Machine Learning Lect 3

Gradient Descent

Dr. Prerana Mukherjee

9 Aug '20

Ques AND function implementation using Single Layer Perceptron

Truth table of AND

x_1	x_2	t
1	1	1
1	0	0
0	1	0
0	0	0

x_1	x_2	t
1	1	1
1	-1	-1
-1	1	-1
-1	-1	-1

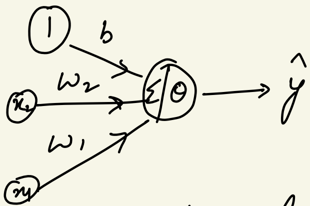
0 → -1 (Label)

1 → 1 (Label)

$y_{in} = \sum_{i=0}^{n} w_i x_i$

Error $= t - \hat{y}$

Perceptron N/W



$\eta = \alpha = 1$

some value b/w 0 to 1

Activation function $\theta = 0$

$$\theta(y_{in}) = \begin{cases} 1 & y_{in} > 0 \\ 0 & y_{in} = 0 \\ -1 & y_{in} < 0 \end{cases}$$

Epoch:

Initialize weights = 0

$w_1, w_2, b = 0$

i)

$y_{in} = \sum w_i x_i + b$

$= b + w_1 x_1 + w_2 x_2 = 0 + 0(1) + 0(1)$

$= 0$

$\hat{y} = \theta(y_{in}) = 0$

$\hat{y} \neq t \Rightarrow$ update weights

$w_i(\text{new}) = w_i \text{ old} + \alpha t x_i$

$\Delta w = \eta (t - \hat{y}) x_i$

$1 - 0$

$$w_{1new} = w_{1old} + \alpha t x_1$$
$$= 0 + 1 \cdot 1 \cdot 1 = 1$$

$$w_{2new} = w_{2old} + \alpha t x_2$$
$$= 0 + 1 \cdot 1 \cdot 1 = 1$$

$$b_{new} = b_{old} + \alpha t$$
$$= 0 + 1 \cdot 1 = 1$$

ii) $$y_{in} = b + w_1 x_1 + w_2 x_2$$
$$= 1 + (1)(1) + (1)(-1)$$
$$= 1 + 1 - 1 = 1$$

$$\hat{y} = \theta(y_{in}) = 1$$

$\hat{y} \neq t \Rightarrow$ update weights

$$w_{1new} = w_{1old} + \alpha (t - \hat{y}) x_1$$
$$= 1 + 1(-1 - 1) \cdot 1$$
$$= 1 - 2 = -1$$

$$w_{2new} = w_{2old} + \alpha (t - \hat{y}) x_2$$
$$= 1 + 1(-1 - 1) - 1$$
$$= 1 + 2 = 3$$

$$b_{new} = b_{old} + \alpha (t - \hat{y})$$
$$= 1 + 1(-2) = -1$$

iii) $$y_{in} = b + w_1 x_1 + w_2 x_2$$
$$= -1 + (-1)(-1) + (3)(1)$$
$$= \cancel{-1} + \cancel{1} + 3 = 3$$

$$\hat{y} = \theta(y_{in}) = 1$$

$\hat{y} \neq t \Rightarrow$ update weights

$$w_{1\,new} = w_{1\,old} + \alpha(t - \hat{y})\,x_1$$
$$w_{1\,new} = -1 + 1(-1 - 1)\,-1$$
$$= -1 + 2 = 1$$

$$w_{2\,new} = w_{2\,old} + \alpha(t - \hat{y})\,x_2$$
$$= 3 + 1(-1 - 1)\,1$$
$$= 3 - 2 = 1$$

$$b_{new} = -1 + 1(-2) = -1 - 2$$
$$= -3$$

(iv) $y_{in} = -3 + (1)(-1) + 1(-1)$

$$= -3 - 1 - 1 = -5$$

$\hat{y} = -1$

$\hat{y} = t \Rightarrow$ no change in weight

Final weights are

$w_1 = 1 \quad w_2 = 1 \quad b = -3$

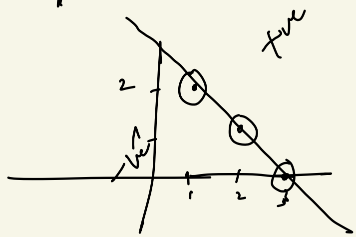
Decision boundary

$$b + w_1 x_1 + w_2 x_2 = 0$$
$$-3 + x_1 + x_2 = 0$$
$$x_1 + x_2 = 3$$

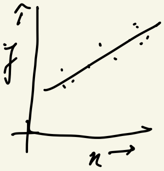
$x_1 = 0 \quad x_2 = 3$

$x_1 = 1 \quad x_2 = 2$

$x_1 = 2 \quad x_2 = 1$



Gradient Descent

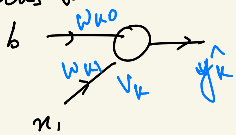


$y = f(x)$

fit a straight line given a set of observations

Linear Regression $y = f(x)$

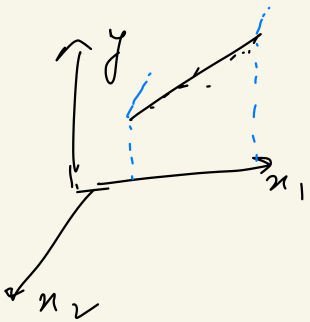
Single neuron which works for linear activation unit $v_k = \hat{\sum_{j=0}} w_{kj} x_j = \hat{y}_k$

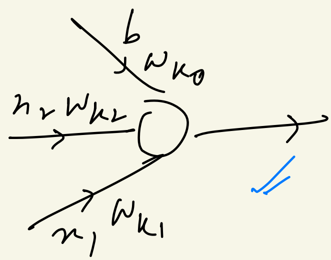


2 parameters: slope & intercept

$y = f(x_1, x_2)$

line fitting → 3D line fitting

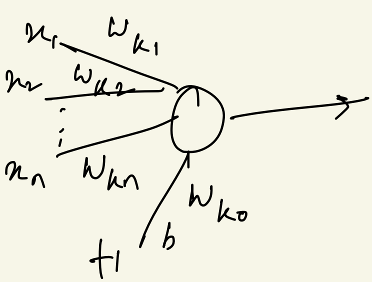




2 slopes components along $x_1 y$ plane w_{k1}
$x_2 y$ plane w_{k2}

+ 1 intercept

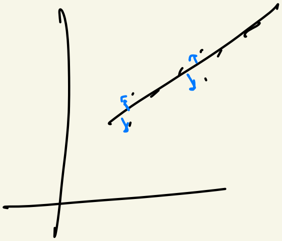
$y = f(x_1, x_2 \dots x_n)$



$y = \sum w_{kj} \cdot x_j$

n-d straight line

So many gradients along individual planes + bias

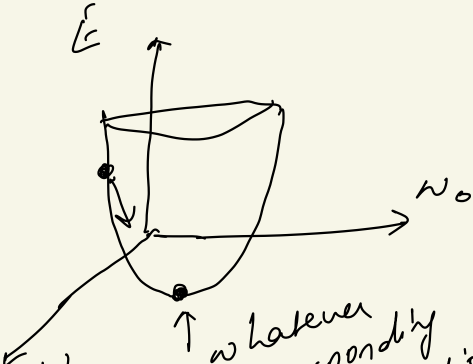


not passing exactly through points

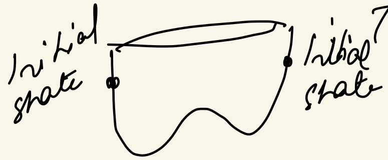
Error values some positive & some negative $\Rightarrow$ combined error can be plot as a function of 2 variables w_1 & w_0
slope intercept

natural tendency is to reach the minima (minimum error)

fit a curve which gives min error. searching all over the place for best combination of parameters (w_1 & w_0)



↑ whatever corresponding position of w_1 & w_0 at this point (best solution)



These 2 minimas could be of different magnitudes out of the several minimas the surface can exhibit.

Only 1 global minima rest are local minimas.

System adopts & learns to reach best fitment & in the process it reaches global minima or gets trapped at local minima.

Gradient point up but the point moves opposite to the direction of the gradient.

Let us have 'p' : no. of observations

$$E = \sum_p E^p$$

$$E^p = \frac{1}{2} \sum_o \left(t_o^p - y_o^p \right)^2 \quad \text{— (1)}$$

Desired Actual

no of different O/Ps

individual output

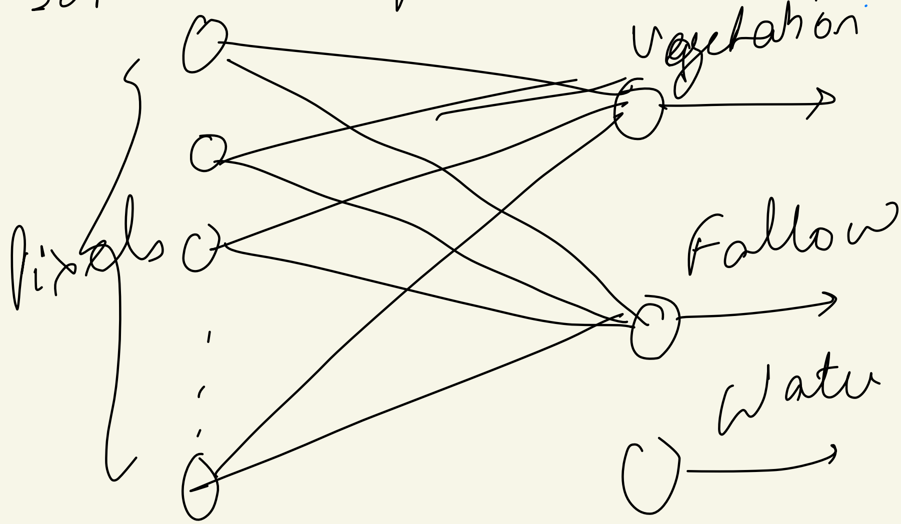
$$y_0 = f_1(x_1, x_2 \ldots x_n)$$

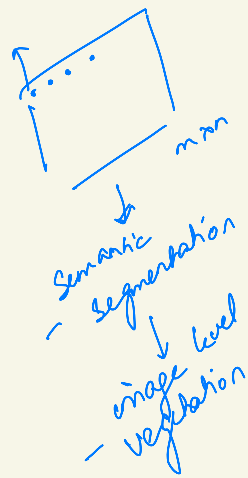
$$y_1 = f_2(x_1, x_2 \ldots x_n)$$

$$\vdots$$

$$y_n = f_n(x_1, x_2, \ldots x_n)$$

Satellite image





$$G = \frac{\partial E}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \sum_p E^p$$

$$= \sum_p \frac{\partial E^p}{\partial w_{ij}}$$

Chain rule

$$\frac{\partial E}{\partial w_{oi}} = \frac{\partial E}{\partial y_o} \cdot \frac{\partial y_o}{\partial w_{oi}}$$

From 1

$$\frac{\partial E}{\partial y_o} = -(t_o - y_o) \quad \cdots \text{ 2}$$

$$y_o = \sum_j w_{oj} x_j$$

$w_1 x_1 + w_2 x_2 + \dots w_{oi} x_i + \dots w_n x_n$

$$\frac{\partial y_o}{\partial w_{oi}} = \frac{\partial}{\partial w_{oi}} \sum_j w_{oj}\, x_j$$

$$= x_i \quad \text{— } (3)$$

$$\frac{\partial E}{\partial w_{oi}} = -\,(t_o - y_o)\, x_i$$

w.r.t. particular weight

input

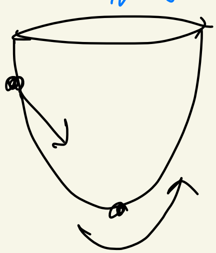
$\Delta w = -\eta \frac{\partial E}{\partial w_{oi}} = \eta (t_o - y_o) \cdot x_i$

$(- -) \Rightarrow +ve$

Error term $\Delta w_{oi} = \eta (t_o - y_o) \cdot x_i$

rate at which it is falling

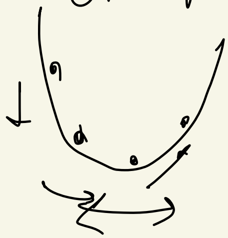
$$w_{oi} = w_{oi} + \Delta w_{oi}$$



η → high it is faster (might cross the minima)
else slower

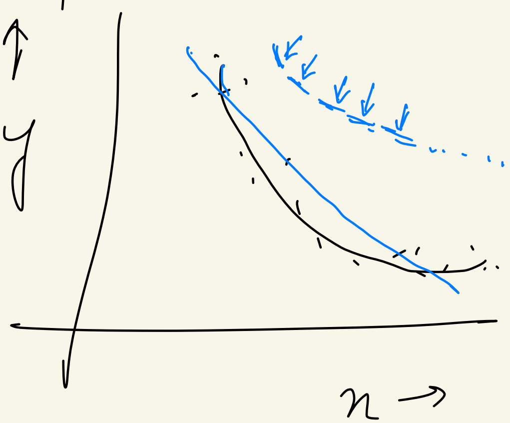
linear n-d straight line fitting

neural n/w → linear activation fn



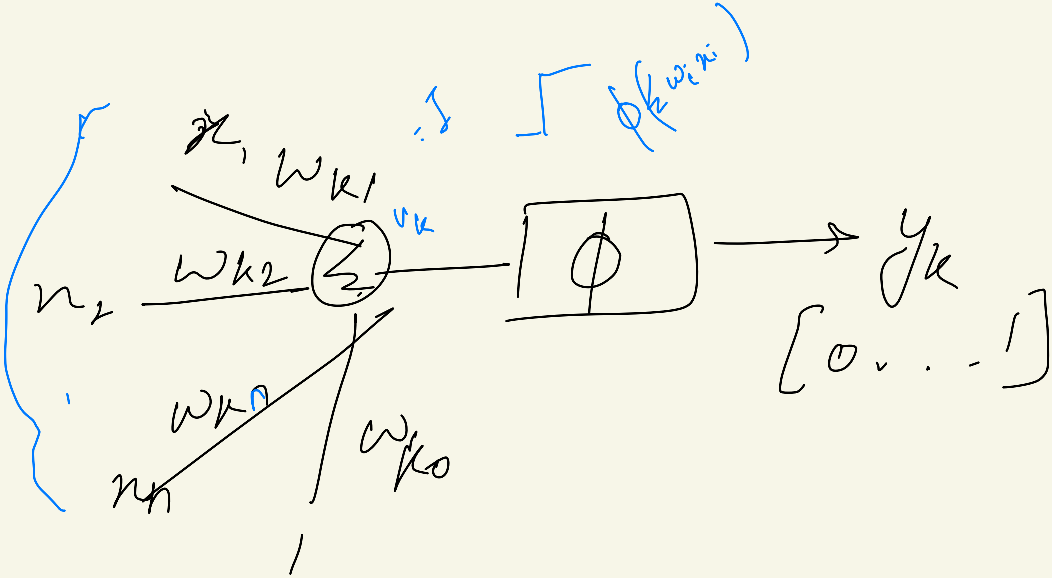
oscillate to reach minima

0.1



non-linear curve fitting
↓
break it into piecewise linear components
↓
complicated process

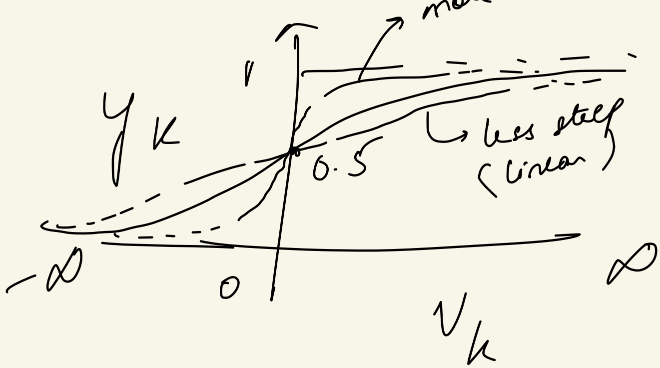
Soln: non-linear model



$$V_k = \sum_{j=0}^{n} x_j \cdot w_{kj}$$

$y_k = \phi(v_k)$

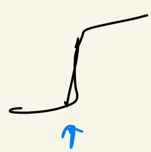
$y = f(x)$ iff $x\uparrow$ $y\uparrow$



i) monotonically increasing

ii) continuously differentiable fn.

(McCulloc & Pitts fn.



$\Rightarrow$ not continuous differentiable

Tunability : ~~shape~~ control

$$\phi(v_k) = \frac{1}{1+e^{-av_k}} \quad \text{(Sigmoid)}$$

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

Quotient rule:

$$\sigma'(z) = \frac{(1+e^{-z})\cdot 0 - (1)(-e^{-z})}{(1+e^{-z})^2}$$

$$= \frac{e^{-z}}{(1+e^{-z})^2} = \frac{1-1+e^{-z}}{(1+e^{-z})^2}$$

$$= \frac{1+e^{-z}}{(1+e^{-z})^2} - \frac{1}{(1+e^{-z})^2}$$

$$= \frac{1}{(1+e^{-z})} - \frac{1}{(1+e^{-z})^2}$$

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

$$= \sigma(z)\,(1-\sigma(z))$$