Day: Date:

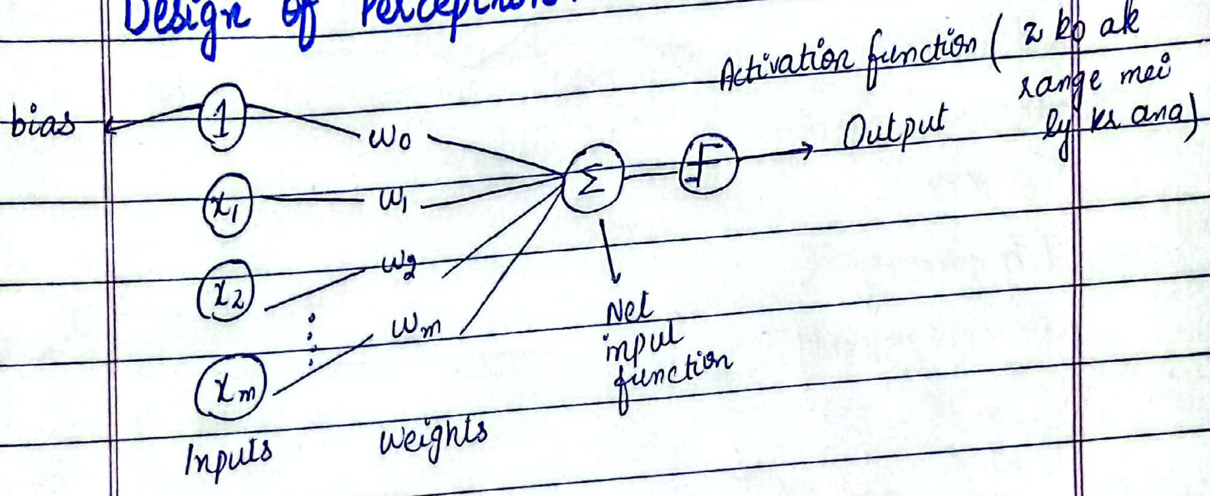
# 100 days of deep learning

## Perceptron:

↳ Perceptron is an algorithm used for supervised machine learning.

⇒ Also known as mathematical model or function model.

## Design of Perceptron:




$$\therefore z = w_1 x_1 + w_2 x_2 + b$$

net input function

Activation Function :-

$z >= 0 \longrightarrow 1$ ( output)

$z < 0 \longrightarrow 0$ ( output)

## Example:

| iq | cgpa | placement |
|---|---|---|
| 78 | 7.8 | 1 |
| 69 | 5.1 | 0 |

Stages < Training / Prediction

Day: Date:

$x1 = iq \quad x2 = cgpa$

Then, w1, w2, b --- values find = ? (During training)

Prediction :-

$w1 = 1, \; w2 = 2, \; b = 3$

$iq = 100, \; cgpa = 5.1$

$100 * 1 + 2 * 5.1 + 3 * 1 = 113.2 > 2$ (Output 1)

**Deep learing is inspired by human nervous system but slightly :**

Complex

Preprocessing

Neuroplasticity.

**Weights :**

↳ Which feature is important or which is not

↳ Strength of connections between neurons

**Bias :**

fine tuning of output

↳ It enable the model to learn even when all input features are zero.

Imagine :

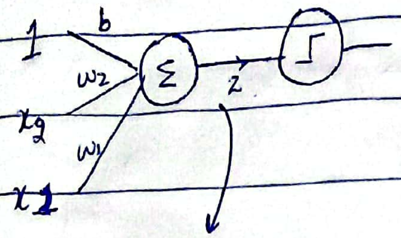
Weather — temperature, humidity → not a good day to go outside → bias helps neural networks make prediction or decision, even when all input factors are not favourable

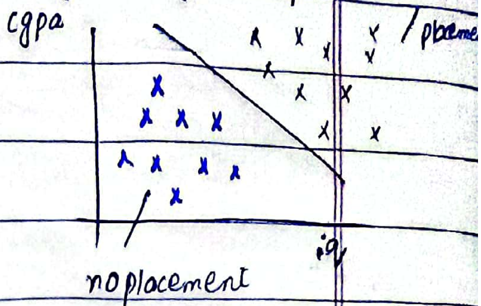
Should go outside = ?

Day:                                                                 Date:

# Geometric Intuition:

iq    cgpa    placed




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

binary classification

$\boxed{y = 1, 0}$ ↙ output

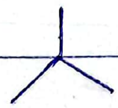
$$y = f(z) = \begin{cases} 1 & z \geq 0 \\ 0 & z < 0 \end{cases}$$

$w_1 = A$ , $w_2 = B$ , $b = c$

$x_1 \Rightarrow x$ , $x_2 \Rightarrow y$

$\boxed{Ax + By + c = 0}$ ↙ equation of line

line → 2D
→ 3D →

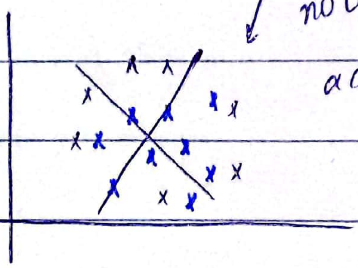


4D

$$f(z) = w_1 x_1 + w_2 x_2 + w_3 x_3 + b \geq 0$$

$$Ax + By + Cz + b \geq 0$$

# Limitation:



no linear accuracy!

Perception only works in linear / sort of linear

→ seperate with only one line.

Sou of linear

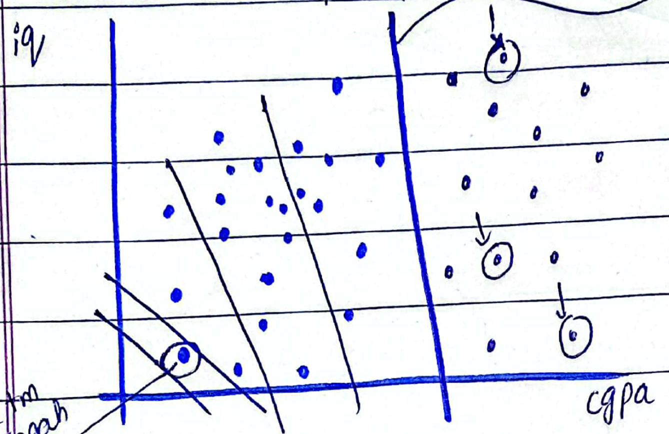
└→ AK green point → red region
AK red point → green region

accuracy ↓
but OK.

## Perceptron trick:

How to train perceptron : = ?

$Ax + By + C = 0$




kya tum apni jagah py ho.

loop → 1000 → also known as epochs

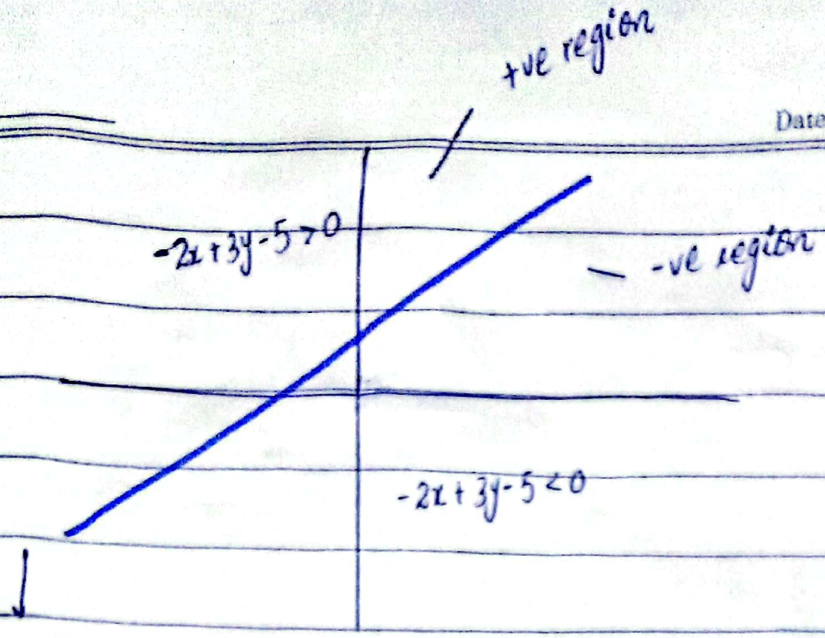
└→ random student select

apray region mei ho then next move line (no updation in $w_1, w_2, b$).

Also use : while loop → converge ( correct classify).

## How to identify positive region and negative region : ?

Day: Date:




Find which label exist in which region =?

## Transformation:

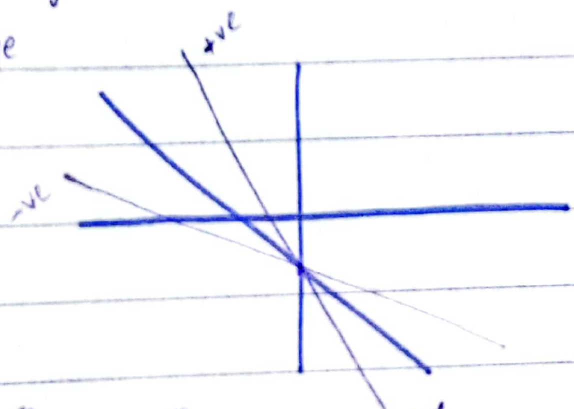
How to move line in 2D environment =?

$Ax + By + C = 0$

change.

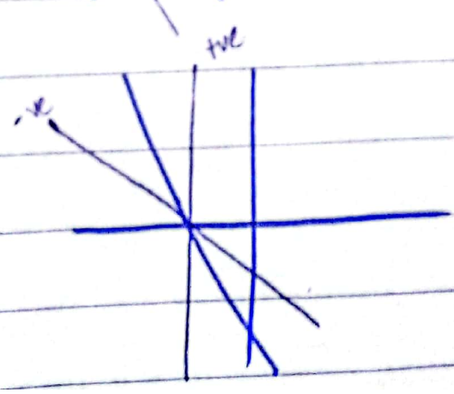
$Ax + By + c = 0$

change



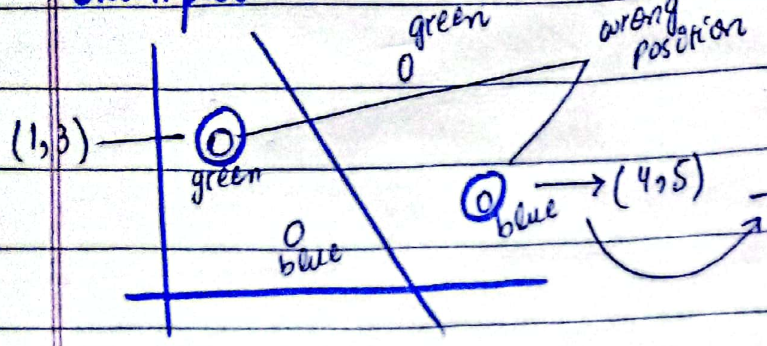

$Ax + By + c = 0$

change



Day: Date:

## Example:




$$
\begin{array}{rrr}
 & 2 & 3 & 5 \\
- & 4 & 5 & 1 \\
\hline
\end{array}
$$
$-2x - 2y + 4 = 0$

$$
\begin{array}{rrr}
 & 2 & 3 & 5 \\
+ & 1 & 3 & 1 \\
\hline
\end{array}
$$
$3x + 6y + 6$

itna bara jump not possible → Use learning rate.

**coef = coef - learning rate * coordinates**

## Algorithm:

| $x_0$ | $x_1$ | $x_2$ | $y$ |
|---|---|---|---|
| | cgpa | iq | placed |
| 1 | 7.5 | 61 | 1 |
| 1 | 8.9 | 109 | 1 |
| 1 | 7.0 | 81 | 0 |

$Ax + By + c = 0$

$w_0 + w_1 x_1 + w_2 x_2 = 0$

$w_0 = c,\ w_1 = A,\ w_2 = B$

$w_0 x_0 + w_1 x_1 + w_2 x_2 = 0$

$\sum_{i=0}^{2} w_i x_i = 0$

$\therefore > 0 \rightarrow 1$

$\therefore < 0 \rightarrow 0$

Scanned with CamScanner

Day: Date:

epoch → 1000, $\eta = 0.01$

for i in range (epochs):

randomly select a student

if $X_i \in N$ and $\sum_{i=0}^{2} w_i x_i \geq 0$ — but ~~predict~~ lies in positive region.

$W_{new} = W_{old} - \eta X_i$

if $X_i \in P$ and $\sum_{i=0}^{2} w_i x_i < 0$

$W_{new} = W_{old} + \eta x_i$

actual in negative place

## Simplified Algorithm:

for i in 1000

randomly select a student

$$w_n = w_0 + \eta \underbrace{(y_i - \hat{y}_i)}_{0} x_i$$

$w_n = w_0$ ] — correct position

$$w_n = w_0 + \eta \underbrace{(y_i - \hat{y}_i)}_{-1} x_i$$

$\eta$ → step size or adjustment

learning rate

| $y_i$ | $\hat{y}_i$ | $y_i - \hat{y}_i$ |
|---|---|---|
| 1 | 1 | 0 |
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | -1. |

How big of steps to take while adjusting itself to make better predictions.

steps ↓ → long time to learn

steps ↑ → overshoot →

~~make mistakes~~

Scanned with CamScanner

Day: Date:

## Problem with Perceptron trick:

How much good result =?

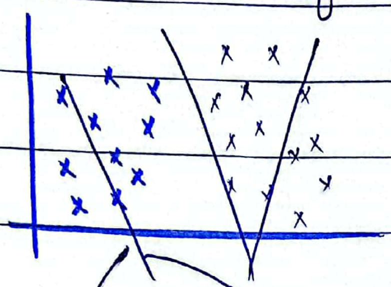
wahi best line hai ??

bar bar ak point select hona ( tu line move hi ni karein gi w1 w2 and b update hi ni ho gay)

**We use loss function instead of trick.**

Loss function

└ number of misclassified points.



$w_1, w_2, b$ → give a number.

$f(w_1\ w_2\ b)$

└→ 25 > error

23

23 < 25

line better

loss minimum

⇒ loss function is a method to tell how much model analyze or predict the output.

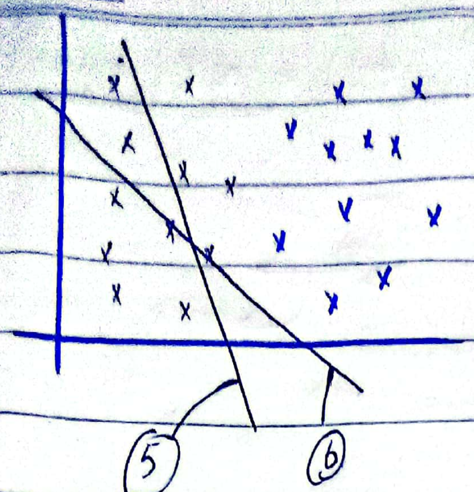
**Loss can detect:**

→ Misclassified points

→ Distance

Scanned with CamScanner

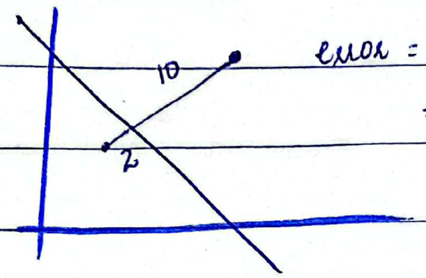
Day: Date:




misclassified points.

misclassified points $\uparrow$ error.

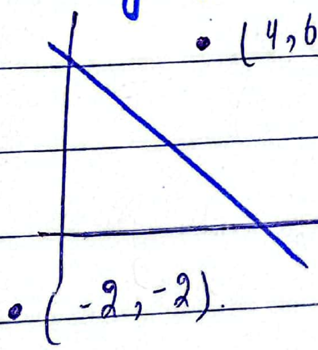
# distance




error = 10 + 2

= 12.

## Actually perceptron loss detect Method:



• (4, 6) $= 2(4) + 3(6) + 4 = 30$

dot product.

$2(-2) + 3(-2) + 4 = |-6| = 6$

• (-2, -2).

$\underbrace{30 + 6 = 36.}_{\text{Error.}}$

## Perceptron Loss function Formula:

Day: Regularization
Date:

$$L(w_1, w_2, b) = \frac{1}{n}\sum_{i=1}^{n} (y_i, f(x_i)) + \underline{\alpha R(w_1, w_2)}$$

$$L(y_i, f(x_i)) = \max(0, -y_i f(x_i))$$

$f(x_i) \rightarrow w_1 x_1 + w_2 x_2 + b$

prediction

placement = ?

$z$

Simplest form:

$$L = \frac{1}{n}\sum_{i=1}^{n} \boxed{\max(0, -y_i f(x_i))}$$

hinge loss.

number of rows in data.

$$L = \frac{1}{n}\sum_{i=1}^{n} \max(0, -y_i f(x_i)).$$

where $f(x_i) = w_1 x_{i1} + w_2 x_{i2} + b$

number of rows

| | | | |
|---|---|---|---|
| 1 | $x_1$ | $x_2$ | $y$ |
| 2 | $x_{11}$ | $x_{12}$ | $y_1$ |
| 3 | $x_{21}$ | $x_{22}$ | $y_2$ |
| ⋮ | | | |

$x_{ij}$

rows columns.

$$\max(0, \underbrace{-y_i f(x_i)}_{x}) = \max(0, x)$$

$x \geq 0 = x$

$x \leq 0 = 0$

$-y_i f(x_i) \geq 0$ = value

$< 0 = 0$

$$L = \frac{1}{2}\left[\max(0, -y_1 f(x_1)) + \max(0, -y_2 f(x_2))\right]$$

2 point

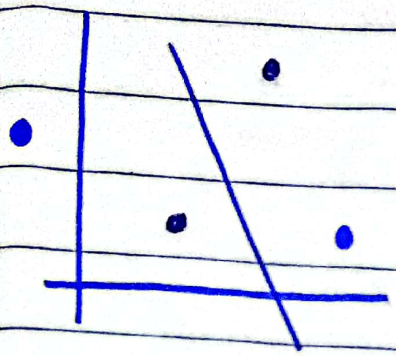
breakdown

$x_1, x_2$

Scanned with CamScanner

Day:
Date:

# Geometric intuition:



| cgpa | iq | placed |
|---|---|---|
| 7 | 8 | 1 |
| 6 | 8 | -1 |
| 4 | 2 | 1 |
| 1 | 1 | -1 |

| $y_i$ | $\hat{y}_i$ |
|---|---|
| 1 | 1 |
| -1 | -1 |
| 1 | -1 |
| -1 | 1 |

4 types.

1) $\max(0, -y_i f(x_i))$ 1

$w_1 x_1 + w_2 x_2 + b \geq 0 \rightarrow +ve$

1 $f(x_i) = +ve$

+ve +ve = -ve ← $-(y_i)$

$\max(0, -ve) = 0$

loss (zero).

2) $y_i$ $f(x_i)$

-1 -ve = +ve ← $-(y_i)$ = -ve

$\max(0, -ve) = 0$

loss(zero).

⇒ loss mei zero contribution.

Day: Date:

3): $y_i$ $f(x_i)$

+ve -ve = -ve = +ve

max( 0, +ve) = +ve.

4): $y_i$ $f(x_i)$

-ve +ve = -ve = +ve

max( 0, +ve) = +ve.

⇒ misclassified point → loss contribute.

$w_1 w_2 b = ?$

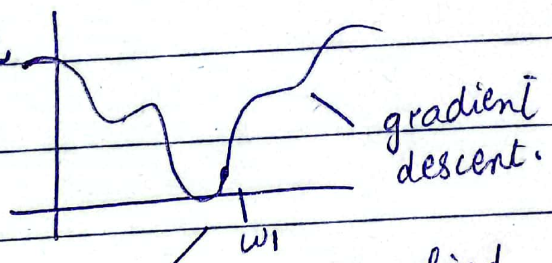
↓ aisi value findout krni hai = ?

Using gradient descent

$$L = \arg\min \boxed{\frac{1}{n} \sum_{i=1}^{n} \max(0, -y_i f(x_i))}$$ ↙ minimize

where $f(x_i) = w_1 x_{i1} + w_2 x_{i2} + b$

$L(w_1 w_2 b)$

$L(w_1)$ 


w1 ki aisi value find krni hai jis sy loss km ho jai

Day: Date:

for i in epochs:

$$w_1 = w_1 + \eta \frac{\partial L}{\partial w_1}$$ 0.0.1 → partial derivative of L with respect to w1

$$w_2 = w_2 + \eta \frac{\partial L}{\partial w_2}$$

$$b = b + \eta \frac{\partial L}{\partial b}$$

$$\left[ \frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}, \frac{\partial L}{\partial b} \right] = ?$$

**Loss Function diffeerientiation:**

$$L = \frac{1}{n} \sum_{i=1}^{n} \max(0, -y_i f(x_i))$$

where $f(x_i) = w_1 x_{i1} + w_2 x_{i2} + b$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial f(x_i)} \times \frac{\partial f(x_i)}{\partial w_i}$$

$$\begin{cases} 0 & \text{if } y_i f(x_i) \geq 0 \\ -y_i & \text{if } y_i f(x_i) < 0 \end{cases}$$

$$\frac{\partial f(x_i)}{\partial w_i} = x_{i1}$$

$$\frac{\partial L}{\partial w_1} = \begin{cases} 0 & \text{if } y_i f(x_i) \geq 0 \\ -y_i x_{i1} & \text{if } y_i f(x_i) < 0 \end{cases}$$

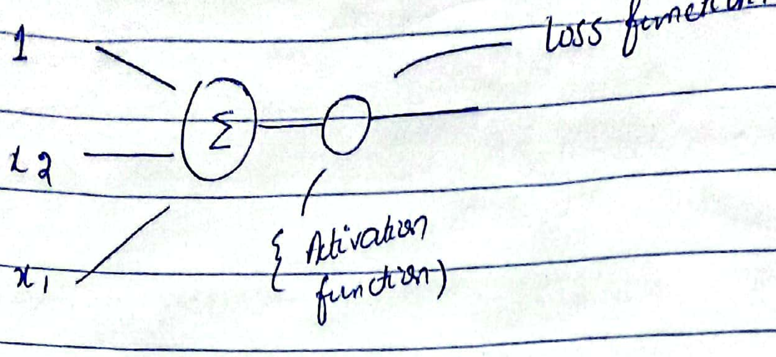
$$\frac{\partial L}{\partial w_2} = \begin{cases} 0 & \text{if } y_i f(x_i) > 0 \\ -y_i x_{i2} & \text{if } y_i f(x_i) < 0 \end{cases}$$

$$\frac{\partial L}{\partial b} = \begin{cases} 0 & \text{if } y_i f(x_i) \geq 0 \\ y_i & \text{if } y_i f(x_i) < 0 \end{cases}$$

Day:      Date:

## More loss function:

mathematical model.




Step function ⟶ sigmoid function

Probability ( 1, -1)

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

loss function

↳ binary cross Entropy

$$L = -y_i \log \hat{y_i} + (1-y_i) \log 1-\hat{y_i}$$

Gradient descent ↳ SGD ⟶ output probability.

Perceptron == Logistic Regression

↳ Activation function ; loss function

Sigmoid    binary cross Entropy.

Perceptron ⟶ flexibility.

## Multiclass:

Activation    loss function

Softmax    categorical cross entropy

$$6 = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \qquad L = \sum_{j=1}^{H} y_j \log(\hat{y_j})$$

softmax Regression.

$$6 = \frac{e^{z_1}}{e^{z_1}+e^{z_2}+e^{z_3}}$$

Day:
Date:

Perception == Regression.

Activativation $\longrightarrow$ linear

$z \geq 0 \quad 1$

$z < 0 \quad -1$

loss function

$\downarrow$

mse $(y - \hat{y}_i)$.

Perceptron is flexible by design. Just change Activation function and loss function.

| Loss function | Activation | Output |
|---|---|---|
| Hinge Loss | step | perceptron $\rightarrow$ binary classification -1, 1 |
| log-loss (binary cross Entropy | Sigmoid | logistic regression 0 - 1 $\rightarrow$ binary classification |
| Categorical cross Entropy | softmax | Softmax Regression probability $\rightarrow$ multi class classification |
| MSE | linear | linear regression $\rightarrow$ number. |

Problem with Perceptron:

└→ only works on linear data.

2. Tensor flow playground

Not works in XOR data.