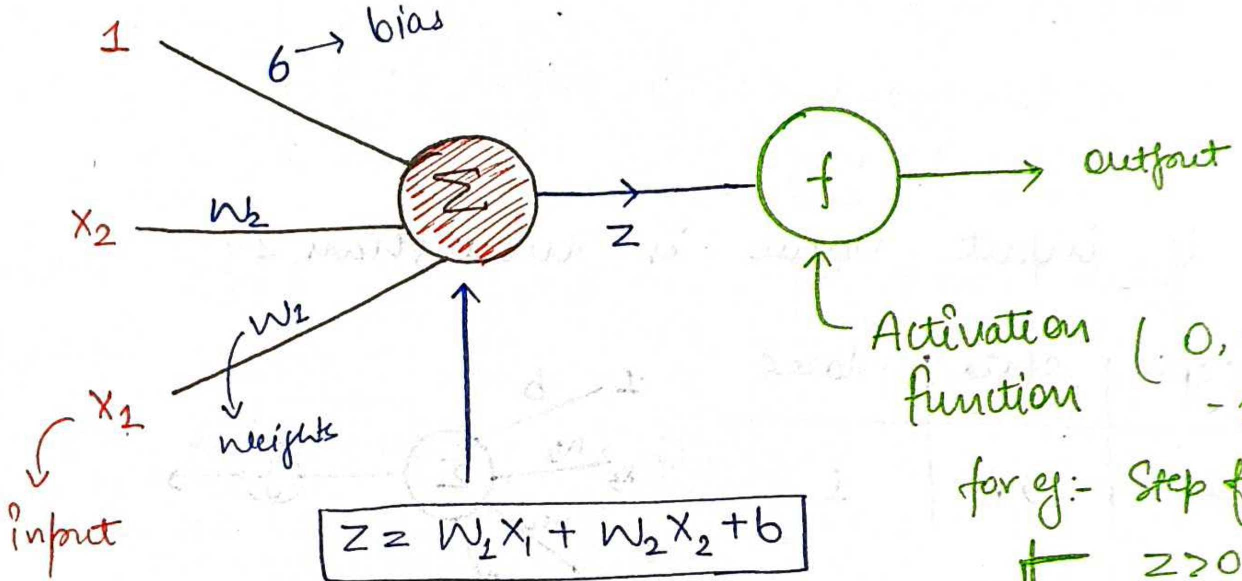
DEEP LEARNING

Perceptron

Q What is Perceptron?

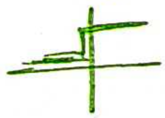
Perceptron is a algorithm $\xrightarrow{\text{like}}$ Supervised ML
- Mathematical Model
- Mathematical function



$$\boxed{z = w_1x_1 + w_2x_2 + b}$$

Activation function (0, 1 or -1, 1)

for eg:- Step function



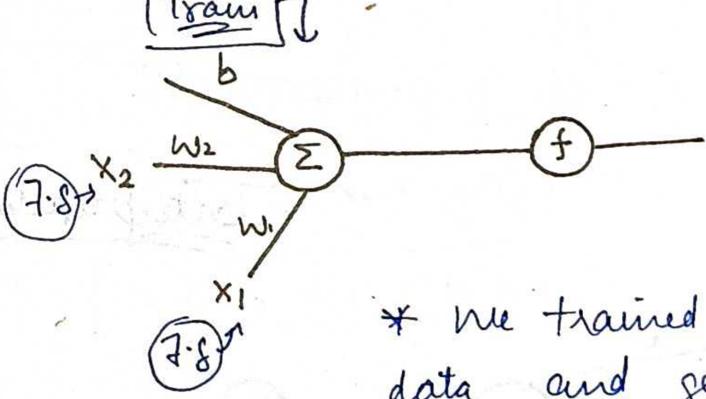
$z > 0 \rightarrow 1$

$z < 0 \rightarrow 0$

for eg: 1000 Student data

iq	cgpa	placed
78	7.8	1
69	5.1	0

Train



process

1. Train
2. Test

$x_1 \rightarrow$ iq

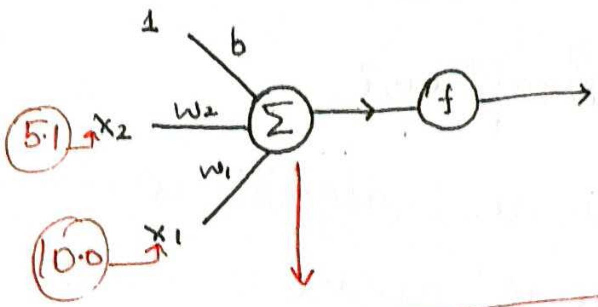
$x_2 \rightarrow$ cgpa

* we trained the data and get the value of $w_1 = 1$, $w_2 = 2$ and $b = 3$

test data

iq	cgpa
5.1	10.0

$w_1 = 1$, $w_2 = 2$, $b = 3$



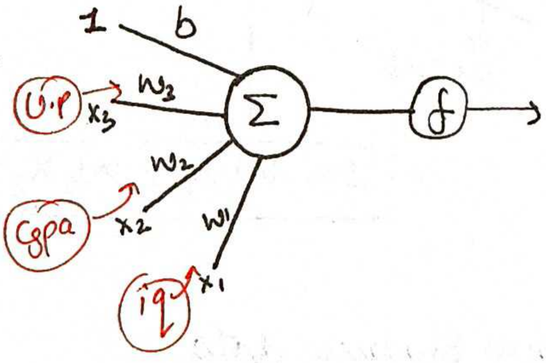
$Z > 0 \Rightarrow 1$

$$\boxed{100 \times 1 + 3.1 \times 2 + 1 \times 3 = 113.2}$$

* Z is greater than 0
  So, output is 1.
* If Z is less than 0. So, output is 0.

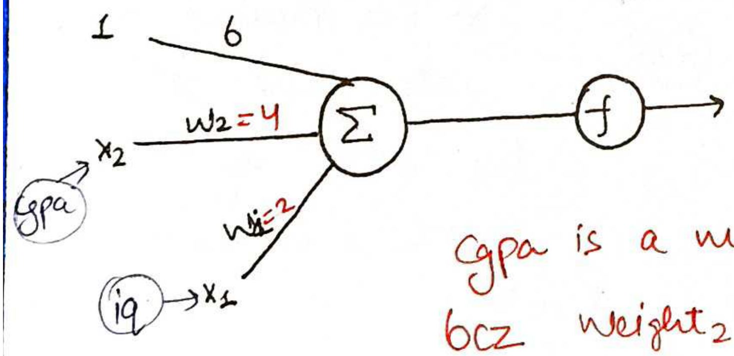
What if input value is more than 2?

iq	cgpa	State	placed
7.8	7.8	U.P	1



* No. of input ↑↑
  No. of x perceptron ↑↑

Interpretation



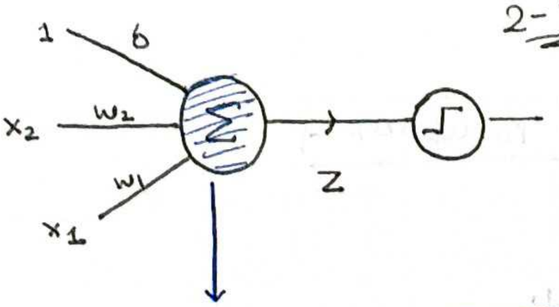
iq	cgpa	placed
78	7.8	1
69	5.1	0

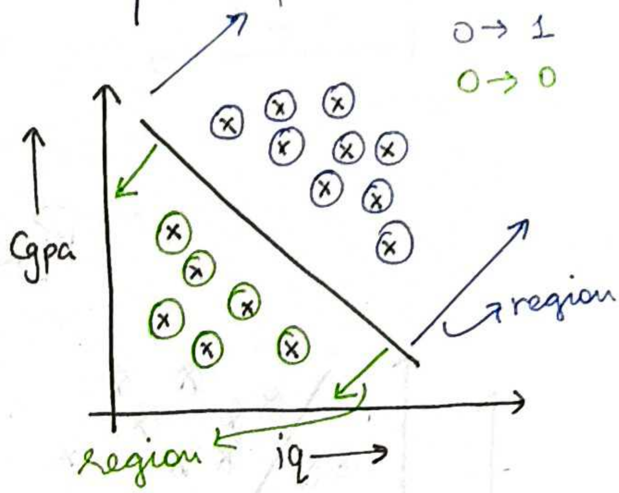
cgpa is a major factor to predict placed bcz $weight_2 = 4$ and iq is less factor to predict placed bcz $weight_1 = 2$.

Geometric Intuition

2- Input

iq	cgpa	placed





$y = 1, 0$

$$\boxed{z = w_1 x_1 + w_2 x_2 + b}$$

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

$\boxed{Ax + By + C \geq 0} \rightarrow$ region

$\boxed{Ax + By + C < 0} \rightarrow$ region

$w_1 \Rightarrow A \quad w_2 \Rightarrow B \quad b \Rightarrow C$

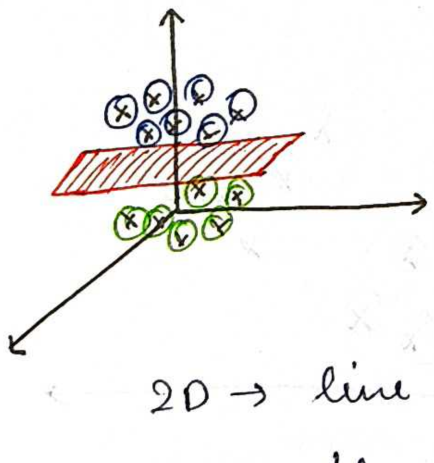
$x_1 \Rightarrow x \quad x_2 \Rightarrow y$

$\boxed{Ax + By + C = 0}$
↳ line

More than 2 Input

iq	cgpa	12th mark	placed

3 input



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

$\boxed{Ax + By + Cz + b \geq 0}$
↳ plane

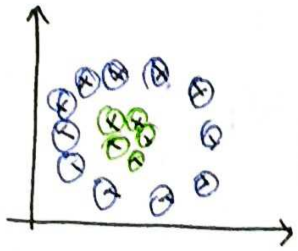
2D → line

3D → plane

4D → hyperplane

⋮

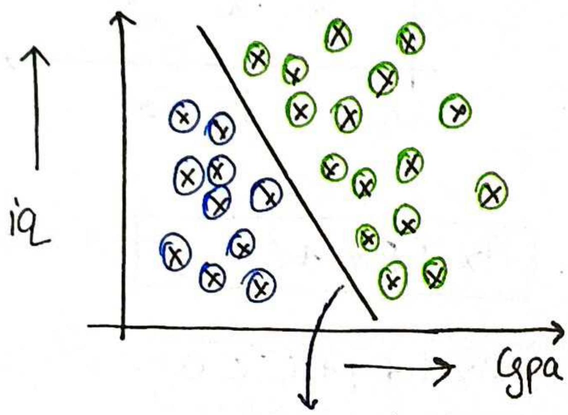
* Perceptron work only with linear / sort of linear data



⇒ Perceptron fail in this type of data.

| Code :- Perceptron 1 |

Perceptron Trick



$y = mx + b$

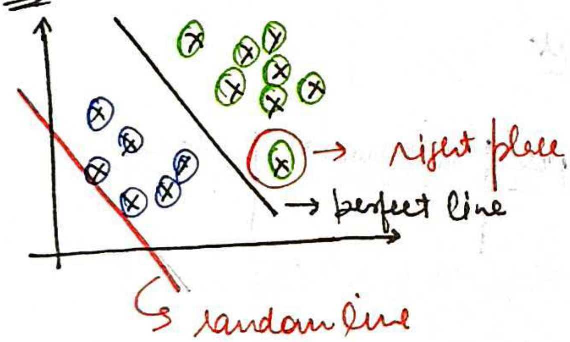
$Ax + By + C = 0$

→ We have to find A, B, C

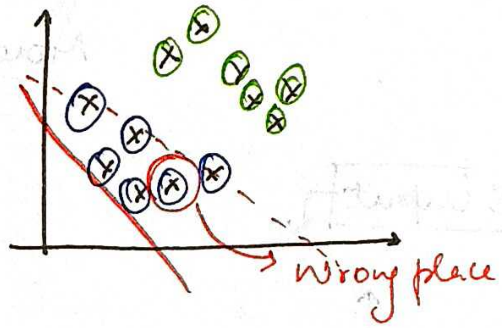
* first step is to take random value of A, B and C.

* Second, select random input and check whether that point is right place or not.

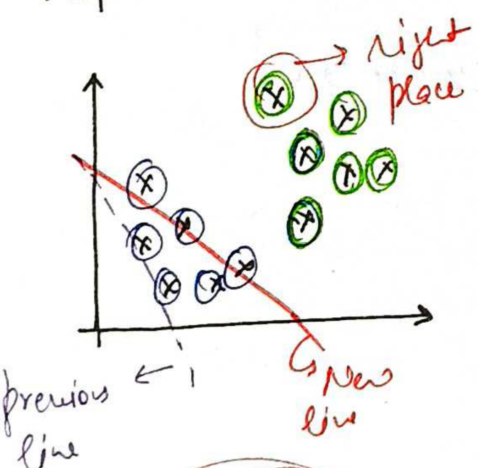
Step 1

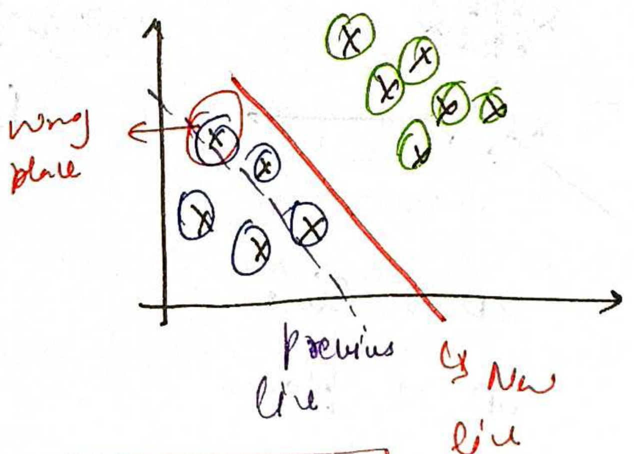


Step 2



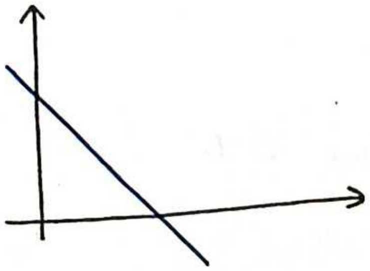
Step 3





Check misclassified ← New line → Shift or change A, B, C → Loop 1000 times

How to label regions?



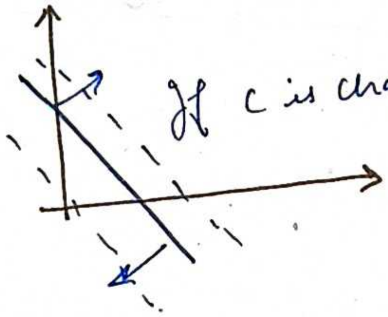
If $Ax_1 + By_1 + C > 0$ then point in 1 region

If $Ax_1 + By_1 + C < 0$ then point in 0 region

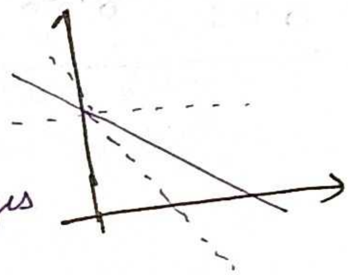
If $Ax_1 + By_1 + C = 0$ then point on the line

Transformation

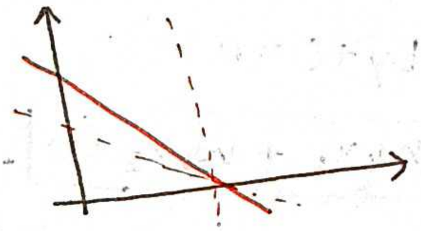
$Ax + By + C = 0$



If C is changing



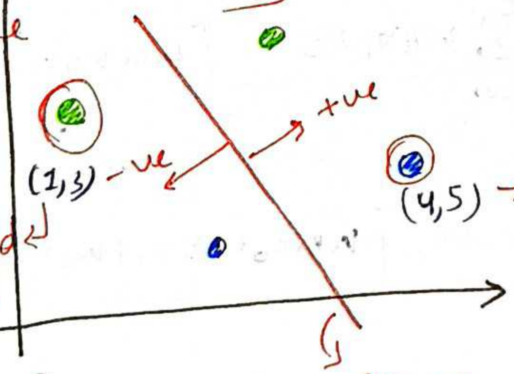
If A changes



* If C↓ then line goes up or If C↓ then line goes down ···›

<--- [Add when line move -ve side]

[Subtract when line move +ve side]



(4,5) → Add 1 (4, 5 1)

1 Add

(1, 3, 1)

$2X + 3y + 5 = 0$

$$\begin{array}{r} 2 \quad 3 \quad 5 \\ +\ 1 \quad 3 \quad 1 \\ \hline 3x + 6y + 6 = 0 \end{array}$$ → New line

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

New line

We cannot directly subtract or Add co-ordin[ates]
bcz line highly change direction. So we
use learning rate.

↳ 0.01 or sometime 0.1

for eg: previous example

co-ordinate → 2 3 5 } multiply with 0.01
1 3 1

	0.02	0.03	0.05
+	0.01	0.03	0.01
	0.03	0.06	0.07

$$\boxed{coef = coef - (\eta \times \text{co-ordinate})}$$

Algorithm

X_1 cgpa	X_2 iq	Y Placed
7.5	61	1
8.9	109	1
7.0	81	0

$Ax + By + C = 0$

$$\boxed{W_0 + W_1 X_1 + W_2 X_2 = 0} \rightarrow \sum_{i=0}^{2} W_i x_i = 0$$

↳ $W_0 = C, \quad W_1 = A, \quad W_2 = B$

$W_0 X_0 + W_1 X_1 + W_2 X_2 = 0$

$W_0 \times 1 + W_1 \times 7.5 + W_2 \times 81 \rightarrow \sum_{i=0}^{2} W_i x_i = 0$

↳ if $> 0 \rightarrow$ 1

↳ if $< 0 \rightarrow$ 0

$$[W_0 \; W_1 \; W_2] \begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix}$$

$$\boxed{W_0 X_0 + W_1 X_1 + W_2 X_2}$$

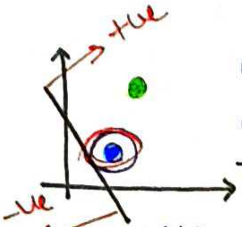
epoch → 1000 , $\eta = 0.01$

for i in range (epochs):

randomly select student

if $x_i \in N$ and $\sum_{i=0}^{2} w_i x_i \geq 0$ → expression

Not placed



Originally data belong to -ve region but we apply expression then answer is +ve region.

$W_{new} = W_{old} - \eta x_i$ → move Negative point

if $x_i \in P$ and $\sum_{i=1}^{2} w_i x_i < 0$

Data is placed originally in positive region but expression's answer is -ve region.

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

Positive when ~~data~~ line move to positive point.

Simplified Algorithm

if $x_i \in N$ and $\sum w_i x_i \geq 0$

$W_n = W_0 - \eta x_i$

if $x_i \in P$ and $\sum w_i x_i < 0$

$W_n = W_0 + \eta x_i$

→

for i in 1000:
random student
$w_n = w_0 + \eta(y_i - \hat{y}_i)x_i$

→ Optimized Code

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

Rows 1 and 2 → $w_0 + \overbrace{\eta(y_i - y_i)}^{0} x_i = w_0$ old (actual = predict)

Row 3 → $w_0 + \eta(1)x_i = w_0 + \eta x_i$

Row 4 → $w_0 + \eta(-1)x_i = w_0 - \eta x_i$

Final Code

for i in range (epochs):
select a random student (i)
$W_n = W_o + \eta (y_i - \hat{y}_i) x_i$

Code → Perceptron-trick.ipynb

code explain

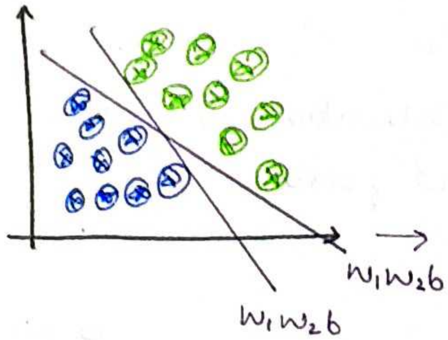
$Ax + By + C = 0 \rightarrow y = mx + c$

where

$m = -\frac{A}{B}$ coef [- -] intercept [ ]

$c = -\frac{C}{B}$

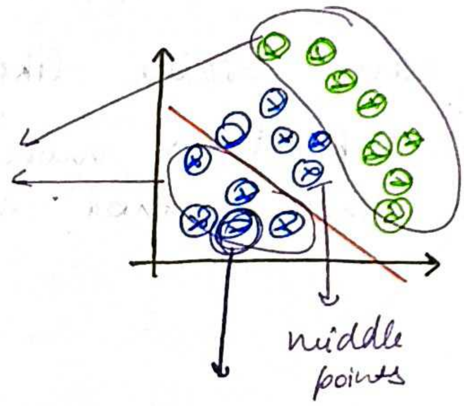
Problem with Perception Trick



→ Problem: Perception not define our classification how much perfect

$\boxed{\text{Not quantify}}$

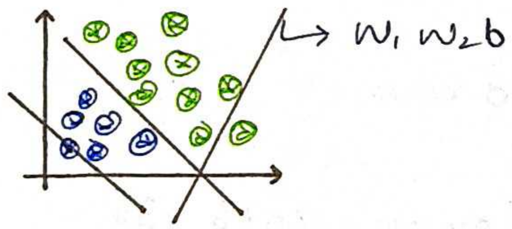
Every time randomly select same group and this same group are correctly classified so middle points are inncorrect.



Sometime randomly select same point every time.

$\boxed{\text{So, we use Loss function to quantify}}$

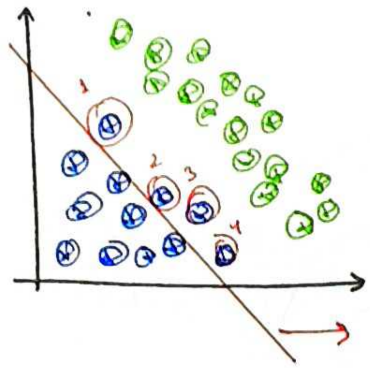
Loss Function



So, loss function gives number to identify which one is better.

Loss function → is a function of $w_1\ w_2\ b \rightarrow f(w_1\ w_2\ b)$ when we chang w_1 or w_2 o b then number will change.

Perceptron Loss Function



$f(w_1, w_2, b) \rightarrow$

number $\rightarrow$ error

misclassified points

$\rightarrow w_1 w_2 b \rightarrow$ point $\rightarrow 4$

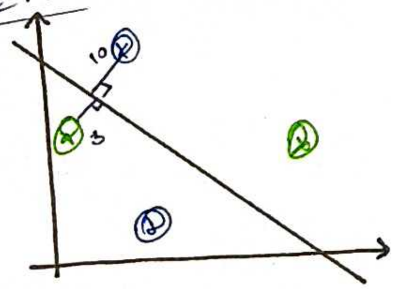
Problem with misclassified points :- Every point give of error same value like $1+1+1+1 = 4$ points

but 3 point $\rightarrow$ more error and 1 point $\rightarrow$ less error

so this method not work.

Distance Error

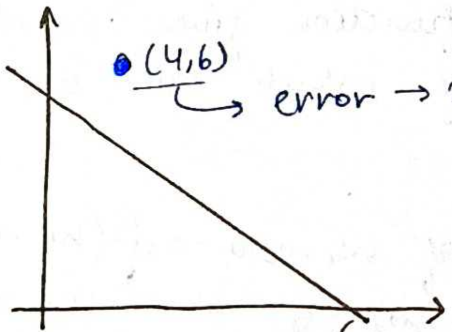
total error $\Rightarrow 10+3 = 13$ $\leftarrow$



But perceptron use diff method for loss function

(4,6)

$\hookrightarrow$ error $\rightarrow 2(4)+3(6)+4 = 30$



Total error $= 30+6 = 36$

$\hookrightarrow 2x+3y+4=0$

(-2,-2)

$\hookrightarrow$ error $2(-2)+3(-2)+4 = |-6| = 6$

This method is more useful cz finding distance is more complex.

Scikit learn (Stochastic Gradient descent) Loss function

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

Loss function

Regularization
this topic discuss later

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

$$f(x_i) = \boxed{w_1 x_1 + w_2 x_2 + b}$$

[when w_1 or w_2 or b change then output will be change]

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

should find loss of every row and then divide by n so find average of loss function

n = num of rows

minimum → gradient Descent

Explanation of Loss function

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

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

X_1	X_2	Y
X_{11}	X_{12}	Y_1
X_{21}	X_{22}	Y_2

row 1, 2, ..., i

X_{ij} → i row, j col

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

let assume this is x.

If $x \geq 0$ → output x

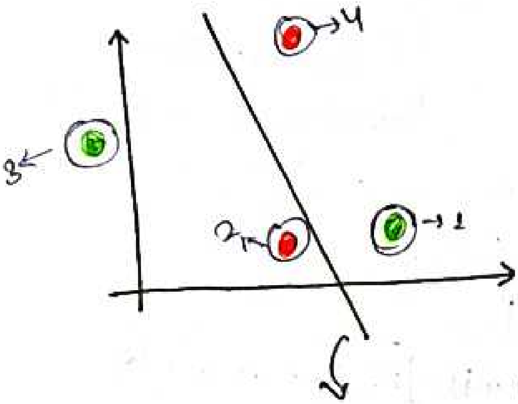
If $x < 0$ → output 0

$$\boxed{-y_i f(x_i)} \geq 0$$

$\ngtr 0$

let assume we have 2 points

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



$w_1, w_2\ b$

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

	y_i	$\hat{y}_i$
1 ←	1	1
2 ←	-1	-1
	1	-1
	-1	1

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

1 point

put 1st point in

$\boxed{w_1 x_1 + w_2 x_2 + b} \rightarrow \max(0 - y_i f(x_i))$

original $f(x_i)$ = +ve
+ve 1 +ve 1

but here is minus $-y_i f(x_i)$

so overall output is -ve.

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

2 point

Same like first

original $f(x_i)$ = +ve
-ve 1 -ve

but overall output is -ve

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

correctly classified
→ max ( ) term would be 0.

If our line classified correctly then that point contribution will be zero bcz already that point are true classified wrong classified point

3rd point

Original $f(x_i)$

+ve -ve = -ve

overall output is +ve → define in 1 point

max (0, +ve) = +ve → Non zero Number

This point will contribute in loss function.

4th point

Original $f(x_i)$

-ve +ve = -ve → define in 1 point

overall output is +ve

This point will contribute in loss function.

Gradient Descent

$$L = \underset{w_1, w_2 b}{\arg\min} \boxed{\frac{1}{n} \sum_{i=1}^{n} \max(0, -y_i f(x_i))}$$

differentiate w.r.t

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

and we get

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

$\boxed{w_1, w_2 b} = 1$

↳ for i in epochs:

$$w_1 = w_1 + \eta \frac{\partial L}{\partial w_1}$$

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

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

Loss Function Differentiation

$$L = \frac{1}{n}\sum_{i=1}^{M} \max(0, -y_i f(x_i)) \quad \text{where } f(x_i) = w_1 x_{i1} + w_2 x_{i2} + b$$

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

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

$$\frac{\partial f(x_i)}{\partial w_1} = x_{i1}$$

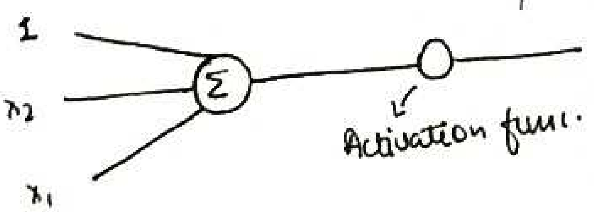
$$\boxed{\frac{\partial L}{\partial w_1} = \begin{cases} 0 & \text{if } y_i f(x_i) \geq 0 \\ -y_i x_{i1} & \text{if } y_i f(x_i) < 0 \end{cases}}$$

$$\boxed{\frac{dL}{dw_2} = \begin{cases} 0 & \text{if } y_i f(x_i) \geq 0 \\ -y_i x_{i2} & \text{if } y_i f(x_i) < 0 \end{cases}}$$

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

More Loss function

We used step func (Activation func.) but we can also use sigmoid func. and also use binary cross entropy (loss func.). That's why perceptron is a flexible. (This is for binary classification.)



For multiclassification → Activation func. → soft max

$$f = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$$

↳ Categorical cross entropy

$$L = \sum_{j=1}^{M} y_i \log(\hat{y}_j)$$

[more flexible]

Perceptron also use as linear Regression

Activation → linear but in classification we use

$z \geq 0$ 1

$z < 0$ 0

$z \to z$

↳ which mean we are not using any activation func.

Loss function → Linear reg

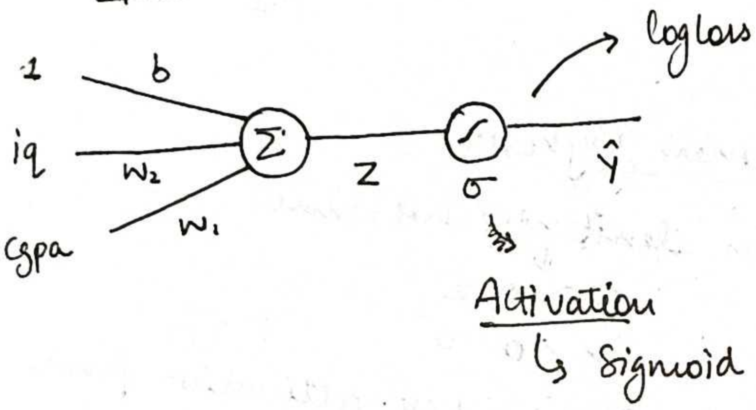
↳ mse → $(y_i - \hat{y}_i)^2$

Perceptron is a mathematical model it can be use any ways.

Loss function	Activation	Output
Hinge loss	Step func.	perceptron → binary classifi (-1, 1)
log-loss (binary cross entropy)	Sigmoid	Logistic regression (0-1) → binary classif (Probability) ↳ output
Categorical cross entropy	Softmax	Softmax → multiclass regression classifi (Probability)
mse	linear	linear regression ↳ Number ↳ output

Multiple perceptron

Perceptron with sigmoid



Sigmoid → probability
0 → 1

Eg:- $w_1 = 5$, $w_2 = 10$, $b = 3$

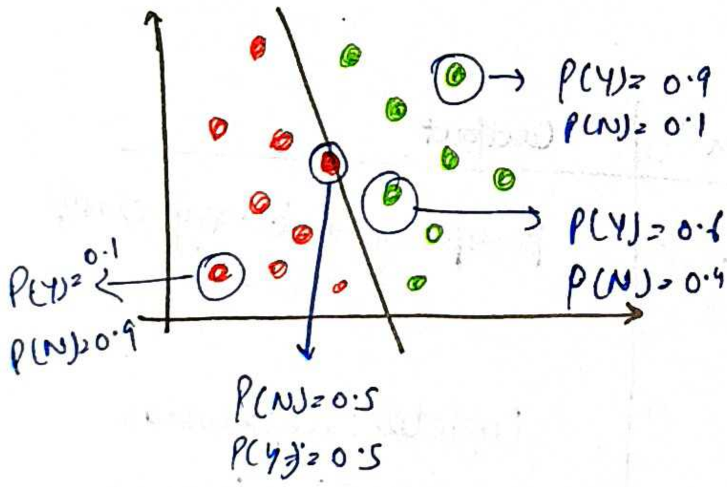
line → $5x + 10y + 3 = 0$

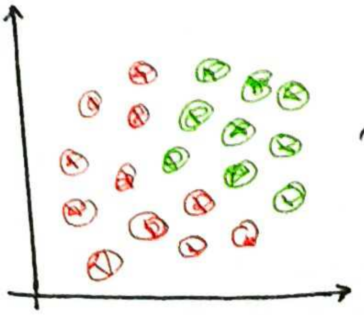
cgpa = 8.7, iq = 8.7 → $5 \times 8.7 + 10 \times 8.7 + 3 = 0$

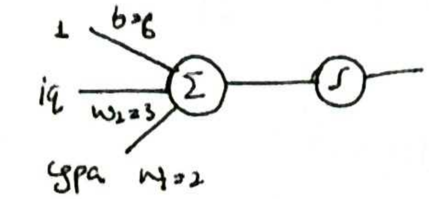
Calculate z ⇒ $\frac{1}{1+e^{-z}} =$ 0 → 1

P(Y) = 0.8
↳ Probability of yes

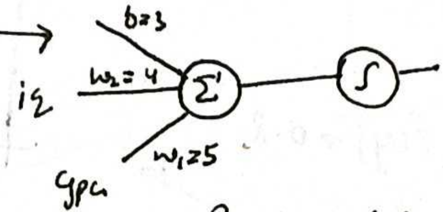
P(N) = 1 − 0.8 = 0.2
↳ prob of no.



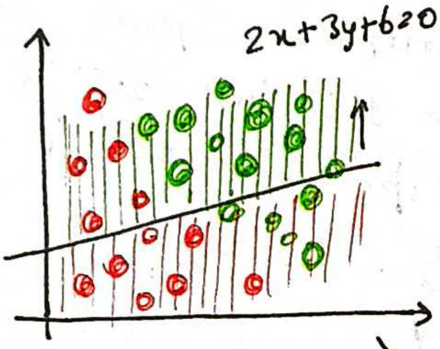




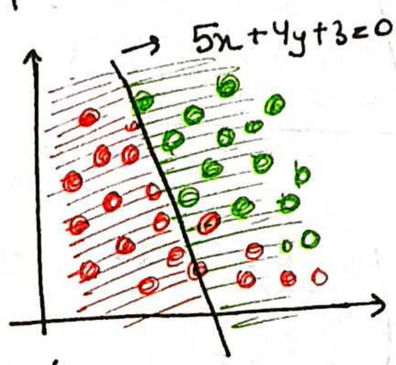
Perceptron(1)



Perceptron(2)

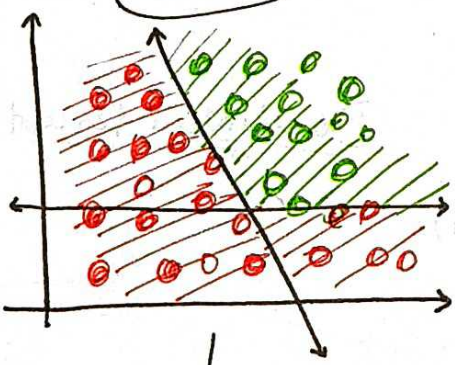


Perceptron(1)

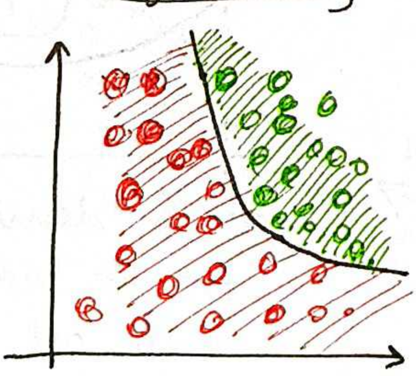


Perceptron(2)

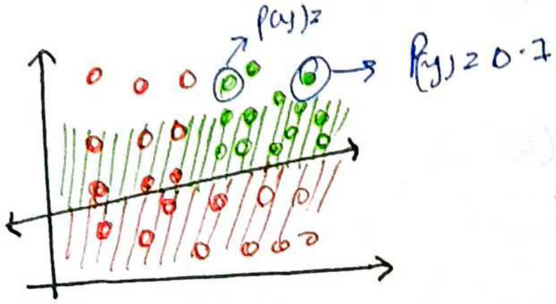
Combine



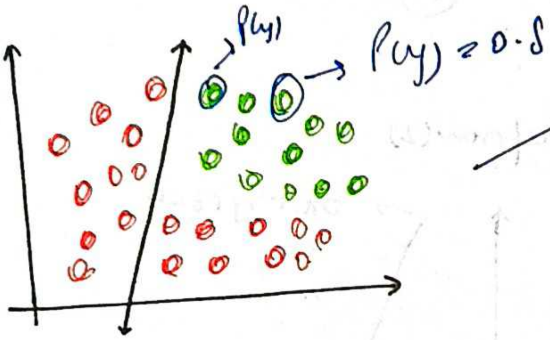
After smoothing

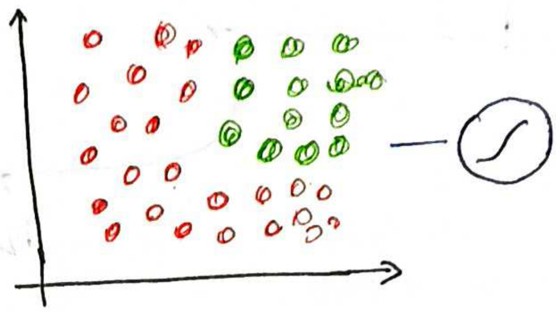


Explain with Mathematically

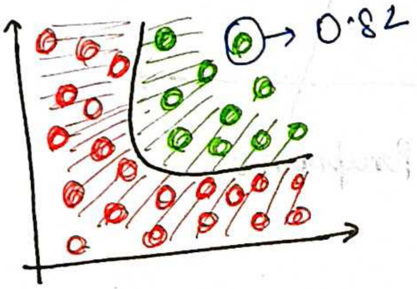


+

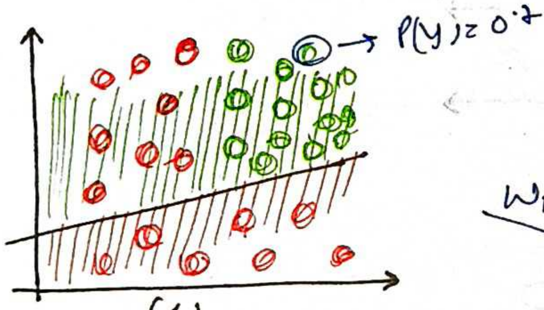




when we Add first and second perceptron prob is $0.7 + 0.8 = 1.5$ but 1.5 is not possible cz prob ($0 \to 1$). So, we use (Z) $\frac{1}{1+e^{-1.5}} = 0.82$

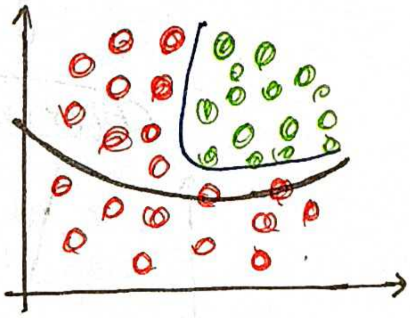


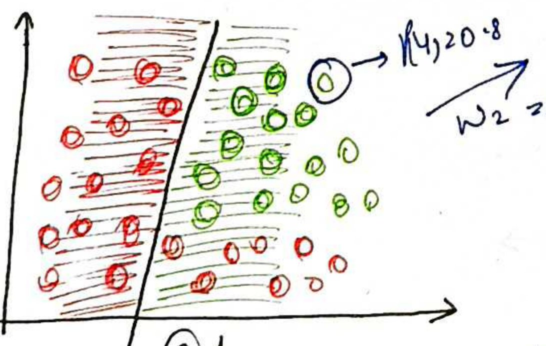
Domination Perceptron



(1)

$w_1 = 10$



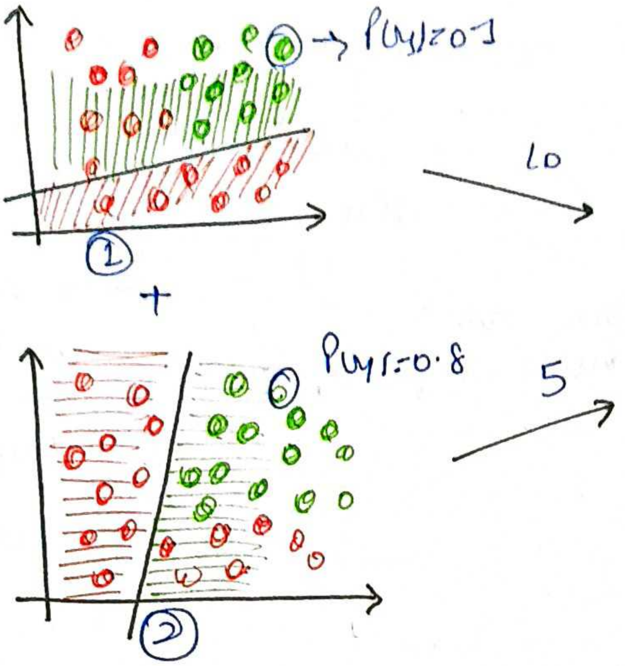


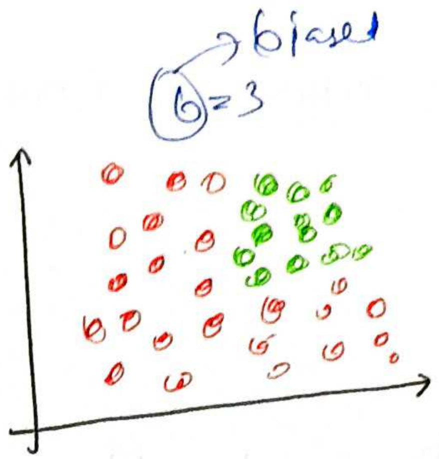
(2)

$w_2 = 5$

* In domination, we add Prob of both perceptron and also add weight of each perceptron. for domination a perceptron weight of (1) perceptron increase.

Add Biased

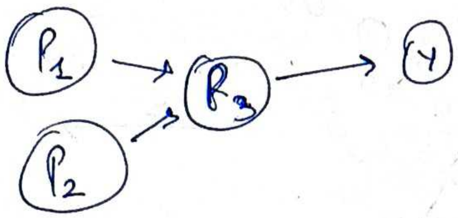


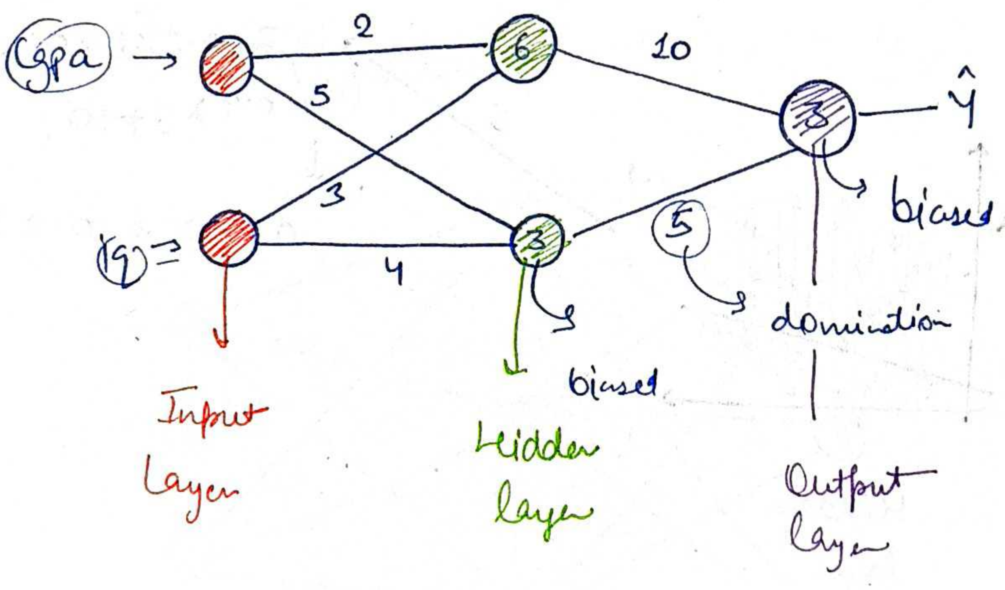


$\boxed{Z} = 0.7 \times 10 + 0.8 \times 5 + 3$

$\hookrightarrow \sigma(z)$ 3 → biased

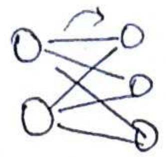
The output of ① and ② perceptron is the input of the third perceptron is know as Multi layer perceptron.





Adding nodes in hidden layer

Architecture → how nodes connecting with each other

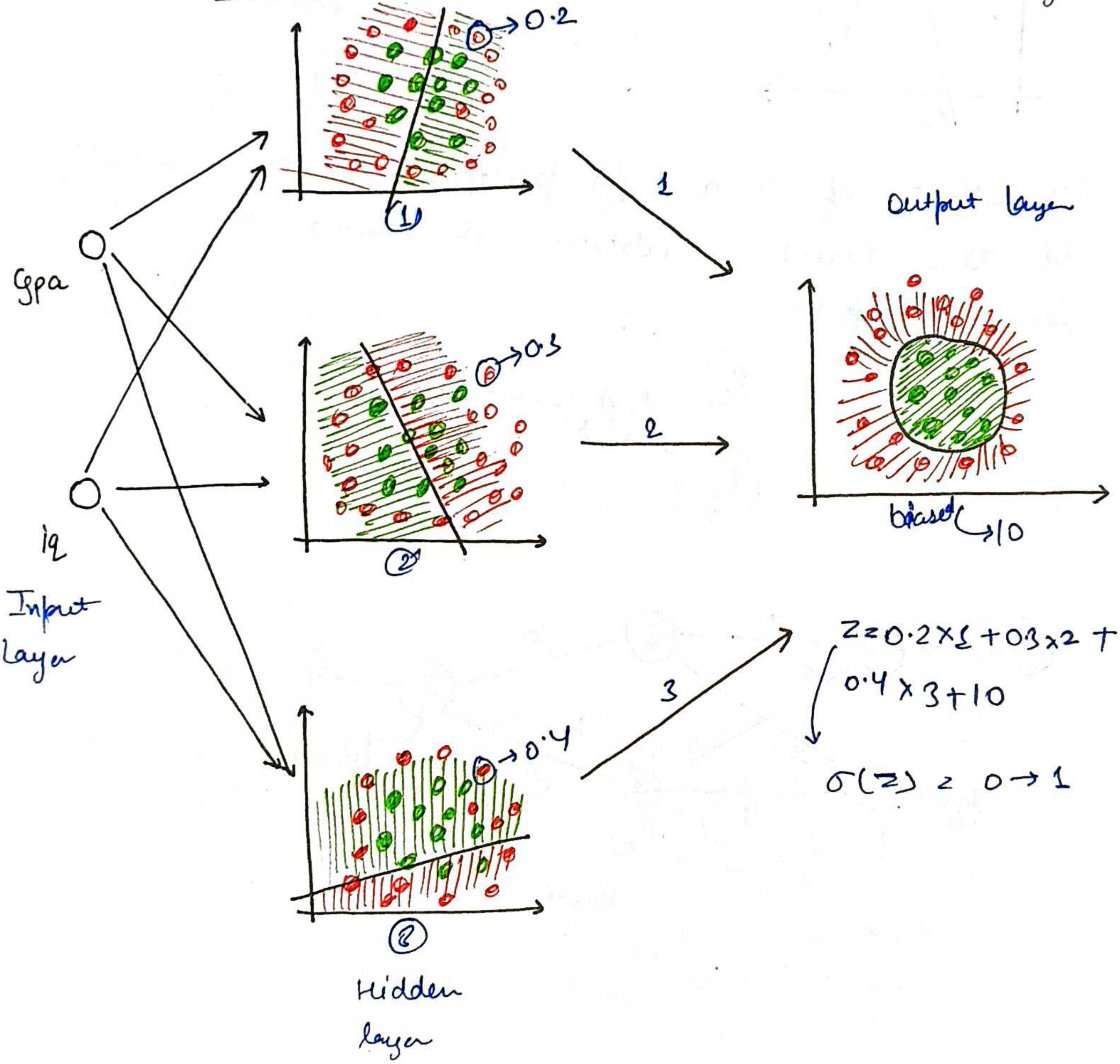


* we can change the architecture

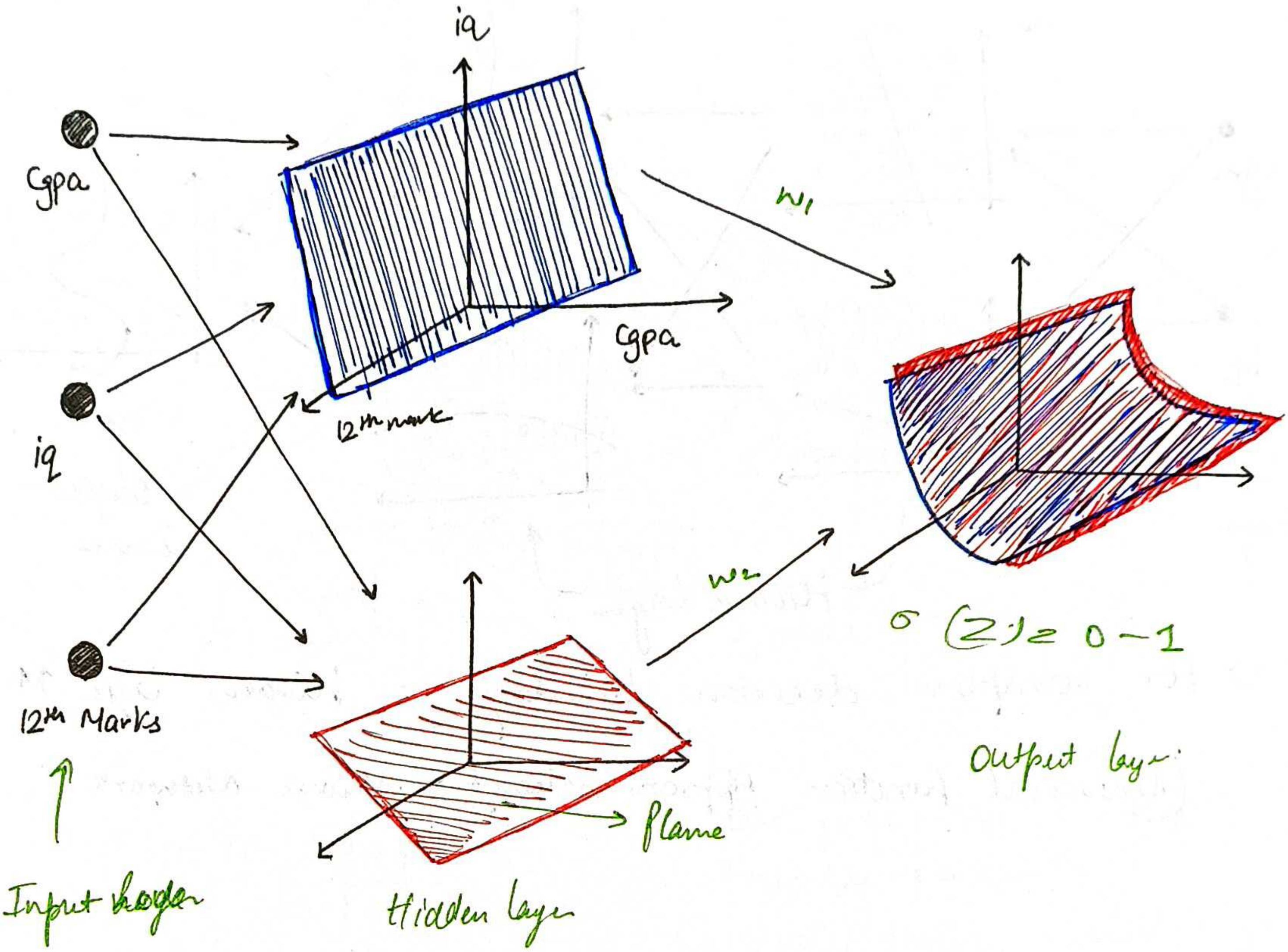
① we can add node or perceptron

→ Add Node in input layer

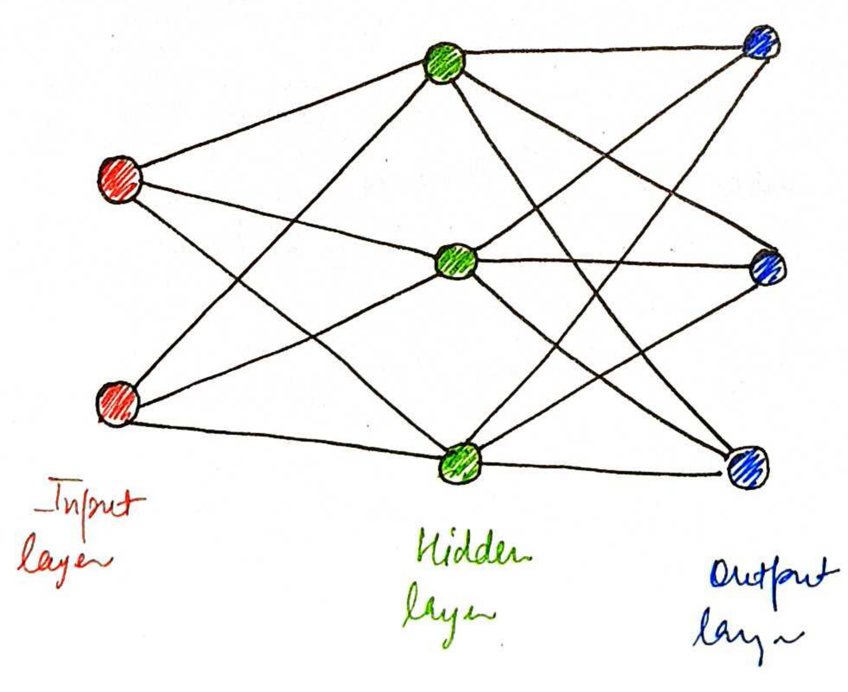
① Add perceptron



② Adding nodes in input

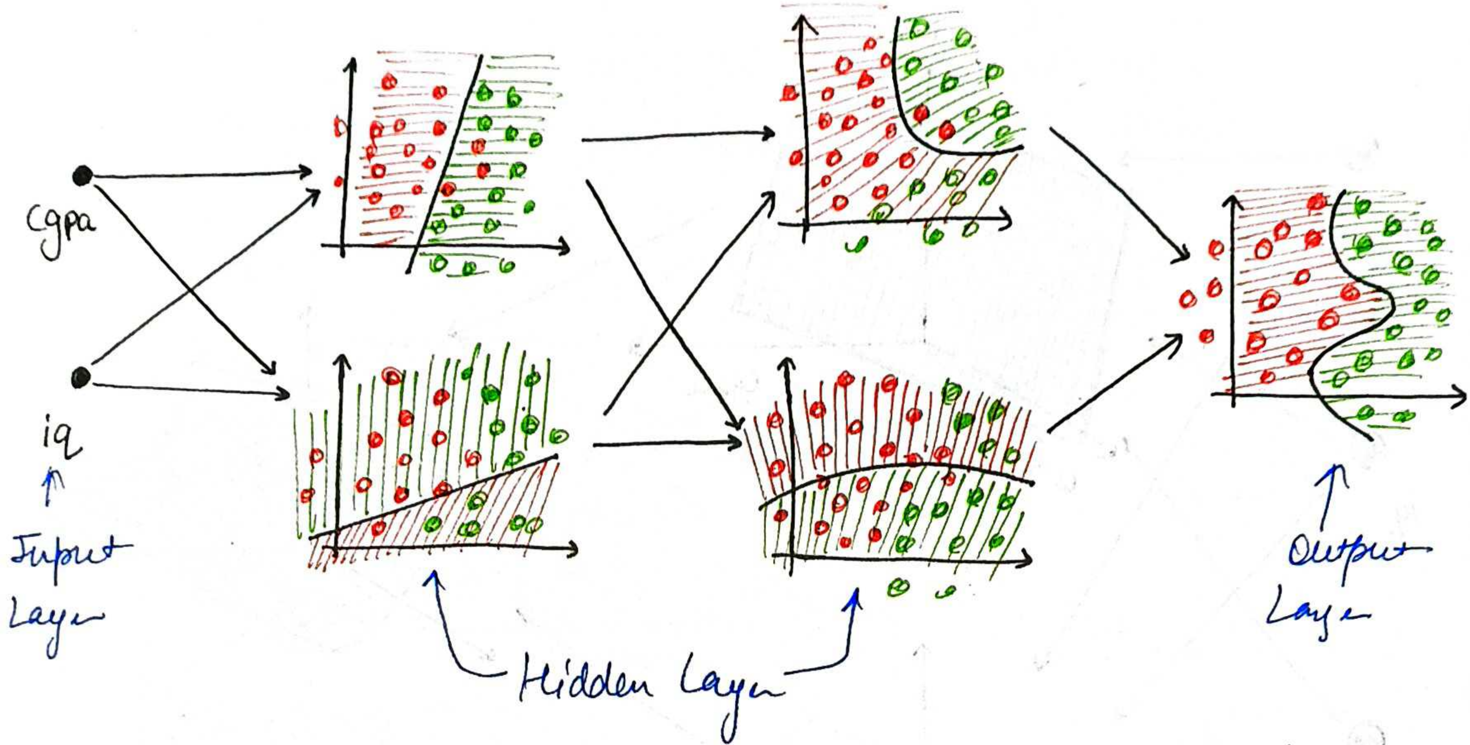


③ Adding Nodes in output node



* this is used for multi-Classification like Identify Cat, Dog and Human.

④ Deep Neural Network → Add hidden layer



→ For complex decision boundary → Hidden layer ↑↑

[Universal function Approximation] → Neural Network

Multi-Layer Perceptron

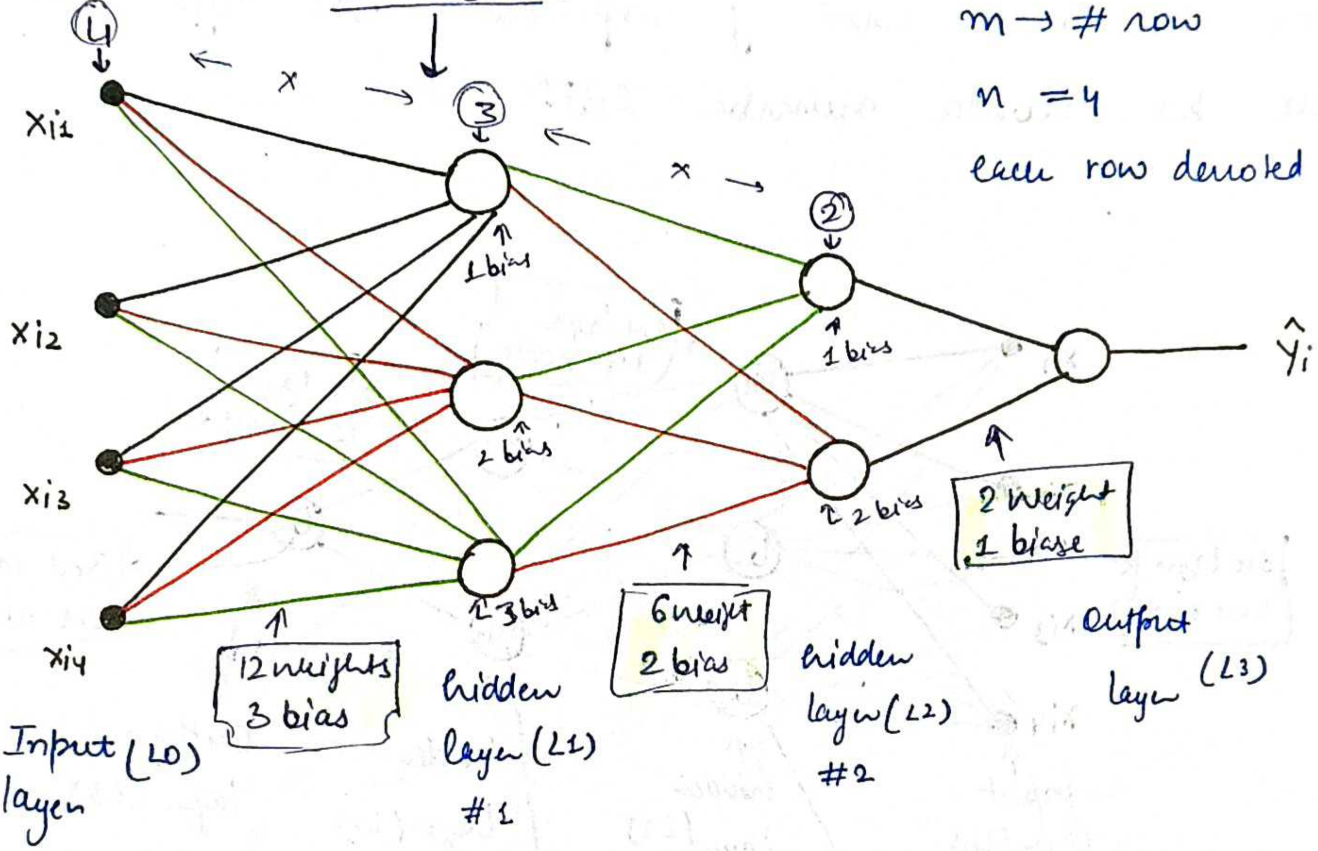
Data → {m × n}

m → # row

n = 4

each row denoted by x_i

Architecture



1	2	3	4	$\hat{y}_i$
-	-	-	-	-

i

* Our first step is to find trainable parameter

↳ during training, backpropagation algo finds weights and biases. how many

→ Architecture

$4 \times 3 = 12 + 3 = 15$

4 = Input Node, 3 = hidden Node (L1), 12 = weight, 3 = biases

$3 \times 2 = 6 + 2 = 8$

3 = hidden Layer (L1), 2 = hidden Layer (L2), 6 = weight, 2 = biases

$2 \times 1 = 2 + 1 = 3$

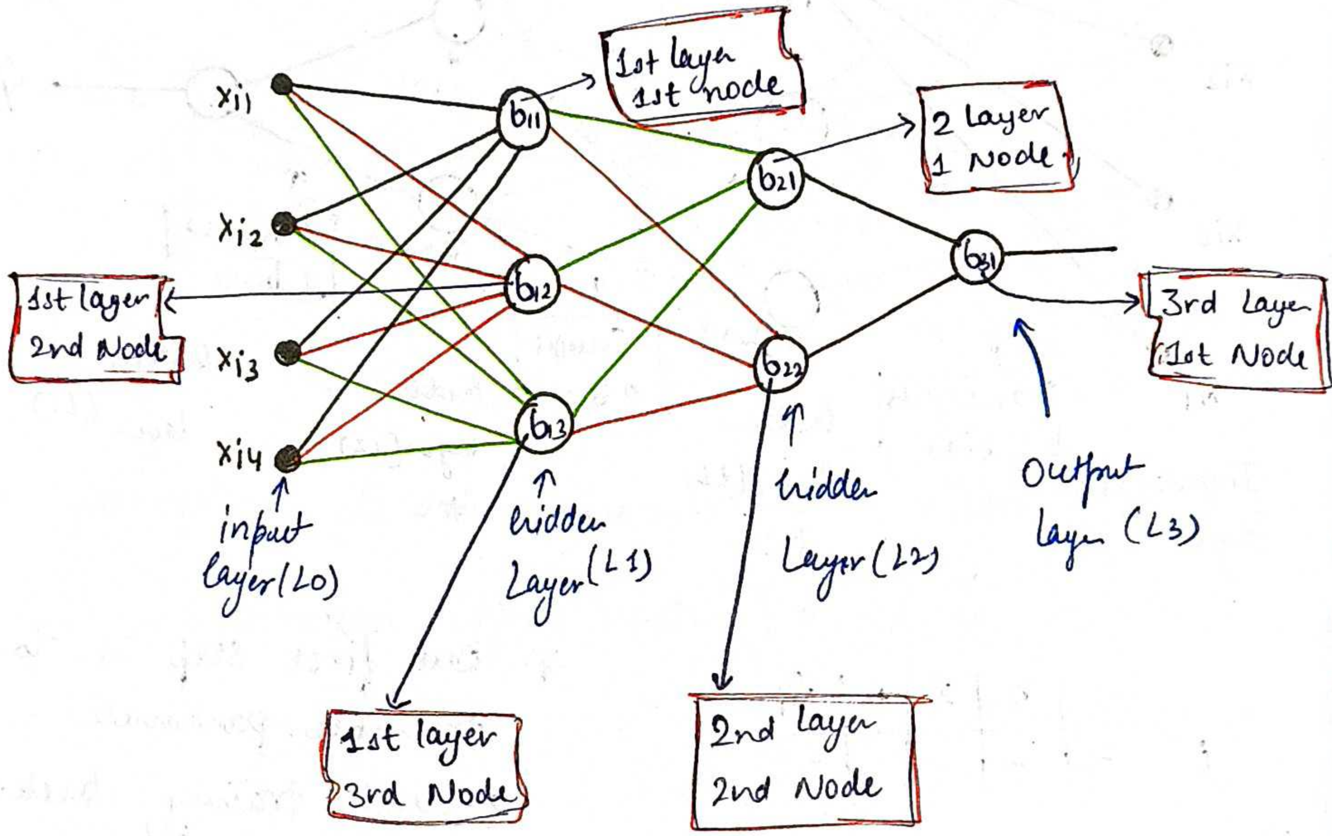
2 = hidden Layer (L2), 1 = output layer, 2 = weight, 1 = biases

$15 + 8 + 3 = 26$

So, Backpropagation trying to find ~~the~~ value of the 26 weights and biases.

Notation of bias

Notation of bias is bij where i is represent layer number and j represent "iss layer mai node ka konsa number hai."

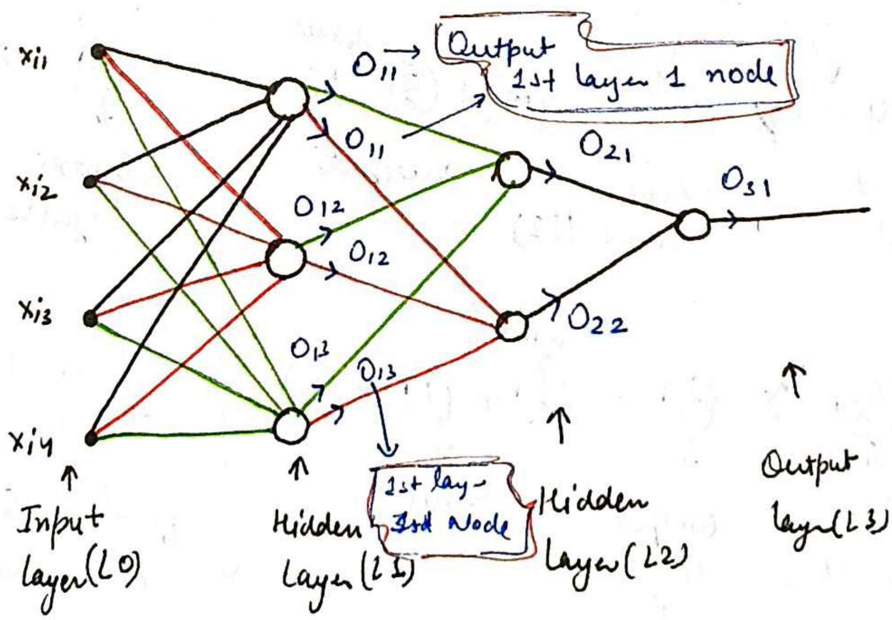


Notation of output

↳ O_{ij}

Layer No.

No. of node of the Same layer No.



Notation of Weight

↳ W_{ij}^{k}

i → perious Node Number

j → entering Node Number

k → Entering layer Number