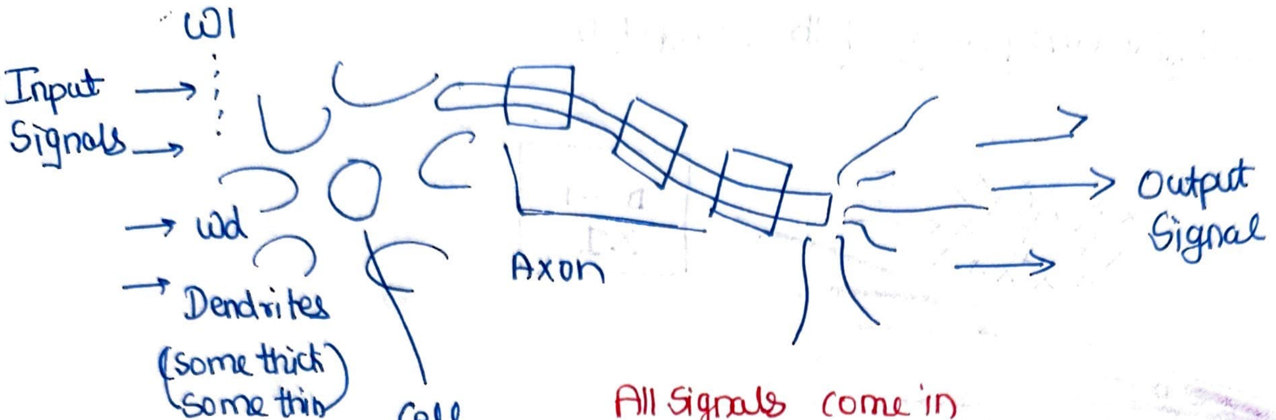
Week 3

The perceptron for Binary Classification:-



All signals come in
When there are enough signals (Threshold)
The neurons gets activated (activation)

$$Z = w_1 x_1 + w_2 x_2 + \dots w_d x_d$$

net input

Activation function

$$\varphi(z) = \begin{cases} 1 & z \geq \theta \text{ threshold} \\ 0 & z < \theta \end{cases}$$

Some books have (just be aware) +1 / −1

$b = -\theta$

redefine net input $Z = w_1 x_1 + \dots \dots w_d x_d + b$

minus threshold $b = -\theta$

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

Some books consider $b = w_0$ (as another Dendrite) usually 1

$X = [x_1 \dots \dots, x_d]$ $\quad Z = X \cdot W^T + B$

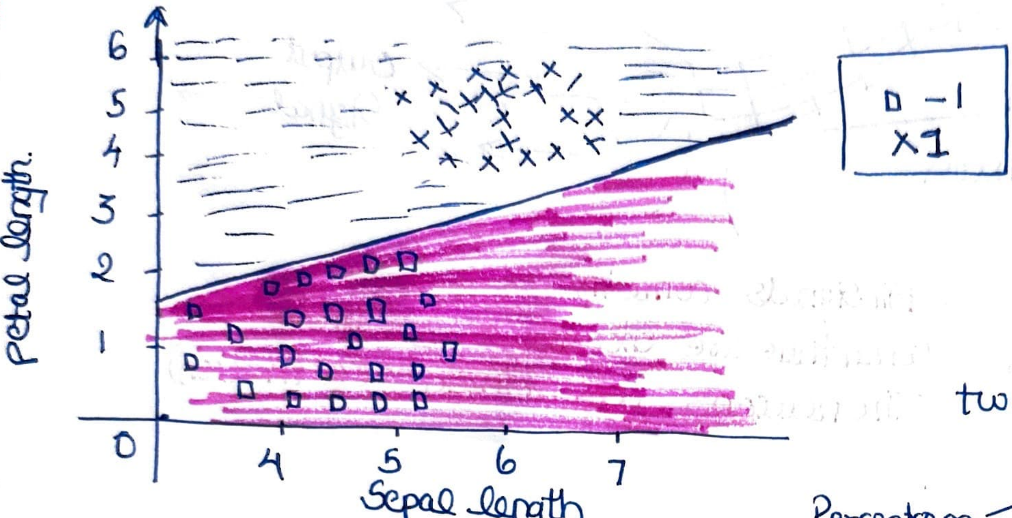
$W = [w_1 \dots \dots, w_d]$ matrix Dot Product

When we use w_0, (Just a side note)

$$Z = X' \cdot w^T.$$

$\hookrightarrow X' = [1, x_1 \dots x_d]$

* Linear Separability and power of the perceptron :-



X

two outputs

Perceptrom → 0, 1

↓ used as ↓

Binary Classifier.

given

X: Xpos, Xneg

Single lines that separates then → called linearly separable)

(in reality – hyperplane)

$$Z = x \cdot w^T + b$$

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

$Z = 0 \Rightarrow x \cdot w^T + b = 0$ eqn of a line that separates in two parts

Perceptron fully learns : X linearly separable

* The Perceptron Algorithm :-

function w = Perceptron-fit (X, y)

Initialize randomly a d-dimensional vector w and a Scalar b

error Flag = True

while error Flag do # new epoch starts here

error Flag = False

for $j = 1$ to n

$z = b + x_j w^T$

$\hat{y} = 1$ if $z > 0$, or 0 otherwise # in linear regression we use $\hat{y} = z$

if $\hat{y} \neq y_j$ then error Flag = True

- $b = b - (\hat{y} - y_j)$

for $k = 1$ to d:

- $w_k = w_k - (\hat{y} - y_j) x_{jk}$ → no learning rate ?

return w, b

X: linearly Separable

- good point → no error → do nothing
- bad point → update w, b

* infinite loop? no, we'll make appropriate update. but its possible

* if no error → no updates
* if correct & predicted → different → make update

Claim :- 1 → will eventually terminate (find a perceptron that will work 100%)

Claim 2 :- number M depends on X Such that the perceptron algorithm will make at most M errors.

random dataset

X: is it linearly separable ? | Run Perceptron | If doesn't terminate in M errors the not linearly separable (If M+1 or more)

* Extensions to multiple labels :-

C_1 : the cat v/s dog classifier, using only data in X_{cat} and X_{dog}

C_2 : the human v/s dog classifier, using only data in X_{human} & X_{dog}

C_3 : the human v/s cat classifier, using only data in X_{human} & X_{cat}.

classifiers

X : X_{cat}, X_{dog}, X_{human}

one-vs-one

assume each pair is linearly separable

	dog	cat	human
$C_1(x) =$	1	0	0
$C_2(x) =$	1	0	0
$C_3(x) =$	0	0	1
	2	0	1

↳ dog

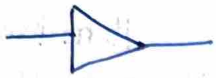
→ Situations with ties

→ we can break them further arbitrarily.

* k labels - $\frac{k(k-1)}{2}$ → number of classifiers

* Calculations with the Perceptron :

YES



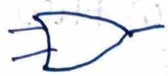
A	OUT
0	0
1	1

NOT

A	OUT
0	1
1	0

AND 

A	B	OUT
0	0	0
0	1	0
1	0	0
1	1	1

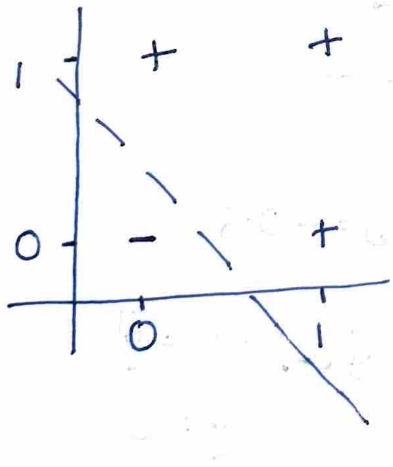
OR 

A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	1

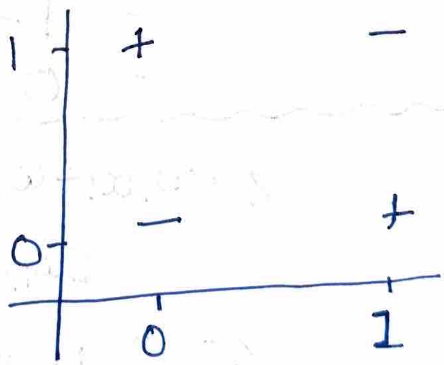
XOR 

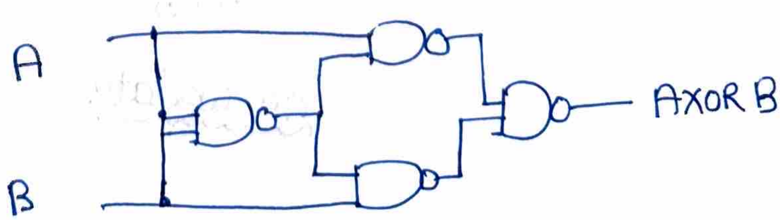
A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	0

logical OR
(linearly separable)



logical XOR
(not linearly separable)





cannot be represent
by single perceptron

* Assignment Perceptron Calculation:-

Truth table

x_1	x_2	y	
0	0	0	1
1	0	0	2
0	1	0	
1	1	1	

Initialize Perceptron (in a way it makes mistakes that will be corrected)

slope $\frac{1}{1} = 1$

$w = [1, 1]$, $b = 0.5$ | Initial Weights

1 $z = w_1 x_1 + w_2 x_2 + b$

$= 1(0) + 1(0) + 0.5$

$= 0.5$

$\varphi(z) = 1 = \hat{y}$

correct label $= y = 0$

$\hat{y} - y = 1 - 0 = 1$

$b = b - (\hat{y} - y) \cdot 1$

$b = 0.5 \, (1-0) \cdot 1$

$b = -0.5$

$w_1 = w_1 - (\hat{y} - y) \cdot 1 \quad \rightarrow x_1$

$= 1 - (1-0) \cdot 1$

$= 0$

$w_1 = 0$

$[w_1, w_2] = [0, 0]$

$w_2 = w_2 - (\hat{y} - y) x_2$

$= 1 - (1-0) \, 0$

$= 0$

$w_2 = 0$

2

$z = w_1 x_1 + w_2 x_2 + b$

$[w_1, w_2] = [0, 0]$ $\quad b = -0.5$

$z = 0 x_1 + 0 x_2 - 0.5$

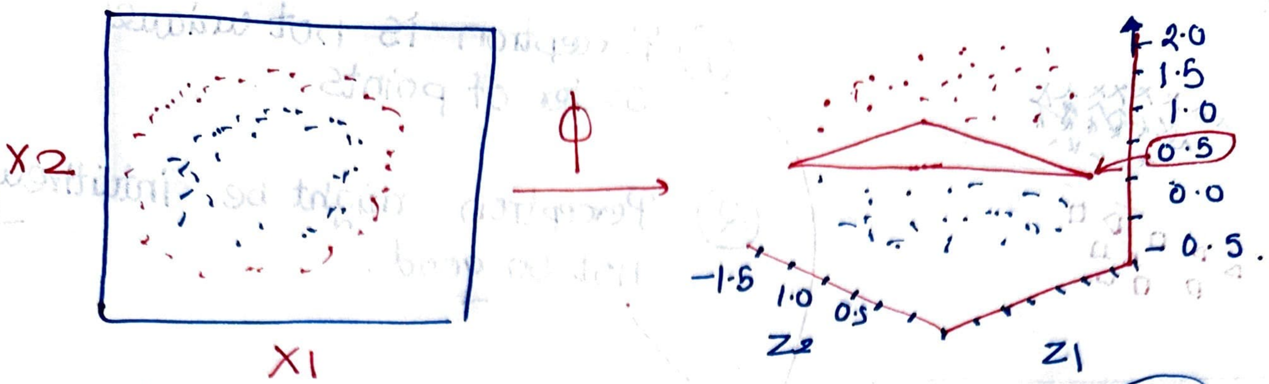
$= -0.5$

$\varphi(z) = 0 = \hat{y}$

$y = 0$

no update

Feature Engineering for linear classification :-



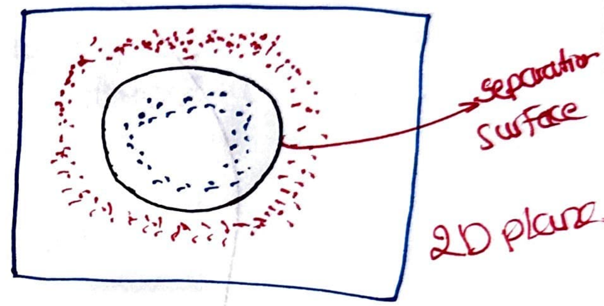
not linear
so not linearly separaple.

$\downarrow \phi^{-1}$ Inverse Transform

activation function

$$(X_1, X_2) \xrightarrow{\varphi} (X_1, X_2, \underbrace{X_1^2+X_2^2}_{}) \quad X_3 = X_1^2+X_2^2$$

add one more dimension
$X_1^2+X_2^2$
So you can separate them.



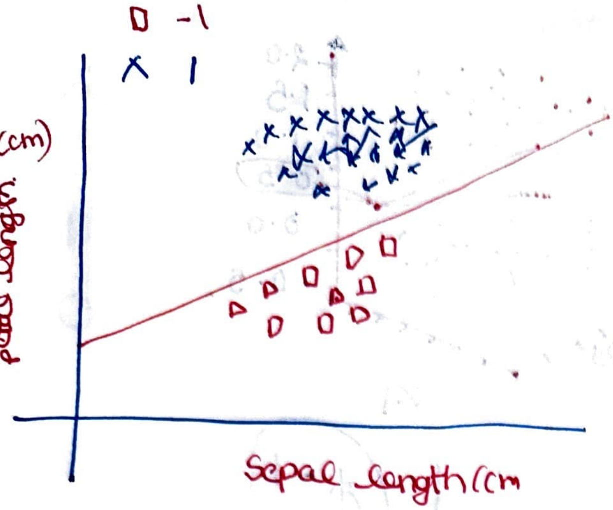
These three are now separable in a hyperplane.

$X_3 = 0.5$ all points that lie on this plane

hare x coordinate aso $\varsigma = X_1^2+X_2^2$
hyperplane separates.

now we take inverse transform and plot them again

* Perceptron Weakness :-



① Perceptron is not unique
order of points

② Perceptron might be intuitively
not so good.

→ if the dataset is already linearly
Separable

→ there won't be a unique line separating
them

→ So, our perceptron might not be the
unique one.

→ This perceptron comes too close to □ than ×
Line is close to red → preference given to Blue class.
So output might not be so good intuitively.