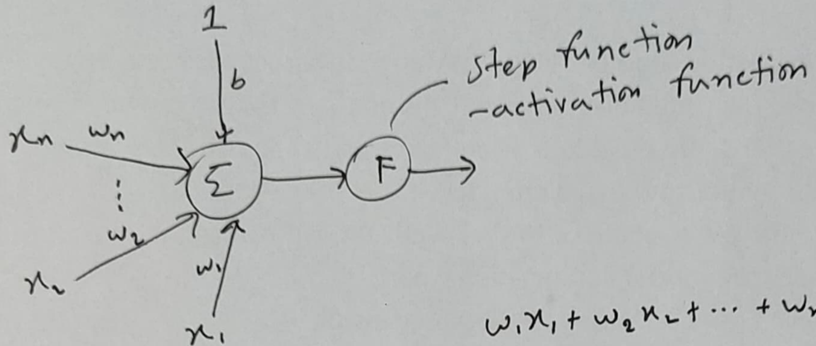
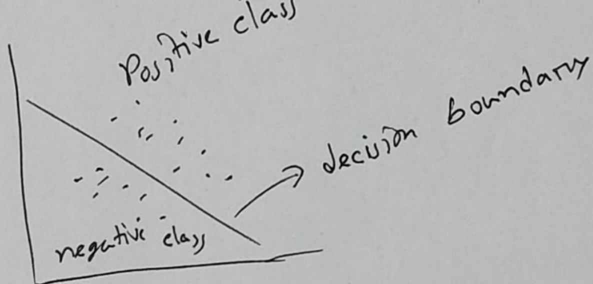
CSE 465 / L-05 / 02.02.2025 /

Perceptron training



$$w_1x_1 + w_2x_2 + \dots + w_nx_n + b$$

$$= \sum_{i=1}^{n} w_ix_i + b$$



Decision boundary function/equation:

$$Ax + By + C = 0$$

Variable that move the line.

A, B = weight
C = bias
} we need to find the right value of them.

Learning process:

- start with random line
- then update the value and shift the line.
  - based on data point.

⊛ $2x + 3y + 5 = 0$

$P(4,2) \in N$

$\sum x_i w_i > 0$ for this line. so prediction is wrong.

Data point

$$\begin{array}{rrrl} & 2 & 3 & 5 \longrightarrow \text{co-efficient of the equation} \\ (-) & 4 & 2 & 1 \longrightarrow \text{always 1} \\ \hline & -2 & 1 & 4 \longrightarrow \text{new line} \end{array}$$

⊛ $2x + 3y + 5 = 0$

$P(-3,-2) \in P$

$\sum x_i w_i < 0$, prediction is wrong.

$$\begin{array}{rrrr} & 2 & 3 & 5 \\ (+) & -3 & -2 & 1 \\ \hline & -1 & 1 & 6 \end{array}$$

⊛ learning algorithm - 1 : $\boxed{\text{slide-10}}$

slow down the learning process:

algorithm-2 : $\boxed{\text{slide-11}}$

$0 < \eta \Rightarrow$ eta < 1

Final Combined algorithm

Algorithm-3 $\boxed{\text{slide-12}}$

⇒ This algorithm can't find out the best fit line for new data.

if η value is small, updates takes too long

if large, can oscillate and never converge.

L-06/04.02.2025/

Loss function:

- mathematical function that measure how well a machine learning model's predictions match the actual values.

- evaluates the model performance
- guides optimization algorithms
- smaller loss mean a better model.

SGD-sklearn:

$$E(\omega, b) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \underbrace{f(x_i)}_{\substack{\text{Predicted} \\ \hat{y}}}) + \alpha R(\omega)$$

actual y

regularization

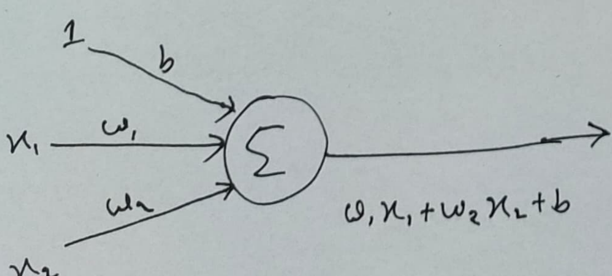
Perceptron

$\hat{y}_i = f(x_i)$

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

$$\underbrace{L(\omega_1, \omega_2, b)}_{\text{weight \& biases}} = \frac{1}{2} \sum_{i=1}^{2} \max(0, -y_i f(x_i))$$

just for convenience with integration



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

Slide - 21

Gradient Descent

$\cancel{\hat{y} = h(x) = ax}$

Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Parameters: θ_0, θ_1

Cost function:

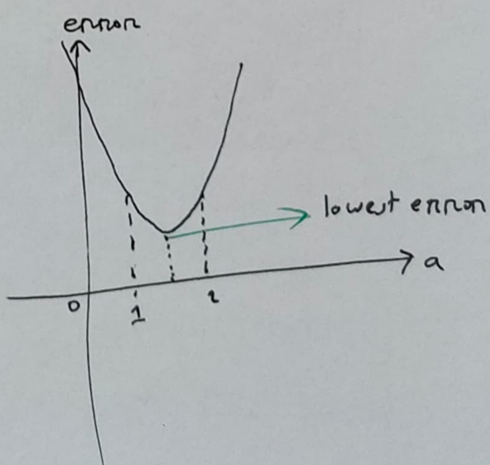
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x_i) - y_i \right)^2$$

Target: $\underset{\theta_0, \theta_1}{\text{minimize}} \; J(\theta_0, \theta_1)$

$\hat{y} = h(x) = ax$

						error $(y_i - \hat{y}_i)^2$
x	1	2	3	4	5	
y	2	5	5	10	11	
$a = 2$	2	4	6	8	10	7
$a = 3$	3	6	9	12	15	38
$a = 4$	4	8	12	16	20	179
$a = 1$	1	2	3	4	5	86
$a = 0$	0	0	0	0	0	275

a	0	1	2	3	4
E	275	86	7	38	179



Algorithm:

Get optimal:

i. choose a random value for a

ii. while (not happy):

compute error, E

compute $\frac{dE}{da}$

$$\boxed{\begin{aligned} step &= \begin{cases} -1 & : \frac{dE}{da} > 0 \\ 1 & : \frac{dE}{da} < 0 \end{cases} \\ a &= a + step \end{aligned}} \Rightarrow a = a - \eta \frac{dE}{da}$$

η → hyper parameter

a → parameter

check my happiness status

a	Error	$\frac{dE}{da}$
4	$\sum_{i=1}^{5} (y_i - \hat{y}_i)^2 = 179$	196 (+)
		~~24~~ 86 (+)
3	38	
		-24 (-)
2	7	

Solution: $step \propto \left| \frac{dE}{da} \right|$

$\therefore step = \eta \frac{dE}{dA}$

η → learning rate

$$\frac{dE}{da} = 2 \sum_{i=1}^{5} (y_i - ax_i)(-x_i)$$

$$= 2 \sum_{i=1}^{5} (ax_i - y_i) x_i$$

$$E = \sum_{i=1}^{5} (y_i - \hat{y}_i)^2$$

$$E = \sum_{i=1}^{5} (y_i - ax_i)^2$$

$$\hat{y} = a + bx$$

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2$$

$$= \frac{1}{m} \sum_{i=1}^{m} (a + bx_i - y_i)^2$$

$$\frac{d}{da} J(a,b) = \frac{2}{m} \sum_{i=1}^{m} (a + bx_i - y_i) \cdot 1$$

$$\frac{d}{db} J(a,b) = \frac{2}{m} \sum_{i=1}^{m} (a + bx_i - y_i) x_i$$

Repeat ~~on~~ until converge:

$$\{$$

$$a = a - \alpha \frac{2}{m} \sum_{i=1}^{m} (a + bx_i - y_i)$$

$$b = b - \alpha \frac{2}{m} \sum_{i=1}^{m} (a + bx_i - y_i) x_i$$

$$\}$$

H.W.

Findout gradient decent function for perceptron.

L-08/11.02.2025/

Algorithm:

repeat until converge:

$$\{$$

$$\theta_i = \theta_i - \alpha \frac{\delta L}{\delta \theta_i}$$

$$\}$$

⊛ For the gradient decent problem:

$$h_\theta(x) = ax^2 + bx + c$$

$$\theta = (a, b, c)$$

$$L = \frac{1}{m} \sum_{i=1}^{m} (ax_i^2 + bx_i + c - y_i)^2$$

$$\therefore \frac{dL}{da} = \frac{2}{m} \sum_{i=1}^{m} (ax_i^2 + bx_i + c - y_i)\, x_i^2$$

$$\therefore \frac{dL}{db} = \frac{2}{m} \sum_{i=1}^{m} (ax_i^2 + bx_i + c - y_i)\, x_i$$

$$\therefore \frac{dL}{dc} = \frac{2}{m} \sum_{i=1}^{m} (ax_i^2 + bx_i + c - y_i) \cdot 1$$

⇒ Therefore, algorithm:

start with ~~ab~~ a, b, c :

repeat until converge :

{

$$a = a - \frac{2\alpha}{m} \sum_{i=1}^{m} (ax_i^2 + bx_i + c - y_i)\, x_i^2$$

$$b = b - \frac{2\alpha}{m} \sum_{i=1}^{m} (ax_i^2 + bx_i + c - y_i)\, x_i$$

$$c = c - \frac{2\alpha}{m} \sum_{i=1}^{m} (ax_i^2 + bx_i + c - y_i)$$

}

For Perceptron:

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

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

row

number of features

$$\theta = (w_1, w_2, b)$$

$\Rightarrow \frac{dL}{dw_1}$:

$$\frac{dL}{dw_1} = \frac{dL}{df(x_i)} \times \frac{df(x_i)}{dw_1}$$

$$L = \begin{cases} 0 & ; -y_i f(x_i) \leq 0 \\ \sum_{i=1}^{n} -y_i f(x_i) & ; -y_i f(x_i) > 0 \end{cases}$$

$$\therefore \frac{dL}{df(x_i)} = \begin{cases} 0 & ; -y_i f(x_i) \leq 0 \\ \frac{1}{n} \sum_{i=1}^{n} -y_i & ; -y_i f(x_i) > 0 \end{cases}$$

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

$$\therefore \frac{df(x_i)}{dw_1} = x_{i1}$$

if $y_i f(x_i) \geq 0$

$$\frac{dL}{dw_1} = 0$$

else

$$\frac{dL}{dw_1} = \frac{1}{n} \sum_{i=1}^{n} -y_i x_{i1}$$

as well as,

$$\frac{df(x_i)}{dw_2} = x_{i2}$$

if $y_i f(x_i) \geq 0$

$$\frac{dL}{dw_2} = 0$$

else

$$\frac{dL}{dw_2} = \frac{1}{n} \sum_{i=1}^{n} -y_i x_{i2}$$

Now,

$$\frac{df(x_i)}{db} = 1$$

$$\therefore \frac{dL}{db} = \frac{dL}{df(x_i)} \times \frac{df(x_i)}{db}$$

$\therefore$ if $y_i f(x_i) \geq 0$

$$\frac{dL}{db} = 0$$

else

$$\frac{dL}{db} = \frac{1}{n} \sum_{i=1}^{n} -y_i$$

✲ Question Pattern:

3 weight and one bias will be given, find out the update rule.

$\Rightarrow$ therefore, final algorithm:

we will only update if,

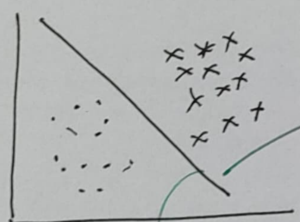
$$-y_i f(x_i) < 0$$

$$w_1 = w_1 + \alpha \frac{1}{n} \sum_{i=1}^{n} y_i x_{i1}$$

$$w_2 = w_2 + \alpha \frac{1}{n} \sum_{i=1}^{n} y_i x_{i2}$$

$$b = b + \alpha \frac{1}{n} \sum_{i=1}^{n} y_i$$

✲ Example:



linearly separable

$y = mx + c$

original equation:

$$w_1 x_1 + w_2 x_2 + b = 0$$

it was plotted on y axis

$$w_2 x_2 = -b - w_1 x_1$$

$$\therefore x_2 = \underbrace{\boxed{-\frac{w_1}{w_2}}}_{m} x_1 \underbrace{\boxed{-\frac{b}{w_2}}}_{c}$$

AND:

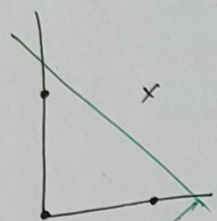
0 0	0
0 1	0
1 0	0
1 1	1

OR:

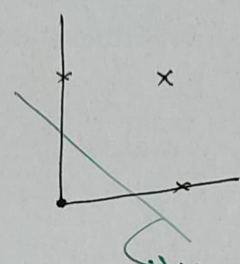
0 0	0
0 1	1
1 0	1
1 1	1

XOR:

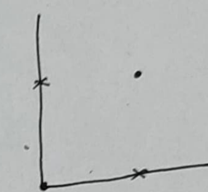
0 0	0
0 1	1
1 0	1
1 1	0



linearly separable



linearly separable



⇒ linearly not separable
- can't train perceptron with this data.

Tensorflow playground:
- for visualise machine learning model.