w.r.t e

α is learning rate $0 \le \alpha \le 1$

x_i is input to node.

Note: XOR Gate implementation is not possible with Hebbs network

Numericals

Hebbian Learning

Q. Implement logic 'AND' operation using 'Hebb Net'.

X_1	X_2	b	Y
1	1	1	1
1	-1	1	-1
-1	1	1	-1
-1	-1	1	-1

Bipolar inputs

→ Initialize weights and bias ie. $w_1 = w_2 = b = 0$
update weights.

1. For input pair $x_1, x_2, b = (1, 1, 1)$

$w_1(new) = w_1(old) + (x_1 * y)$
$= 0 + 1 \times 1 = 1$

$w_2(new) = w_2(old) + (x_2 * y)$
$= 0 + 1 \times 1 = 1$

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

2. For input pair $(x_1, x_2, b) = (1, -1, 1)$

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

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

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

3. For input pair $(x_1, x_2, b) = (-1, 1, 1)$

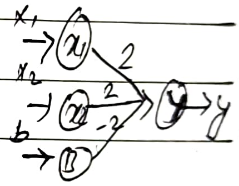
$w_1(new) = w_1(old) + (x_1 * y) = 0 + (-1) \times (-1) = 1$
$w_2(new) = w_2(old) + (x_2 * y) = 2 + (1) * (-1) = 1$
$b(new) = b(old) + y = 0 + (-1) = -1$

4. For input pair $(x_1, x_2, b) = (-1, -1, 1)$

$w_1(new) = w_1(old) + (x_1 * y) = 1 + (-1) \times (-1) = 2$
$w_2(new) = w_2(old) + (x_2 * y) = 1 + (-1) \times (-1) = 2$
$b(new) = b(old) + y = (-1) + (-1) = -2$

So, the final weight matrix is $(2, 2, -2)$

$\therefore (w_1, w_2, b) = (2, 2, -2)$



Testing network

Now, net value is.

Net value $= w_1 x_1 + w_2 x_2 + b$

For input $(x_1, x_2) = (1, 1)$
Net $= 2 \times 1 + 2 \times 1 - 2 = 2$

For input $(x_1, x_2) = (1, -1)$
Net $= 2 \times 1 + 2 \times (-1) - 2 = -2$

For input $(x_1, x_2) = (-1, 1)$
Net $= 2 \times (-1) + 2 \times 1 - 2 = -2$

For input $(x_1, x_2) = (-1, -1)$
Net $= 2 \times (-1) + 2(-1) - 2 = -6$

The results are all compatible with the original data. (i.e. 1st value is +ve, all other -ve)

Decision Boundary

We have Eqn,

$$w_1 x_1 + w_2 x_2 + b = y$$

$$2x_1 + 2x_2 - 2b = 0$$ (from Heb's Network)

Since, bias $b = 1$

$$2 \cdot x_1 + 2x_2 - 2 = 0$$

$$2x_1 + 2x_2 = 2$$

$$x_1 + x_2 = 1$$

$$x_2 = 1 - x_1$$

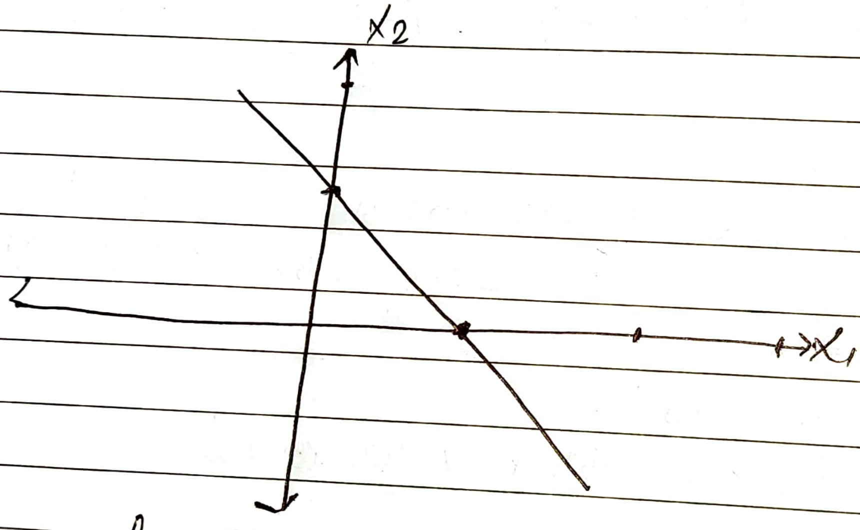


fig: Decision Boundary of AND function

classmate

Perceptron Learning

Q. Implement AND function using Perceptron network.

X_1	X_2	b	Y
1	1	1	1
1	-1	1	-1
-1	1	1	-1
-1	-1	1	-1

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

=) Initialize the weights and bias = 0 and learning rate to 1.

$W_1 = W_2 = b = 0$ & $\alpha = 1$

1. For input pattern 111 i.e. $X_1 = 1$, $X_2 = 1$, $t = 1$ (output)

Calculate the Net input, $y_{in} = b + W_1 X_1 + W_2 X_2$

$= 0 + 0 * 1 + 0 * 1$

$= 0$

Since, $y_{in} = 0$

So, $y = f(y_{in}) = 0$

Check $y = t$, since $y = 0$, and $t = 1$ so; $y \neq t$, so weight change is required.

$W_1(new) = W_1(old) + \alpha t X_1$

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

$W_2(new) = W_2(old) + \alpha t X_2$

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

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

So, $W_1 = W_2 = b = 1$

2. For input pattern 1 -1 -1, i.e. $x_1 = 1, x_2 = -1, t = -1$

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

Since, $y(in) > 0$ $y = f(y_{in}) = 1$

Check, $y = t$, NO, so weight change is required

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

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

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

So, $w_1 = 0$, $w_2 = 2$, $b = 0$

3. For input pattern -1 1 -1, i.e. $x_1 = -1$, $x_2 = 1$, $t = -1$

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

Since, $y(in) > 0$, so, $y = f(y_{in}) = 1$

Check $y = t$, NO, so weight change is required

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

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

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

4. For input pattern -1 -1 -1, i.e. $x_1 = -1$, $x_2 = -1$, $t = -1$

Net input $y_{in} = b + w_1 x_1 + w_2 x_2 = -1 + 1 \times -1 + 1 \times -1 = -3$

Since, $y(in) < 0$, so, $y = f(y_{in}) = -1$

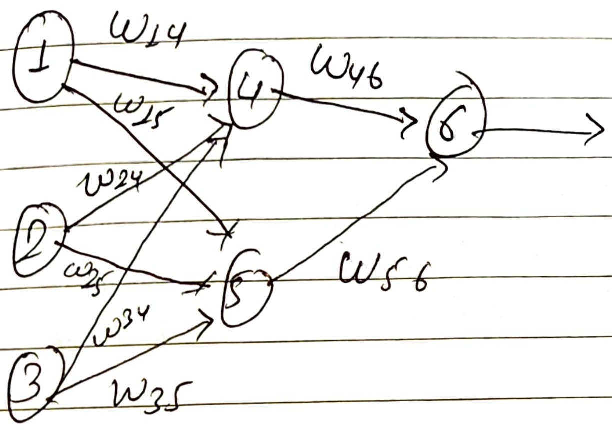
Check, $y = t$, Yes, So weight change is not required

So, Final weights: $w_1 = 1$, $w_2 = 1$, $b = 1$

DATE

Back propagation

Q. Consider the following network



We are tracing the BP for input (1, 0, 1) with label as

Now, let us consider the initial weight & bias value as:

x_1	x_2	x_3	w_{14}	w_{15}	w_{24}	w_{25}	w_{34}	w_{35}	w_{46}	w_{56}
1	0	1	0.2	-0.3	0.4	0.1	-0.5	0.2	-0.3	-0.2

θ_1	θ_2	θ_3
-0.4	0.2	0.1

Now, calculate the output for each node 4, 5, 6 &

Node j	Net input I_j	Output O_j
4	$0.2 + 0 - 0.5 - 0.4 = -0.7$	0.332
5	$-0.3 + 0 + 0.2 + 0.2 = 0.1$	0.525
6	$-0.3 \times 0.332 - 0.2 \times 0.525 + 0.1$ $= -0.105$	0.474

DATE

Then calculate the error at each node

Node J	Err_J
6	$0.474(1-0.474)(1-0.474) = 0.1311$
5	$0.525(1-0.525)(0.1311)(-0.2) = -0.0065$
4	$0.332(1-0.332)(0.1311)(-0.3) = -0.0087$

$\left\{ O_j(1-O_j)(T_j-O_j) \right.$

Now update the weight & bias as follows (use formulas $\Delta w_{ij} = (l)\, Err_j O_i$, and $w_{ij} = w_{ij} + \Delta w_{ij}$) with learning rate $(l) = 0.9$

Weight or bias	New value
w_{46}	$-0.3 + 0.9 \times 0.1311 \times 0.332 = -0.261$
w_{56}	$-0.2 + 0.9 \times 0.1311 \times 0.525 = -0.138$
w_{14}	$0.2 + 0.9 \times -0.0087 \times 1 = 0.192$

And so on

Questions asked from this chapter

Q. Describe mathematical model of neural network. What does it mean to train a neural network? Write algorithm for perceptron learning. (2078 - 10 marks)

Q. What is crossover operation in genetic algorithm? Given following chromosomes, show the result of one-point & two point crossover.

C1 = 01100010

C2 = 10101100

Choose appropriate crossover point as your assumption. (2078 - 5 marks)

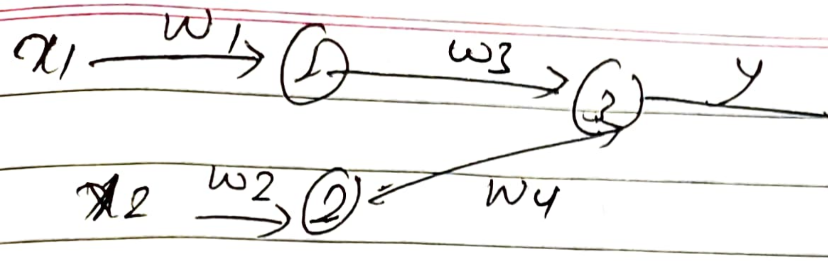
Q. Write an algorithm for learning by Genetic approach. (2076 - 5 marks)

Q. Define mathematical mode of ANN. Discuss how Hebbian learning algorithm can be used to train a neural network. Support your answer with an example. (2076 - 10 marks)

Q. What is machine learning? How genetic algorithm can be used to train agents? Discuss the operations of genetic algorithm. (2076 (old) - 5 marks)

Q. What does it mean to train a neural network? Consider following neural network, how back-propagation can be used to train it? (2076 (old) - 6 marks)

DATE



Q. Define learning. Why learning framework is required? Explain about learning framework with block diagram & examples. (2067 - 6 marks). (2069 - 6 marks) (2071 - 6 marks)

Q. What is back propagation? Explain all the steps involved in it with example. (2067 - 6 marks)

Q. What do you mean by machine vision? Discuss the components of machine vision system. (2074 - 5 marks)

Q. Consider a feed-forward neural network with your own assumptions of inputs & weights & express it mathematically. (2076 (old) - 6 marks)