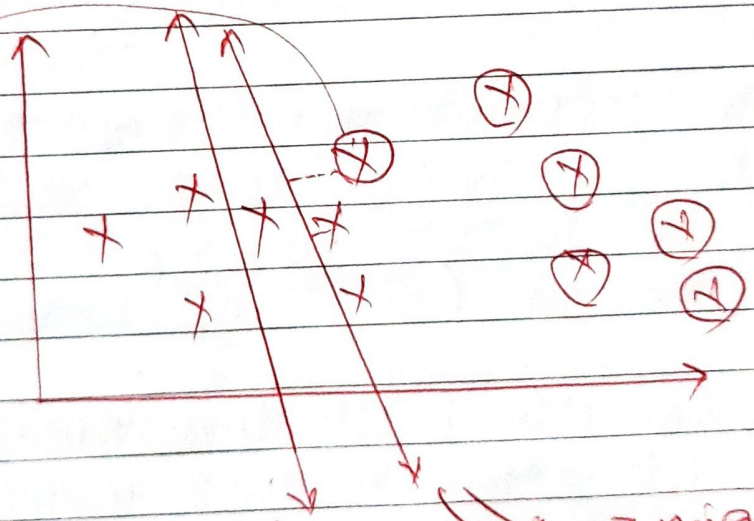
LOSS FUNCTION FOR PERCEPTIONS

- Loss function is function of w_1, w_2, w_{bord} b.
- We need such values of weights and bias that our error / loss become minimum.
- Let's find error function for perceptron,

i) No. of misclassified points (could be this is error function).

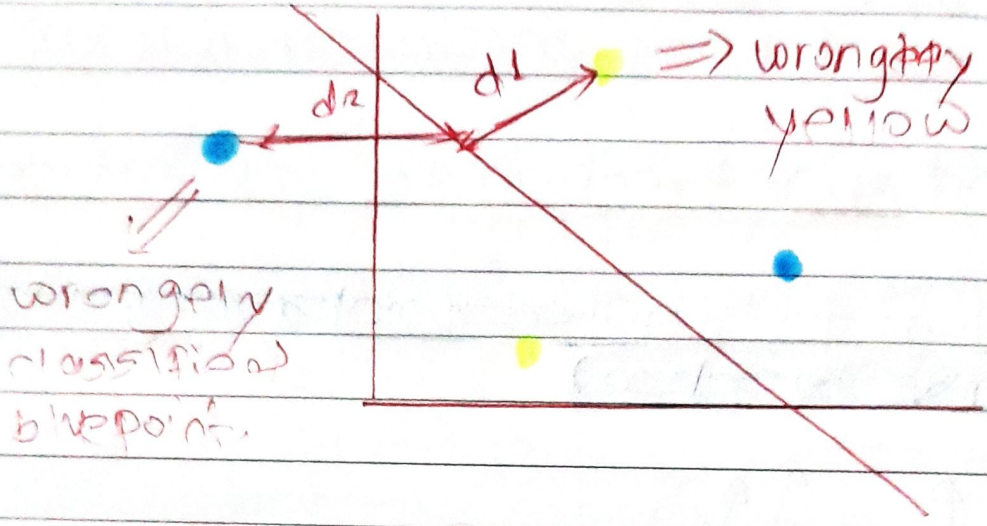


4 misclassified point

3 misclassified point after transformation

- Above approach treat every point as same but it cannot tell which point make how many magnitude of error.
  → This point make more error that other points (Basically this point misclassified badly)

1) Second approach could be calculate distance of points from our model line separation.

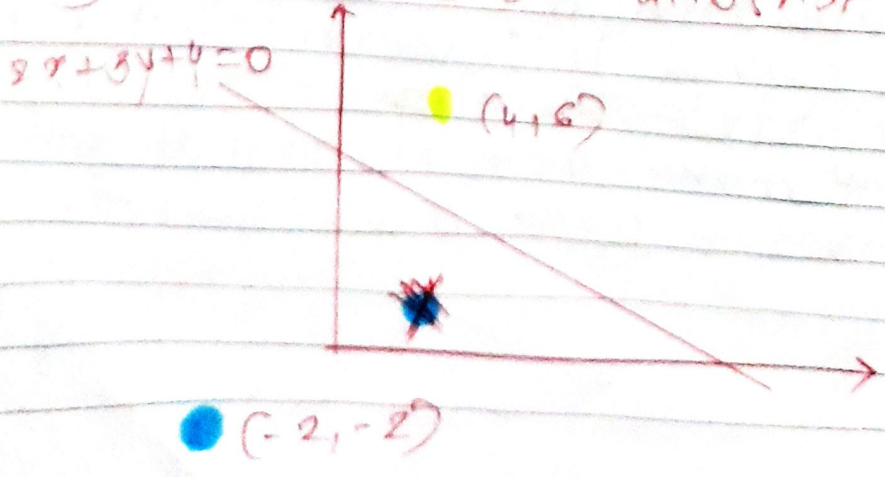


- To calculate perpendicular distance of points from this line, for misclassified points
  - (d1) ⇒ 10
  - (d2) ⇒ 13

  10 + 13 = 23 (total error)

- This loss function better than previous loss function, Because we bring magnitude of error for each miss classified point.

Also, separation uses different loss function than distance as misclassification magnitude. Let's see another approach.



- Both of this 'blue' and 'yellow' point are misclassified point.

- Rather finding distance, we put point's coordinate's in model line equation (i.e. perceptron equation)

  - (4, 6) $\Rightarrow$ 30 | (2×4 + 3×6 + 4 = 30)|
  - (-2, -2) $\Rightarrow$ +6 | (2×-2 + 3×-2 + 4 = -6)|

- Rather than distance we use this, Because calculating distance computation is exp. naive.

- This is just dot product.

- In sklearn what is actual loss function for perceptron ?

- Go to SGD page in documentation,

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

- This is general equation of SGD (error)

- $\alpha R(w)$ $\Rightarrow$ Regularization parameter

This is very close to hinge loss function,

$$L(y_i, f(x_i)) = max(0, 1 - y_i f(x_i))$$

- SGD perception से काम करना है उसका function ही हो जाता है.

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

- So, according SGD general equation our loss function for perceptron becomes,

$$L(w_1, w_2, b) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) + 0$$

(No regularization for now)

$$\alpha R(w_1, w_2) = 0$$

- **What is $f(x_i)$?**

$$f(x_i) = w_1 x_1 + w_2 x_2 + b = z$$

- So final loss function is,

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

⇒ Average error of all data points/rows

n ⇒ no. of rows in data point.

What is y_i?

- Our output in data. (Basically 1 or 0)

- This loss function depends on three things (w_1, w_2, b) basically loss function is function of this (w_1, w_2, b)

- We have to make a such a changes in w_1, w_2 and b such that $L(w_1, w_2, b)$ value becomes minimum.

- Mathematically, (We have to use gradient descent).

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

Before that what is geometric intention of this loss function?

- This is our loss function,

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

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

- We have this dataset,

	x_1	x_2	y
1	x_{11}	x_{12}	y_1
2	x_{21}	x_{22}	y_2
⋮			
ith row			
⋮			
n			

- **What is max(0, -yf(xi))?**

Let's assume, $-yf(x_i) = x$,

$\max(0, -yf(x_i))$ becomes $\max(0, x)$

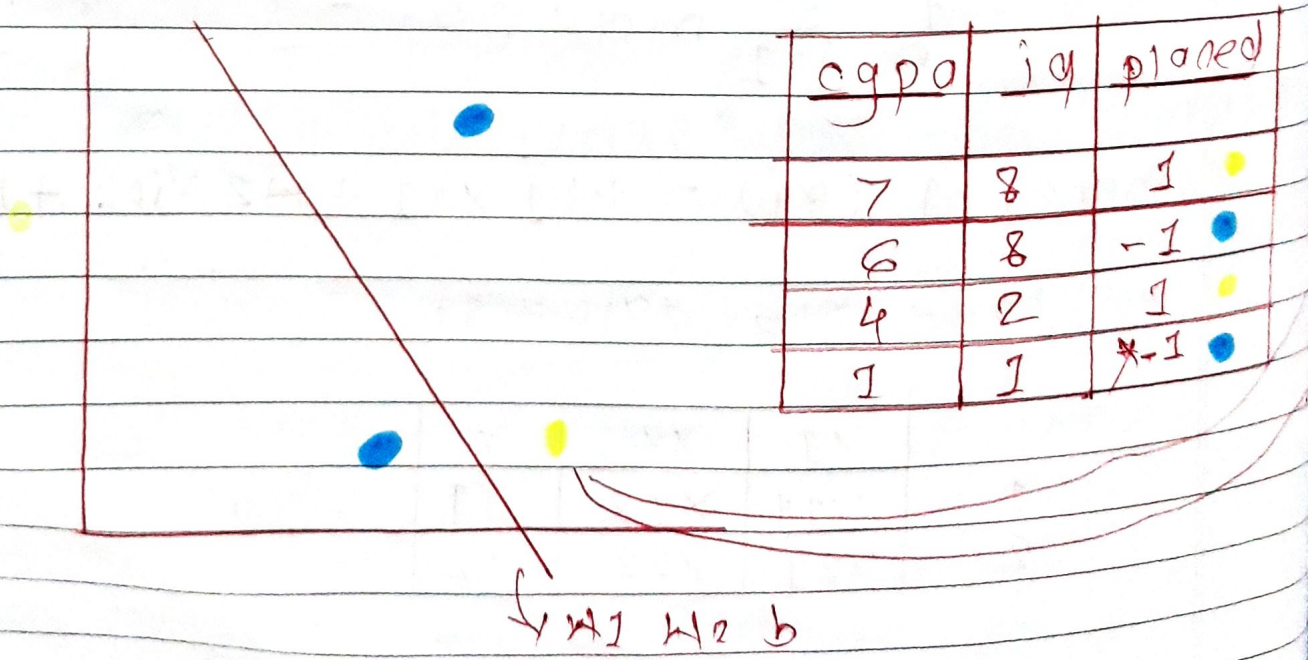
- Consider we have only two points, so break down this loss function.

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

$f(x_1) = w_1 x_{11} + w_2 x_{12} + b$ (first row)

$f(x_2) = w_1 x_{21} + w_2 x_{22} + b$ (second row)

- **Geometric intuition**



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

- Line context में four points हो सकते है,

y_i	x_i
1	1
1	-1
-1	1
-1	-1

- We are going to find meaning of this point, for this, $\max(0, -y_i f(x_i))$
- i) condition one, $y_i = 1$ and $\hat{y}_i = 1$
  → for this point, this point is on possitive side of line and not classified after putting this point co-ordinate in line equation

  $w_1 x_1 + w_2 x_2 + b$ it will become possitive.

$w_1 x_1 + w_2 x_2 + b \geq 0 = f(x_i)$

$\max(0, -1 \times (+ve))$ $\quad y_i = 1$

$\max(0, -\text{negative value})$

max becomes 0.

- ii) condition 2 , $y_i = -1$ and $\hat{y}_i = -1$

$+\hat{y}_i = -1$

$w_1 x_1 + w_2 x_2 + b < 0$

-ve x -ve = +ve , so,

max (0, $-y_i$ (-ve value)

max (0, - (-1 (-ve value))

max (0, - (+ve value)

~~max (0, 0~~

so, max becomes zero , i.e. no error.

- iii) condition 3 , $y_i = 1$ and $\hat{y}_i = -1$.
(misclassified point)

$+\hat{y}_i = +1$

$f(x_i)$ becomes negative.

- (+1 x - (negative $f(x_i)$))

max (0, +ve)

+ve value

If point is misclassified then it has some contribution in loss function.
If point is correctly classified it will has zero contribution.

iv) condition $y_i = -1$ $\hat{y} = 1$

$-1(-1 \times f(x) + ve)$ (misclassified point)

$\max(0, +ve)$

+ve value.

- $\hat{y}_i$ predicted from 'z' calculation and actual value put in loss equation y_i

$\max(0, y_i(w_1 x_1 + w_2 x_2 + b))$

Gradient Descent

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

where $f(x_i) = w_1 x_{i1} + w_2 i_2 + b$

for i in epochs:

$$w_1 = w_1 + \eta \frac{dL}{dw_1}$$

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

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

- For w_1,

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

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

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

combining

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

- Going by same logic, for w_2

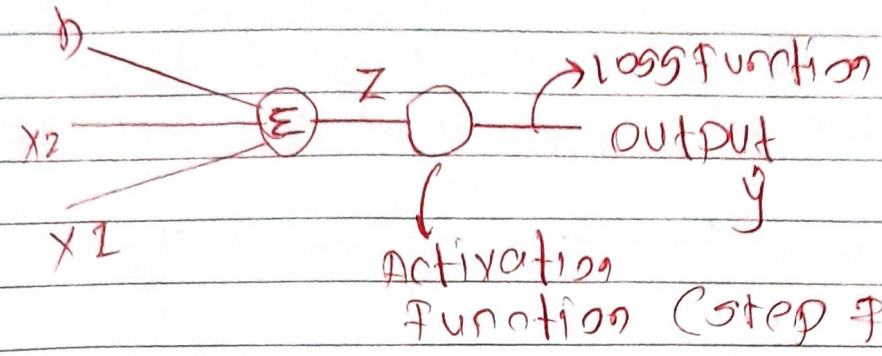
$$\frac{\partial L}{\partial w_2} = \begin{cases} 0 & \text{if } y_i f(x_i) \geq 0 \\ -y_i x_{i2} & \text{if } y_i f(x_i) < 0 \end{cases}$$

- Also for, b

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

- Refer notebook from yjdep.

More Loss function



- To get output in probabilities we can replace with sigmoide function as a activation function.

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

- Binary cross entropy as loss function and sigmoide as activiation function then our perceptron equalls to logistic regression.

- For multiclass classification,

Activation function ⇒ softmax ⇒ categorical cross entropy (loss function) ⇒ this is softmax regression

- For regression

Activation function ⇒ Linear → No activation function, z as z (output)
Loss function ⇒ mse

- change activation function and loss function we get parameter every time by using SGD.

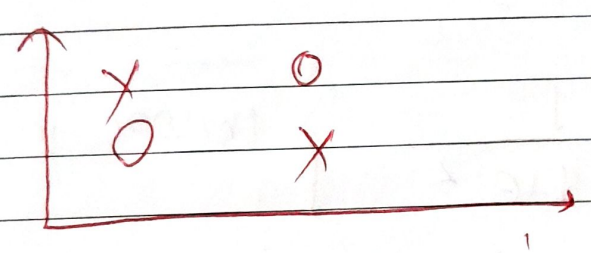
- In summary,

This is because design of perceptron is flexible.

Loss function	Activation	Output
Hinge-loss	step	perceptron - binary classifier.
Log-loss (binary cross entropy)	sigmoide	Logistic regression 0 - 1 binary classifier
categorical cross entropy	softmax	softmax regression (Multiclass classifier) output is probability
MSE	linear	Linear regression

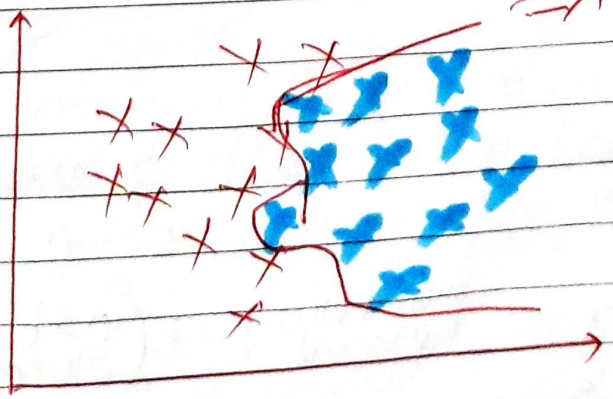
Problem with perceptron

- It not work with non-linear data.
- perceptron work with OR dataset, AND dataset but not work XOR dataset (it cannot even able create boundary / decision tree between point).
- you can check this with tensorflow playground / neural net tool.



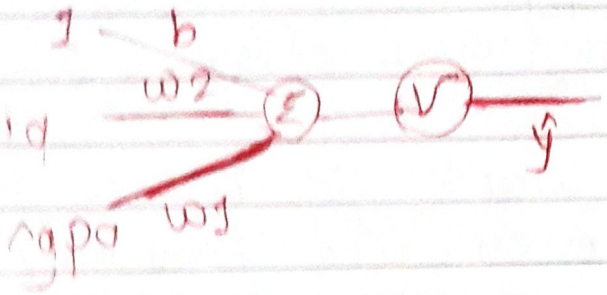
The problem

- We want to make algorithm that going to ~~make~~ capture non-linearity in data by using multiple perceptron.



- Before we have to know perceptron with sigmoid.

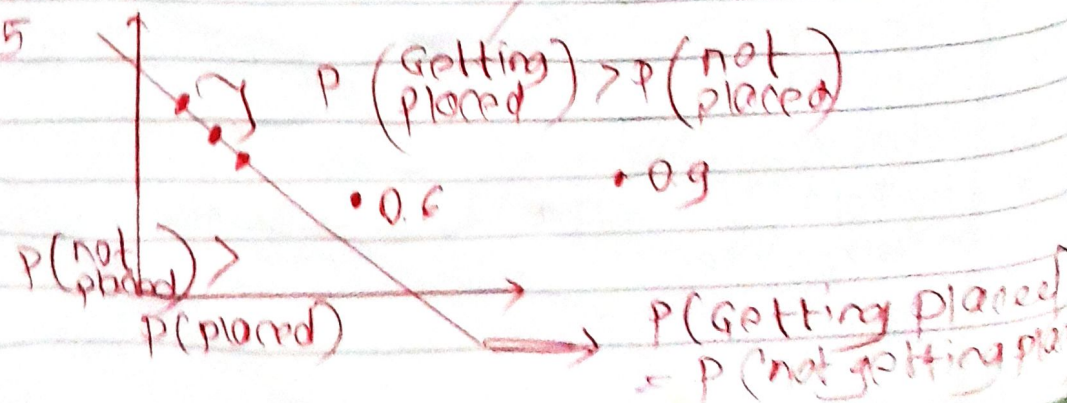
Perceptron with sigmoide.



- And loss function is log-loss (logistic regression)
- Sigmoide activation function gives probability between 0 and 1

$$\text{sigmoide}(z) = \frac{1}{1+e^{-z}}$$ } Answer in 0-1.

- Line equation is after calculating weights and bias
  $w_1 \text{cgpa} + w_2 \text{Iq} + b = z$.
- And we put this 'z' value in sigmoide activation function.
- If point lies exactly on this line, then probabilities of that point getting placed and not placed is exactly similar.



- जैसे जैसे हम line से दूर जायेंगे वैसे वैसे negative probabilities बढ़ेंगी, in P(getting placed) > P(not getting placed) region and opposite side Probability of getting placed reduce होती है