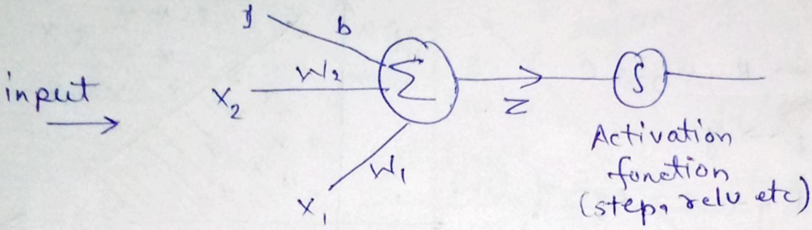
14-Apr-2024

PERCEPTRON

what is perceptron?
it is an algorithm and it is used in supervised ml algo. due to design of this it is basic building block of DL

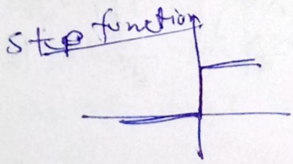


$W_1, W_2 \rightarrow$ weights
$b \rightarrow$ bias
$x_1, x_2 \rightarrow$ input
S → Range में लाना

it is also a mathematical model/function.

$$Z = W_1 X_1 + W_2 X_2 + b$$

Range में लाने के लिए हम लोग एक function का use करते है for example step function, relu etc.



$$\begin{rcases} Z \geq 0 \\ Z < 0 \end{rcases} \begin{matrix} \rightarrow 1 \\ \rightarrow 1 \end{matrix}$$

x_1 iq	x_2 cgpa	placed
78	7.8	1
69	5.1	0

training data

use training data and find the value of W_1 and W_2 then now we can predict the output using perceptron.

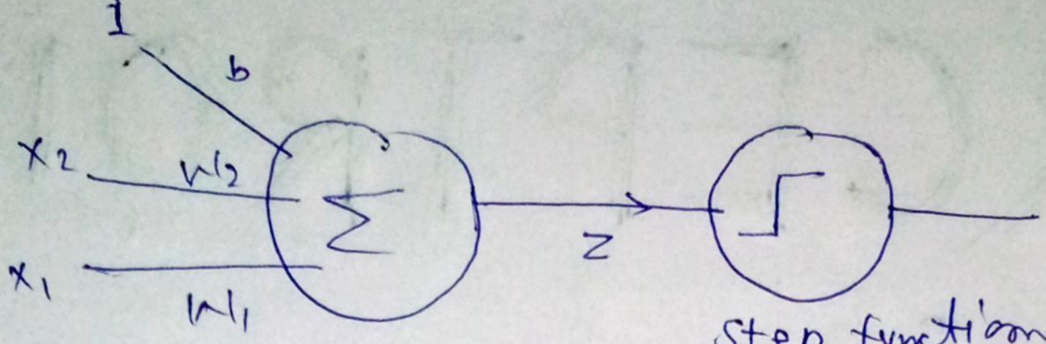
for example. $W_1 = 1$ $W_2 = 2$
$b = 3$ then
cgpa = 5.1, iq = 100 make prediction.

$100 \times 1 + 5.1 \times 2 + 3 \times 1 = 113.2 \geq 0$

* hence placed

* weight (w_1, w_2) are tell us about connection strengths (feature importance बताता है)

Geometric Intution of perceptron:



step function (activation function)

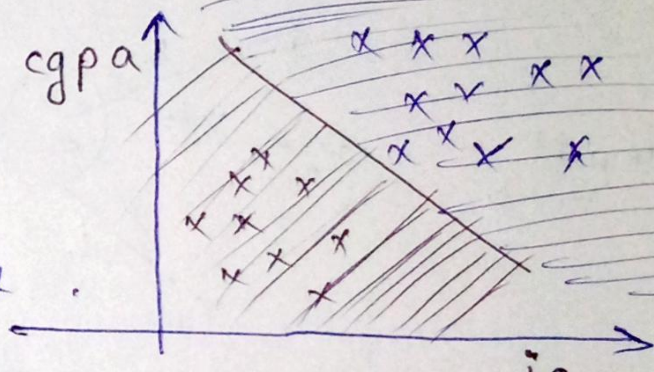
$y = 1$ ~~and~~ or 0 $\qquad z = w_1 x_1 + w_2 x_2 + b$

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

if $w_1 \to A \quad w_2 \to B \quad b \to C$

$x_1 \to x \quad x_2 \to y$

then $Ax + By + C =$ eqn of line.

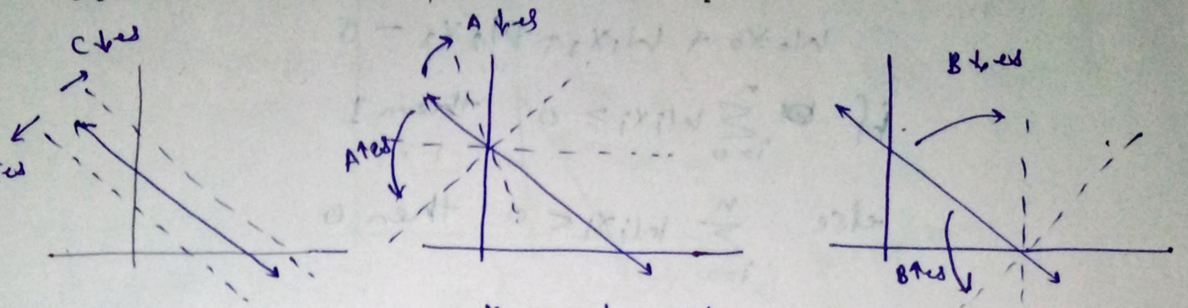


$Ax + By + C \geq 0 \longrightarrow$ positive region

$Ax + By + C < 0 \longrightarrow$ negative region.

* hence perceptron is binary classifier.
* the limitation of perceptron is only apply on linear or sort of linear data को ही classify कर सकता है।

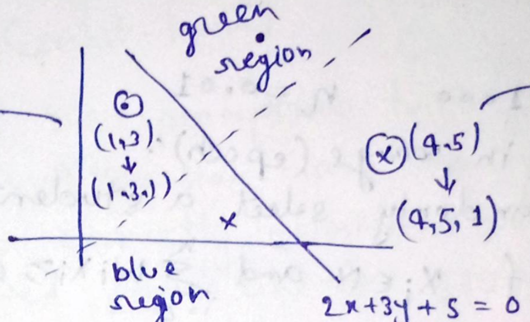
How to transform line?



if c changes   if A changes   if B changes.

2	3	5
1	3	1

$3x + 6y + 6 = 0$



2	3	5
4	5	1

$-2x - 2y + 4 = 0$

अगर -ve point +ve region मे है तो हम उसके coordinates मे last में 1 add कर देंगे और फिर line के coefficients से subtract कर देंगे।
और अगर +ve point -ve region मे है तो same process as above but at last add with line's coefficient.

$$\text{new coefficient} = \text{old coefficient} - \eta \times \text{coordinates}$$

Here, learning rate $(\eta) = 0.01$

in ML The long transformation example in above. we can't do that, we do this is step wise

Algorithm:

X_0	X_1	X_2	Y (prediction)
1	cgpa	iq	placed or not
1	7.5	81	1
1	8.9	109	1

$Ax + By + c = 0$

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

if $\sum_{i=0}^{n} W_i X_i \geq 0$ then 1

else $\sum_{i=0}^{n} W_i X_i < 0$ then 0

Here $W_0 = C$, $W_1 = A$, $W_2 = B$

Now,

epoch = 1000 $\quad \eta = 0.01$

for i in range (epoch):

randomly select a student

if $X_i \in N$ and $\sum_{i=0}^{n} W_i X_i \geq 0$] -ve point +ve region में है

$W_{new} = W_{old} - \eta X_i$ — (i)

else $X_i \in P$ and $\sum_{i=1}^{n} W_i X_i < 0$] +ve point -ve region में है।

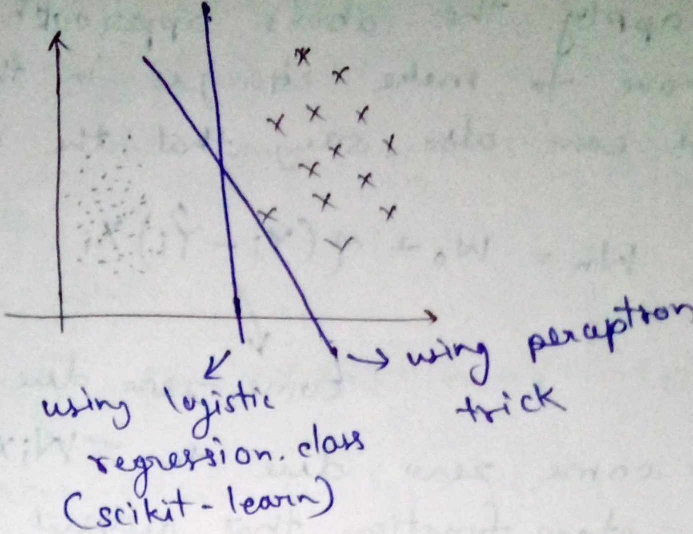
$W_{new} = W_{old} + \eta X_i$ — (ii)

simplified formula:

$W_{new} = W_{old} + \eta (Y_i - \hat{Y}_i) X_i$ — (iii)

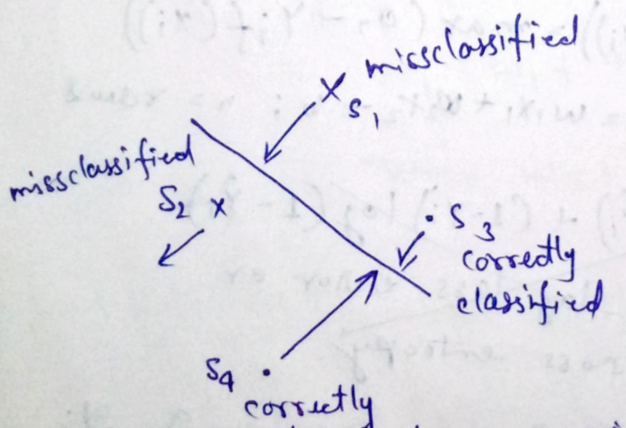
	Actual Y_i	prediction $\hat{Y}_i$	$Y_i - \hat{Y}_i$
case-1	1	1	0
case-2	0	0	0
case-3	1	0	1
case 4	0	1	-1

in case-1 and case-2 No updates in weight but in case-3 and case-4 weight equation becomes eqn (ii) and (i)



- if we want the perceptron trick model is work as the logistic regression class then here we also considered the correctly classified point also ~~मतलब~~ correctly classified point line ~~को अपने से दूर भेजेंगे~~ Till the equilibrium phase.

  missclassified – line pull
  correctly classified – line push!



in missclassified:
- a point has less distance (S_1)
  . less magnitude to pull
- a point have large distance (S_1) large magitude to pull

vice - versa for also
correctly classified point

* for all of thess above approach our alg. (perceptron trick) is work similar as the logistic regression class

- if we apply the above approach then our plam have to make changes in the equation (iii) or we can also say that the case-3 and case-4

$$W_n = W_o + \eta (Y_i - \hat{Y}_i) X_i$$

come zero due to this

- $Y_i - \hat{Y}_i$ is come zero due to $\Sigma W_i X_i$ this function. that is step function that output always the 0 or 1 hence now we have to use the sigmoid function.

Loss function in perceptron:

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

→ Regularisation.

stochastic gradient descent general function.

Here, L = loss function and R = Regularization

in perceptron: $L(Y_i, f(x_i)) = \max(0, -Y_i f(x_i))$

Here, $f(X_i) = w_1 X_1 + w_2 x_2 + b$; n = rows

$$L = -\frac{1}{n} \sum_{i=1}^{n} Y_i \log(\hat{Y}_i) + (1 - Y_i) \log(1 - \hat{Y}_i)$$

the eqn called log loss error or Binary cross entropy.

- cross entropy मे हम values को minimize करते है मतलब वह model Best होगा जिसका cross entropy minimum होगा,

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

Hing loss function

in $\max(0, -Y_i f(x_i))$

if $-Y_i f(x_i) \geq 0$

then $\max(0, -Y_i f(x_i)) = -Y_i f(x_i)$

else. 0

for two point

Loss function $= \frac{1}{2}\left[\max(0, -Y_1 f(x_1)) + \max(0, -Y_2 f(x_2))\right]$

Here $f(x_2) = W_1 x_{21} + W_2 x_{22} + b$

Gradient descent (optimization Algo):

$$L = \underset{W_1, W_2, b}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \max(0, -Y_i f(x_i))$$

where, $f(x_i) = W_1 x_{i1} + W_2 x_{i2} + b$

for i in epochs:

randomly select the w_1, w_2, b, $w_1 = 1, w_2 = 1, b = 1$

$$W_1 = W_1 + \eta \frac{\partial L}{\partial W_1}$$

$$W_2 = W_2 + \eta \frac{\partial L}{\partial W_2}$$

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

now calculate $\left[\frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}, \frac{\partial L}{\partial b}\right]$ where $\eta = 0.01$

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

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

$$\boxed{\begin{aligned} \frac{\partial L}{\partial w_1} &= \begin{cases} 0 & \text{if } Y_i f(x_i) \geq 0 \\ -Y_i X_{i1} & \text{if } Y_i f(x_i) < 0 \end{cases} \\ \text{similarly} \quad \frac{\partial L}{\partial w_2} &= \begin{cases} 0 & \text{if } Y_i f(x_i) \geq 0 \\ -Y_i X_{i2} & \text{if } Y_i f(x_i) < 0 \end{cases} \\ \frac{\partial L}{\partial b} &= \begin{cases} 0 & \text{if } Y_i f(x_i) \geq 0 \\ -Y_i & \text{if } Y_i f(x_i) < 0 \end{cases} \end{aligned}}$$

More Loss function:

perceptron is very flexible because here we can use other loss function. like sigmoid function

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

(Binary cross entropy) $L = -Y_i \log \hat{Y}_i + (1-Y_i)\log(1-\hat{Y}_i)$

perceptron == logistic Regression only when
Activation function is sigmoid and
loss function is binary cross entropy

Loss function	Activation	output
Hing loss	Step	perceptron → binary classification (-1, 1)
log-loss (Binary cross entropy)	sigmoid	logistic Regression 0 to 1 (binary classify)
categorical cross entropy $L = \sum_{j=1}^{M} y_i \log(\hat{y}_i)$	softmax $f = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$	softmax Regression (multiclass classification)
mean squared error	linear	~~linear~~ linear Regression (number)

SGD

problem with Perceptron:
- only work on linear data.