Deep learning

UNIT - I

1) Biological neuron → 3) Difference b/w them.

2) artificial neuron

4) Different hyper Parameters

5) Limits of Traditional Computer Program.

6) feed-forward neural network

7) Different Activation functions : Sigmoid, Tanh, Relu..

8) Perceptron → working, and/or logic Example.

9) Linear Perceptron as Neurons. & limitations.

---

10) Back Propagation algorithm, in Reducing Error.

11) Gradient descent with Sigmoid Neuron → Derivation

12) delta Rule for training linear neurons.

13) Preventing Overfitting in DL methods & Demonstration.

14) Stochastic & mini batch Gradient. → $\boxed{\text{Comparision}}$

15) Mechanics of ML.

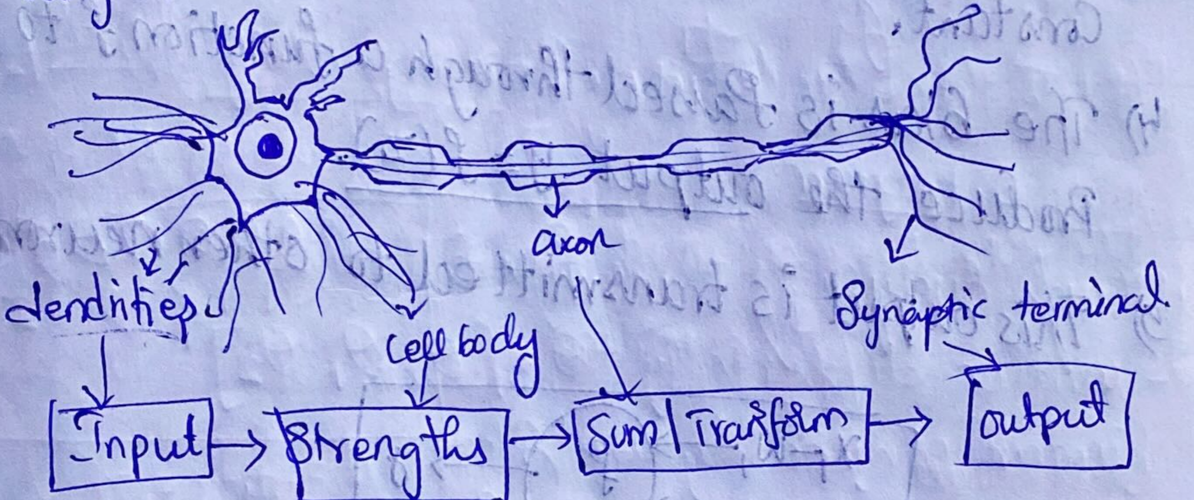
Unit 1 Chapter 1 (Q1, 2, 4, 5, 6, 7, 8, 9, 15).

1) Biological Neurons:

1) The foundational unit of human brain is the neuron.

*) It is the massive biological network that enables us to experience outside world.

*) As it's core, the neuron is optimized to receive info from other neuron / Process the info in a unique way and send it's results to other cell.



→ Explain Process:-

1) The neuron Recieves input along dendrities.

2) Each of incoming connection is dynamically strengthened or weakened based on how it is used.

3) After being weighted by the strength of their respective connection, the input goes to cell body.

4) This sum is then transformed into a new signal that's propagated along the cell axon's & send off to other neuron,

② Artificial neuron:

4703 A

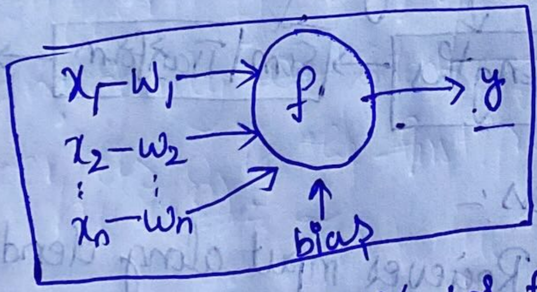
1) ~~S~~uch as Biological neuron the (a.N) takes in some no of inputs $x_1, x_2, \ldots x_n$ each of which is multiplied by weight, $w_1, w_2, w_3 \ldots w_n$.

2) These weighted inputs are, as before, summed together to produce the logit of the neuron, $z = \sum_{i=0}^{n} w_i x_i$.

3) The logit is also called as bias which is constant.

4) The bias is passed through a function f to produce the output $y = f(z)$.

5) This output is transmitted to other neurons



we can express this in vector form

input vector $x = [x_1, x_2, \ldots x_n]$

weight vector $W = [w_1, w_2 \ldots W_n]$

Expression: $y = f(x \cdot W + b)$ where b is bias term

3)

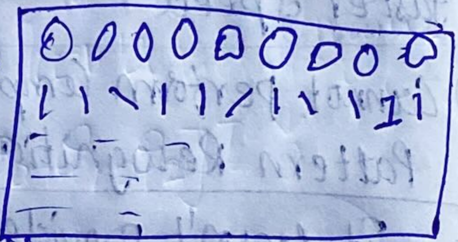
Artificial Neuron	Biological Neuron
1) It is a mathematical model which is mainly inspired by the biological neuron system in the human brain.	It is also composed of several processing pieces known as Neurons.
2) Its Processing was Sequential and centralized.	It Process the info in a Parallel and distribtive manner.
3) Small in size	large in size.
4) It proces the info at a faster speed	It proces the info at a slow speed.
5) Cannot perform Complex Pattern Recognition	Perform complicated tasks.
6) It doesn't provide any feedback	It Provides feedback.
7) No fault tolerance	fault tolerance is their.
8) Operating Environment is well defined and well-constrained	Operating Environment is poorly defined and uncon-strained.
9) Reliability: It is very Vulnerable	It is robust.
10) Response time is measured in milliseconds.	It's Response time is measured in nanoseconds
11) It has very accurate Stru-cture	They are tolerant & ambiguity.

4)

5) Limits of Traditional Computer Programs.

* Traditional computer programs are designed to be very good at two things
  1) Performing arithmetic really fast.
  2) Explicitly following a list of instructions.
* The T.C.P are good at heavy financial number crunching.

Let us take an Example: Recognizing hand written digits Program



although Every digit in the figure is written differently the human can Easily ~~writ~~ Identify the number as 0 in the first row, 1 is in second row etc.,

But if we are said to write Program to automatically read Some one's handwriting, it will be very difficult.

If we might State that we have a "Zero" if our image only has a Single, closed loop. but this isn't really a Sufficient Condition what if Someone Write a messy Zero, it will be a very diffcult task to identify it.

Like the fig 2 describes $\boxed{6}$

It can't ~~defe~~ identify it as 0 or 6 in this case if we even keep the distance between them in the loop.

(15) Mechanics of Machine Learning

→ To tackle the above Handwritten problem we have to use different kind of approch.

→ from childhood to adulthood we often learn things and we learn what is wrong and we will learn from it and train from it.

→ We learn how to multiply numbers, solve Equations, take derivates by set of instructions.

→ We learned to recognize a dog by being shown multiple Examples and being corrected when we made the wrong guess. if our Parents say we are wrong we will modify our model to incoperate this new information.

→ Deep Learning is a Subset of a more general field of AI called ML. which is predicted on this idea of learning.

→ We give model to train and learn instead of set of large instructions.

Lets define our model to be a function $h(x, \theta)$.

The Input(x) is an Example Expressed in Vector form..

$\theta \rightarrow$ vector of the parameter that model use.

Let take an Example: ~~Sleep~~ Predict Exam Performace on ~~Sleep~~ no of hours of Sleep.

We collect a lot of data for each data Point $x = [x_1 \ x_2]^T$, $x_1 \rightarrow$ no of hours of sleep.

$x_2 \rightarrow$ no of hours we ~~spent~~ studying.

Our goal, the model $h(x, \theta)$ with Parameter Vector $\theta = [\theta_0 \ \theta_1 \ \theta_2]^T$

Such that:

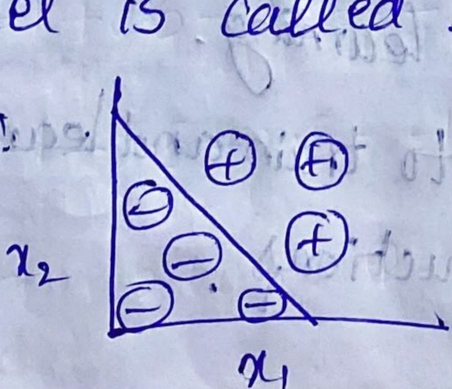
$$h(x, \theta) = \begin{cases} -1 & \text{if } x^T \cdot \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} + \theta_0 < 0 \\ 1 & \text{if } x^T \cdot \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} + \theta_0 \geq 0 \end{cases}$$

-1: bellow average

1: Otherwise.

This model is called linear Perceptron

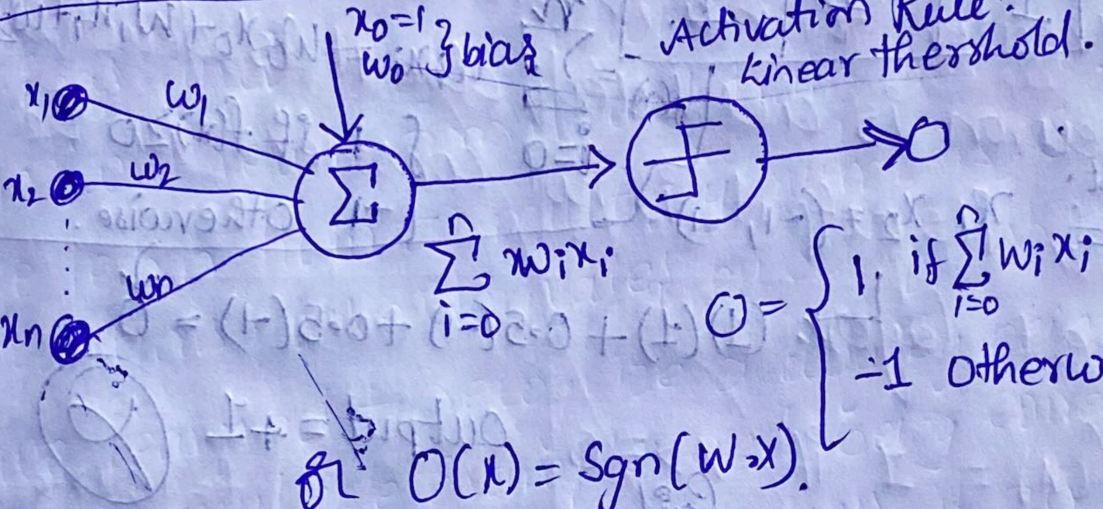
Sample data:



① Linear perceptron as Neurons & limitations:

A linear perceptron or perceptron is a ML algorithm for supervised learning of various binary classification tasks.

The single layer neural network has four parameters: input values, weights, Bias, netsum and an activation function.



$$\sum_{i=0}^{n} w_i x_i$$

$$O = \begin{cases} 1 & \text{if } \sum_{i=0}^{n} w_i x_i \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

or $O(x) = \text{sgn}(w \cdot x)$.

It is very esay to show that our linear perceptron & the neuronal model are perfectly equivalent. And it is quite simple to show that singular neurons are strictly more expressive as linear perceptron.

Trying the Linear Perceptron by using

and gate:

	x_1	x_2	y
1	-1	-1	-1
2	1	-1	-1
3	-1	1	-1
4	1	1	1

Let us take

$W_0 = 1$ (bias) $x_0 = 1$

$w_1, w_2 = 0.5$ (assum)

We know formula:

$$f(x) = \sum_{i=0}^{n} w_i x_i = W_0 x_0 + W_1 x_1 + W_2 x_2$$

$$\begin{cases} 1 & \text{if } f(x) \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

1st: $x_1\ x_2 = (-1, -1)$

$\hookrightarrow (1)(1) + 0.5(-1) + 0.5(-1) = 0$

output $= +1$ X

fail

So change the bias value.

$\boxed{W_0 = -1}$ $\boxed{W_1, W_2 = 0.5}$ $\boxed{x_0 = 1}$

$P(-1,-1) \rightarrow -1 - 0.5 - 0.5 = -2 < 0$ $\boxed{\text{output} = -1}$ ✓

$P(1,-1) \rightarrow -1 + \cancel{0.5} - \cancel{0.5} = -1 < 0$ $\boxed{\text{output} = -1}$

$P(-1,1) \rightarrow -1 - \cancel{0.5} + \cancel{0.5} = -1 < 0$ $\boxed{\text{output} = -1}$

$P(1,1) \rightarrow -1 + 0.5 + 0.5 = 0 = 0$ $\boxed{\text{output} = 1}$

Limitations:-

1) Lack of complexity: They are incaple of handling complex patterns

2) Limited Expressiveness:

3) No feature Hierarchy: due to linear nature they cannot form such hireraches.
(Simple to complex)

4) Inadequate for Complex tasks:

5) Overfitting to linear Assumptions:

Activation functions:

The Activation function is one that outputs a smaller value for tiny inputs and a higher value if it's inputs are greater than threshold.

There are two type of activation functions:

1) linear Activation function.

2) Non-linear Activation function.

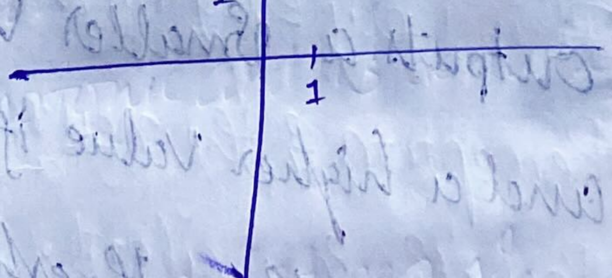
Linear Activation functions:

The function is linear.

*) Binary Step Function:

If the input to the activation function is greater than a threshold then the neuron is activated, else it is deactivated. i.e., its output is not considerd for the next layer.

$$f(x) = 1 \text{ if } x >= 0$$
$$\quad\;\; 0 \text{ if } x < 0.$$



drawbacks:

*) It can be only used while creating a binary classifier

*) This function is not useful when there are multiple classes in the target variables.

*)

Non linear Activation functions:

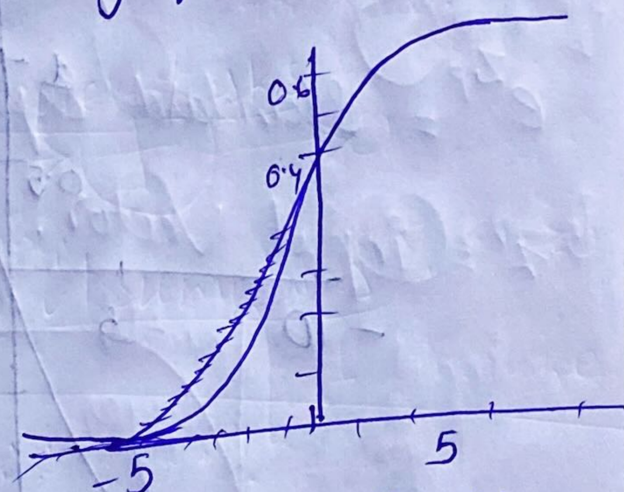
~~The input of~~ The functions will be non linear

1) Sigmoid :-

A Sigmoid function is graphed in "S" shape.

It uses the function:

$$f(z) = \frac{1}{1+e^{(-z)}}$$



*) when the logit is very small, the output of a logistic function is very close to "0".

*) when the logit is very large the output of the logistic function is close to "1".

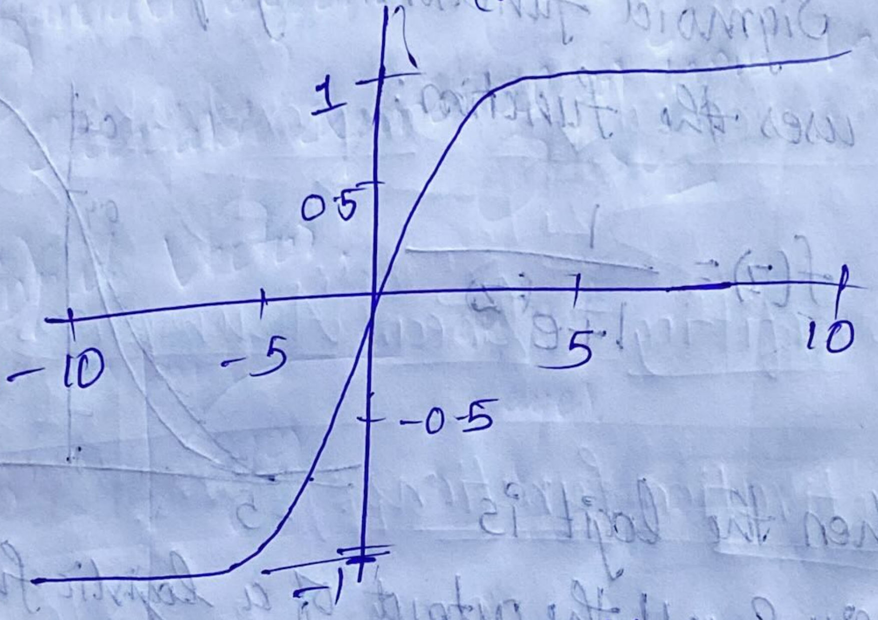
*) In b/w them, the function assumes an S-shape

② Tanh function:

The Tanh function use a Similar kind of S- shaped non- linearity, but insted ranging from 0 to 1, the output of tanh function ranges from -1 to 1.

It uses:

$$f(z) = \tanh(z) = \frac{2}{(1+e^{-2x})} - 1$$



→ The function is differentiable.

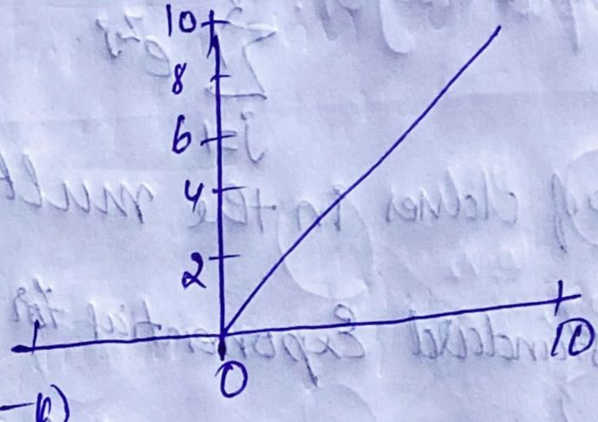
→ The function is monotonic and its derivative not.

→ It is mainly used in between 2 classes.

3) RELU activation function

It is Rectified Linear unit

It uses the function $f(z) = \max(0, z)$, resulting a characteristic hockey-stick-shaped response.



It Ranges from 0 to ∞.

The function is monotonic.

4) Softmax Function:

The Softmax Function canbe used for
→ multiclass classification problems.
→ It is described as combination of multiple Sigmoids.
→ Sigmoid returns values b/w 0 and 1, which canbe treated as probabilities of a data Point belonging to a particular class.

This function returns the probability for a data point belonging to each individual class.

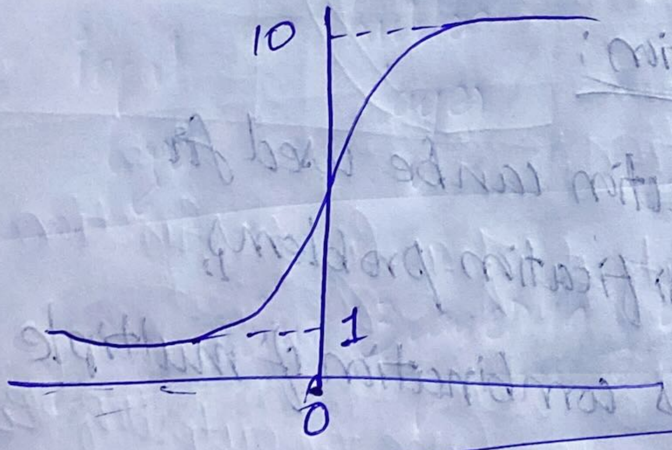
$$\sigma(\vec{z}) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

K = no of classes in the multiclass classifier

e^{z_j} = Standard Exponential for output vector

e^{z_i} = Standard Exponential for input vector.

$\vec{z}$ = input Vector.



Hyper parameters: (not important)

1) The k in KNN
2) Learning rate for training a neural Network
3) Train-test split ratio
4) Batch size
5) Number of Epochs
6) Braches in Decision Tree
7) Number of clusters in clustering algorithms

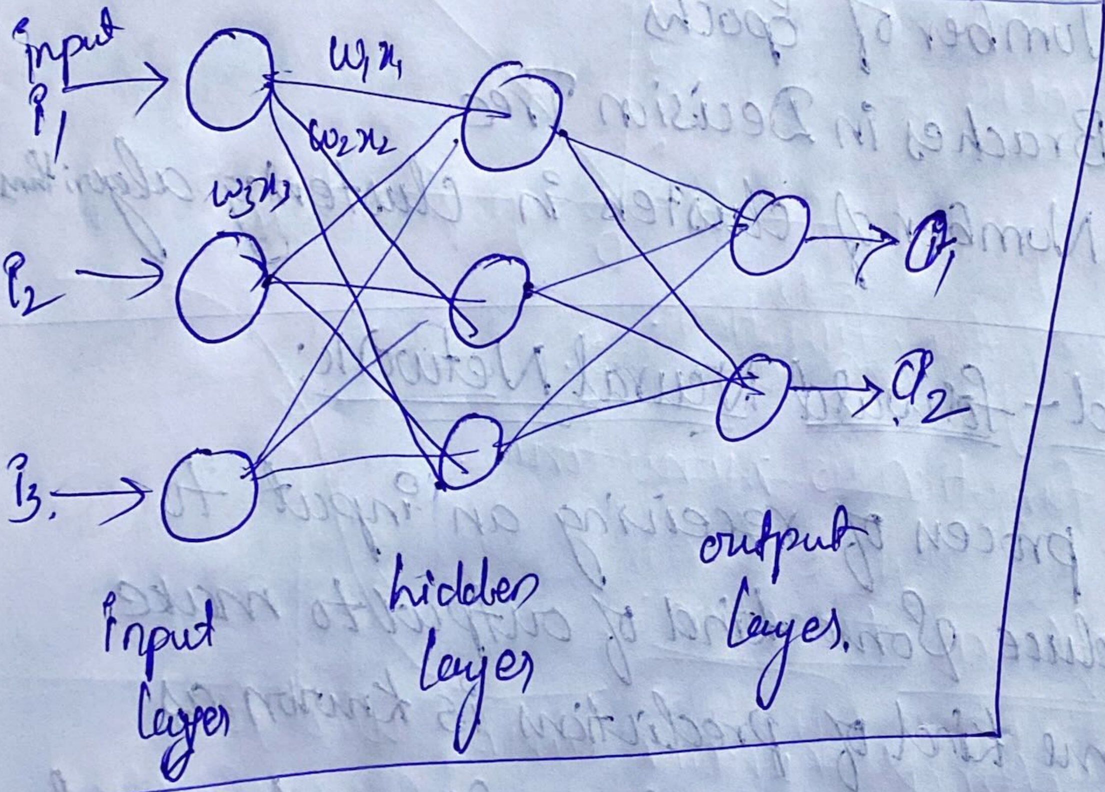
Feed-forward Neural Network:

The process of receiving an input to Produce some kind of output to make some kind of predictions is known as Feed forward. The feed-forward neural network is the core of many other imp neural network such as convolution neural network.

In the feed-forward neural network, there are not only any feedback loops or connections in the network. ~~Here is an~~ Here.

there are 3 layers:

1) Input layer
2) hidden layer
3) output layer.



(Explain about 3 layer in own.)