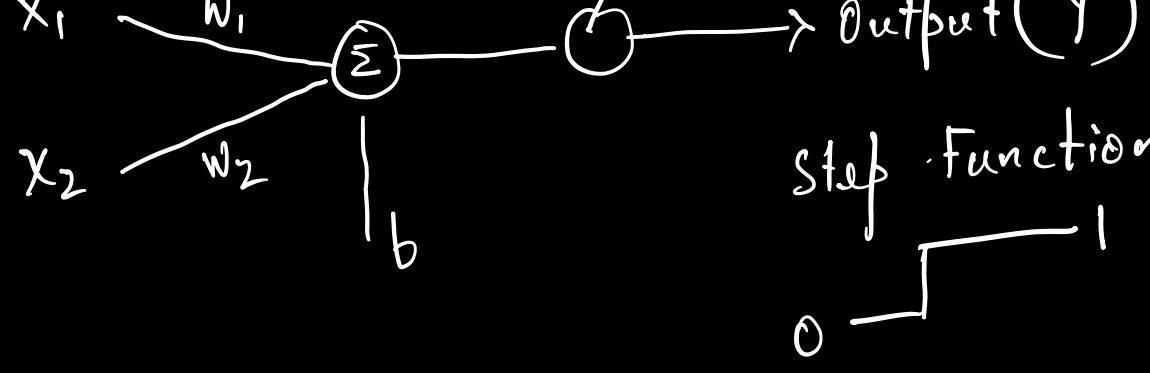
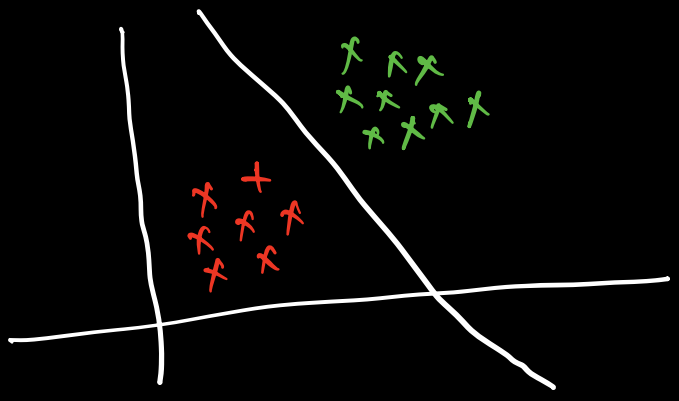
Deep Learning (2)

Neuron / Perceptron

Mathematical Func



Step Function



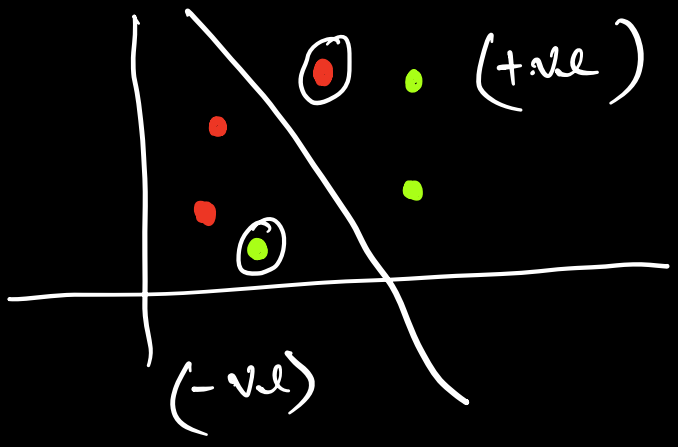
2D - Line
3D - Plane
4D or more - Hyperplane

x_1 - 10th
x_2 - 12th
y

Misclassification Problem

$$z = x_1 w_1 + x_2 w_2 + b$$
$$= Ax_1 + Bx_2 + C$$

parameters = (A, B, C) → changes for the line.

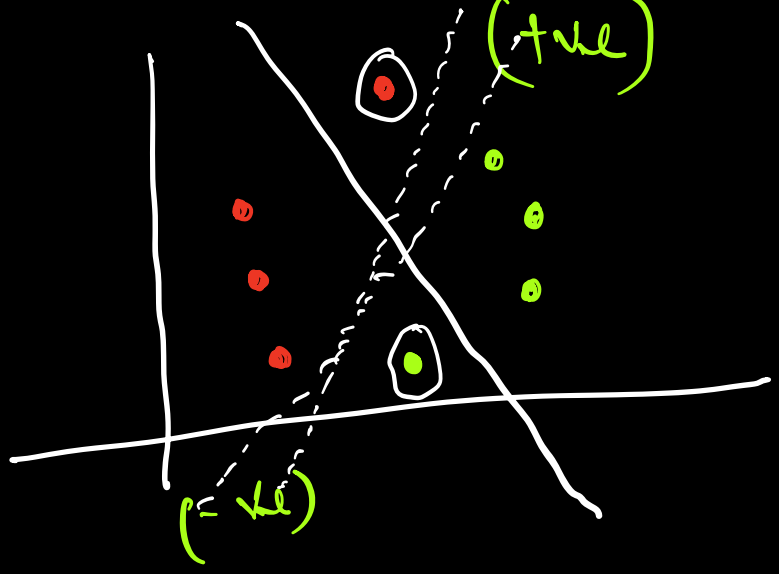


Binary Classificatio

Misclassification Problem

Calculate the errores

1) Errore Function / Loss function
   - ↓ single sample
   - ↓ entire dataset



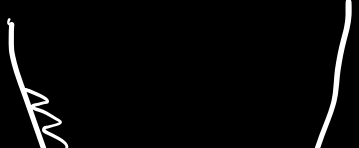
No of errores = 2

$f(w_1, w_2) = 27$

Best Fit line with the least loss = as close to 0

$f(w_1, w_2) = 16$

$f(w_1, w_2) = 18$

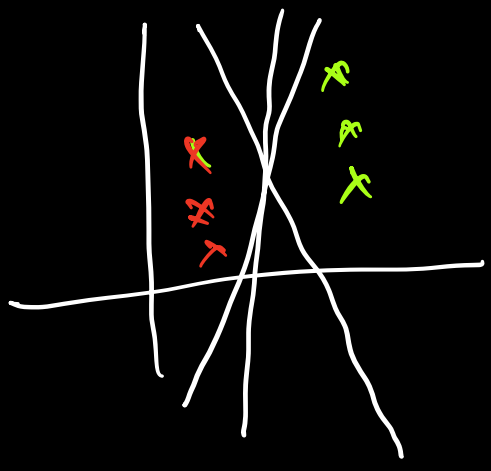


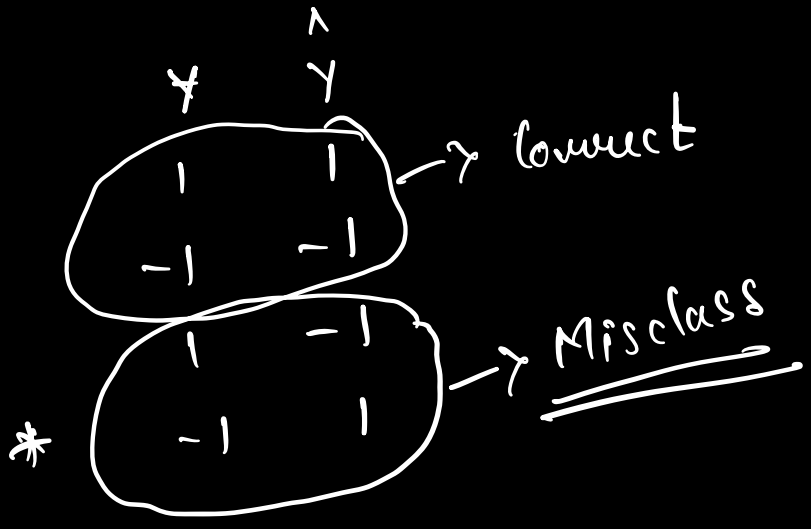
$f(w_1, w_2, b)$

Problem

if Misclassification

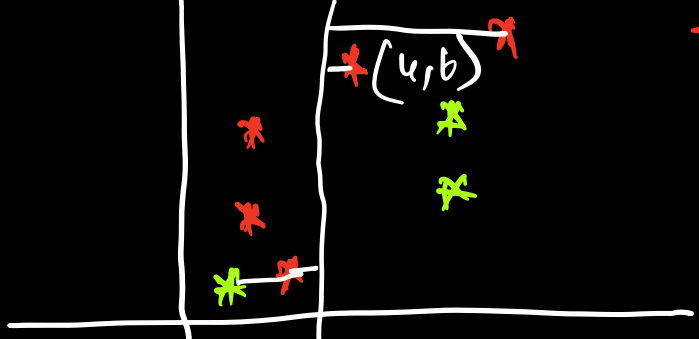
(A, B, C)





Building Approaches

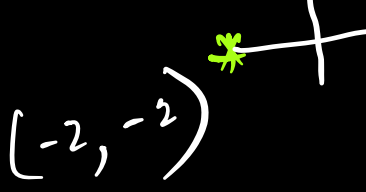
1) Total no of misclassified points
2) Magnitude and distance from the line



$V = 4$

move E

less E



I have to make some changes in line transformation (A, B, C)

3)

$2x + 3y + 5 = 0$

$2(-2) + 3(-2) + 5 = 0$

$-4 - 6 + 5 =$

$-10 + 5$

$= -5$

$2x+3y+5=0$

$2(4)+3(6)+5$

$= 8 + 18 + 5$

$= 18 + 13$

$= 31$

$= 31 + (-5)$

$= 31 - 5$

$= 26$

Loss Functions Perceptron

SGD :- $$E(w,b) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) + \alpha R(w)$$ → Not Required

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

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

$$\left\{ \begin{array}{l} x_i = Ax_i + Bx_2 + C \\ x_i = 2 \end{array} \right\} \; W_1 x_1 + W_2 x_2 + b$$

Rewriting

$$\text{Loss Func} = \frac{1}{n} \sum_{i=1}^{n} \max(0, -y f(x_i))$$

Perceptron

n = rows

x_1	x_2	(Y)
10	12	Admit

Intution :-

$$\max(0, -y f(x_i))$$

$$\left\{ \begin{array}{l} f \geq 0, 1 \\ f < 0, 0 \end{array} \right\}$$ if 0 when the point is correctly classified

Step function else $-y f(x_i)$ when misclassification will take place

Loss Function :-

$$L = \left(\underset{w_1, w_2, b}{\text{argmin}} \right) \frac{1}{n} \sum_{i=1}^{n} \max\left(0, -y f(x_i)\right)$$

Example :-

x_1	x_2	y
x_{11}	x_{12}	y_1
x_{21}	x_{22}	y_2

x_1 → row1, x_2 → row2

x_{ij} : i → row, j → column

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

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

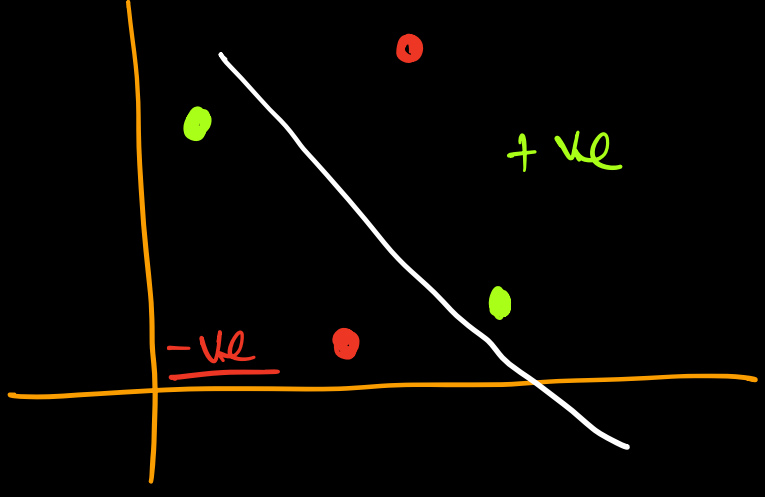
$x < 0$ → correctly classified

or

$x \geq 0$ $(-y f(x_i))$ → misclassified

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

10	12	Admit
68	72	1
82	92	-1
70	72	1
71	91	-1



y	$\hat{y}$
1	1
-1	-1
1	-1
-1	1

-ve -ve
$= -(+)$
$= -$

(+) (-ve)
$= - -$ $= (+)$

(-) (+)
$= (-)(-)$
$= +$

if it is correctly classified.

$\max(0, -)$

Not contributing towards loss.

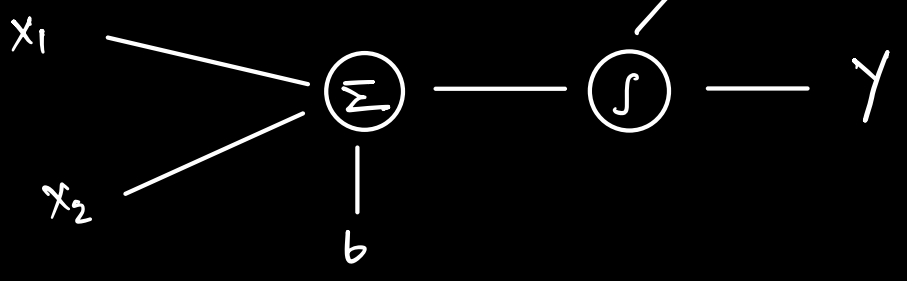
y	$f(x_i)$
+ve	+ve

$= (-)(+)$
$= (-)$

$\max(0, x)$

if $x < 0$ then it is 0

Moves with loss functions



List of A.F

1) Step Function

2) Sigmoid (0, 1)

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

3) Softmax

$$\text{Softmax} = \frac{e^{z_i}}{\sum_{j=1}^{K} e}$$

$K = 3$

$$\text{Class 1} = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

4) Linear

$$z = z$$

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

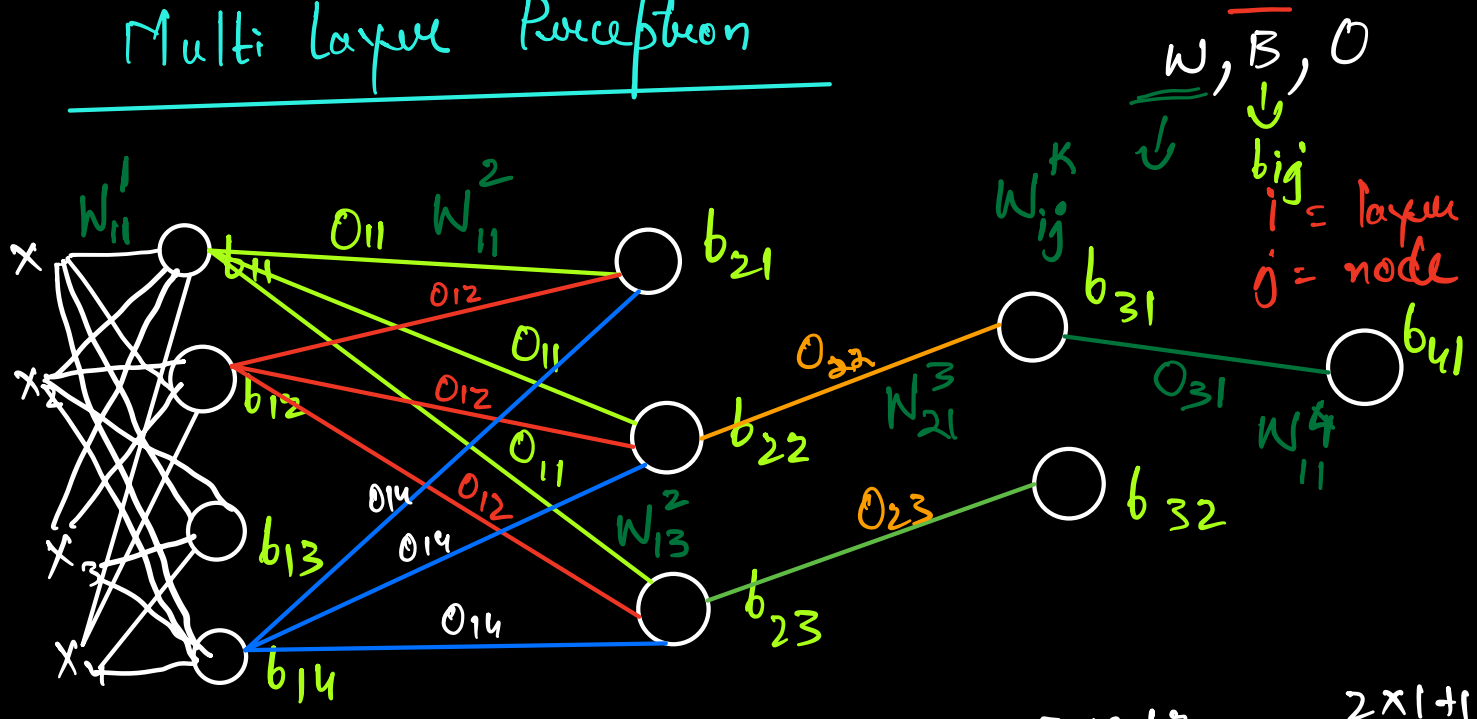
$$\text{MAE } (Y - \hat{Y})^2 \longleftarrow \text{Linear Regression} \quad \frac{Y = z}{\text{(Linear)}}$$

Activation	loss func	Usecase
Linear	MSE	Linear Regression
sigmoid	Logloss / Binary Cross Entropy	Binary Classificati
Softmax	sparse Categorical Cross Entropy	Multi class class

Problems with Perceptron

1) Works with Linear Data

Multi Layer Perceptron



Total Connect

$4 \times 4 + 4$

$= 16 + 4 = 20$

$4 \times 3 + 3$

$= 15$

$3 \times 2 + 2$

$= 6 + 2$

$= 8$

$2 \times 1 + 1$

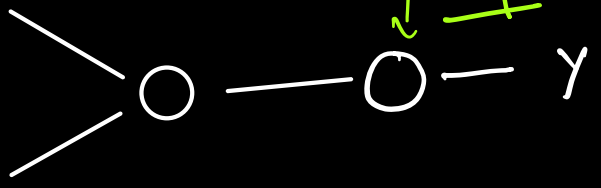
$= 3$

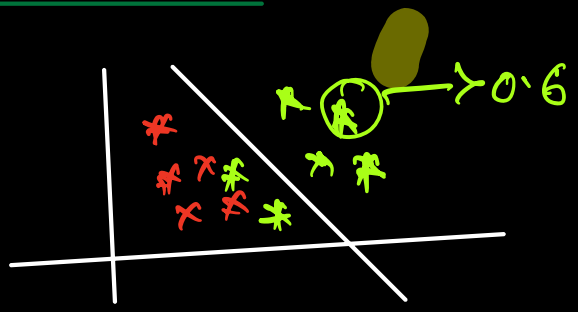
Total Parameters $= 20 + 15 + 8 + 3$

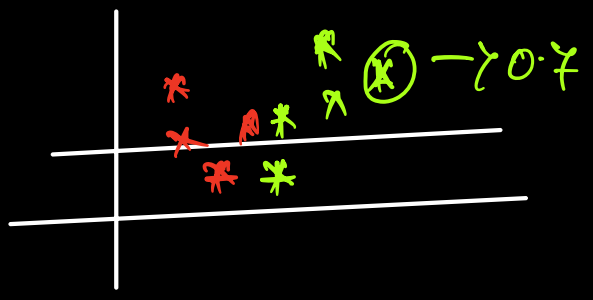
$= 46$

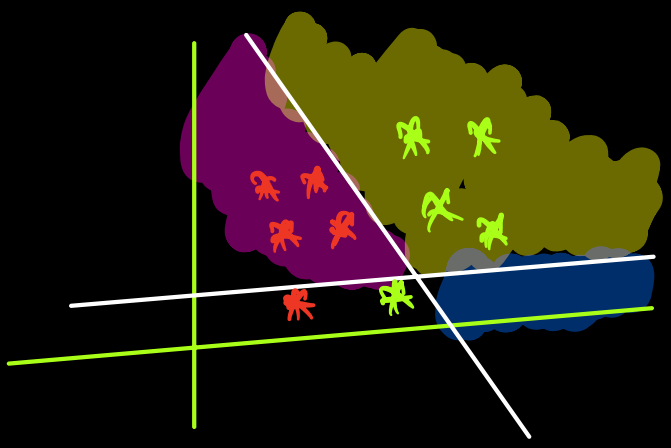
Trainable Params

Working with Non-Linear Data









0.7
+0.6
1.3

Hidden / Dense / Linear / Fully Connected