m - estimate : $\frac{n_c}{n}$ se change $\frac{n_c + mp}{n + m}$ probability of an attribute

But agar koi exist nhi karta, toh zero ho jaye ga

Laplace smoothing : kehtay uniform probability hai (when we have no proper knowledge of underlying distribution)

$P(\dots) = \frac{n_c + 1}{n + k}$ to avoid ÷ by zero

ab continuous values keh liye kaise karein :

→ use Gaussian Naive Bayes :

univariate gaussian $P(X|y; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{\left(\frac{-1}{2}\frac{(x-\mu)^2}{\sigma^2}\right)}$

weights ko kalnay keh liye

$P(X|y); \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{\left(\frac{-1}{2}\frac{(x-\mu_y)^2}{\sigma^2}\right)}$

if conditional independence nhi kartay toh wo naive nhi.

Date: __/__/20__

Probability of Play tennis

<Sunny, cool, high, strong>

n=4, t=2

n x t = 8, 2 priors (Prob of yes & prob of no)

labels can be on right

$P(Yes) = 9/14$

$P(NO) = 5/14$

~~P(Sunny~~

$P(Yes) \rightarrow P(Sunny|Yes) \cdot P(cool|Yes) \cdot P(high|Yes) \cdot P(strong|Yes)$

0.005

$P(No) \cdot P(Sunny|NO)$

0.0206

take max ↓

if category mangen toh

$P(NO|<>) = \frac{0.0206}{0.0206+0.005} = \frac{2}{9}$

but probability mangein toh divide karein by sum of

$P(NO) \cdot P(Sunny|NO)$

$5/14)$

Date: __/__/20__

MON TUE WED THU FRI SAT

$$P(a_1, \ldots a_n) = \sum_{j=1}^{k} \prod_{i=1}^{n} P(a_i | v_j) \cdot P(v_j)$$

$$V_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j | a_1 \ldots a_n) \; P(v_j)$$

↳ classes kitni count such find

$$P(V_{map}) = \underset{v_j \in V}{\operatorname{argmax}} P(a_1, \ldots a_n | v_j) \cdot P(v_j)$$

$P \; 2^n \rightarrow$ (for binary classification)

kitnay possibilities

parabolic hai

$$P(a_1, a_2 \ldots a_n, v_j)$$

$$= P(a_1 | a_2, \ldots a_n, v_j) \cdot P(a_2 \ldots a_n, v_j)$$

$$= P(a_1 | a_2 \ldots a_n, v_j) \cdot P(a_2 | a_3 \ldots a_n, v_j)$$

$$= P \; P(a_3 | a_1 \ldots a_n, v_j) \ldots P(a_{n-1} | a_n, v_j)$$

$$\cdot P(a_n | v_j) \cdot P(v_j)$$

V_{MAP} Conditional independence hai as the class label

$$P(a_1 | v_j) = P(a_1 | v_j, a_2, \ldots a_n)$$

V_{MAP} ↓ very high probability

$n \times$ no. of class → no. of attributes × targets

$$V_{NB} = \underset{v_j}{\operatorname{argmax}} P(a_1 | v_j) \ldots P(a_n | v_j) * P(v_j)$$

$$V_{map} = \underset{v_j}{\operatorname{arg\,max}} P(v_j) \cdot \prod_{i=1}^{n} P(a_i | v_j)$$

Bayesian rule lagaenge qki → gen model kehlaata hai

$$P(y|x) = \frac{P(x|y) \cdot P(y)}{P(x)}$$

→ per(x)

discriminative likelihood model kaun hai?

Naive Bayes Classifier

→ generative model

Posterior Prob

$$V_{map}^{label} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j | a_1, \dots, a_n)$$

(6th chapter)

before bayes rule

a1	a2	...	an	vj

a_0 yahan pe nhi add hoga

j is class keh liye max no. aaye wo class hai

→ after bayes rule

$$P(v_j | a_{1,\dots,n}) = \underset{v_j \in V}{\operatorname{argmax}} \frac{P(a_1, \dots, a_n | v_j) \cdot P(v_j)}{P(a_1 \dots a_n)}$$

↳ dont need

normalize karnay keh liye

V_{map} → class hai toh can't decide label

Date:__/__/20__ MON TUE WED THS FRI SAT

divide data into sets
(k fold cross-validation)

avg performance
kahan validation error increase

avg no. of epochs, uss par train kalein.

Naive Bayes Classifier

goal of ML supervised learning?
decision boundary determine karni

$f : x \rightarrow y$

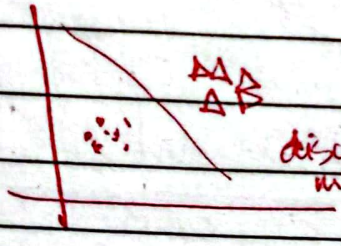
$h_\theta(\theta^T x)$ → 1
→ 0

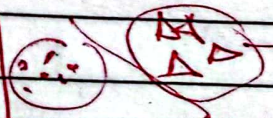
Discriminative Models

- Models that find boundaries
- trying to find class label y given data.

$P(y|X)$ (directly learn karta)

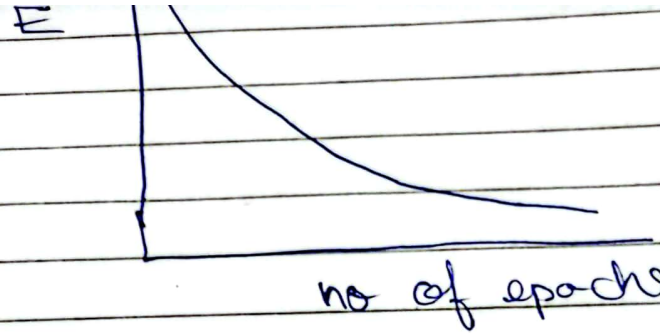
Indirectly? → generative models
↳ they learn $P(X|y)$ & $P(y)$





→ feature kya, prior prob?

Generative model $P(X|y)$ & $P(y)$

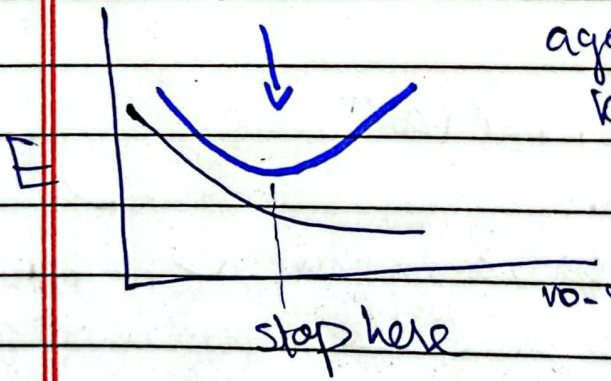


model ka iss kabil nhi
training mein sahi na chalein
→ high bias

testing error high
→ high variance
(may infer we have done overfitting)

find a sweet spot b/w high bias & high variance

validation error

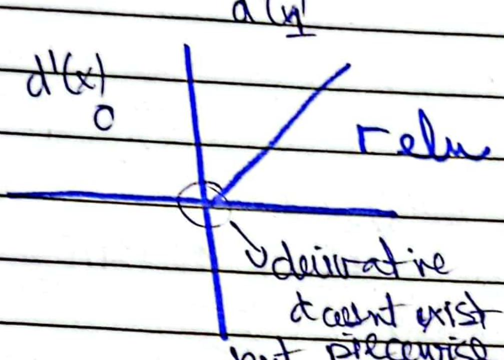


agar ismein bhi kam ho raha error

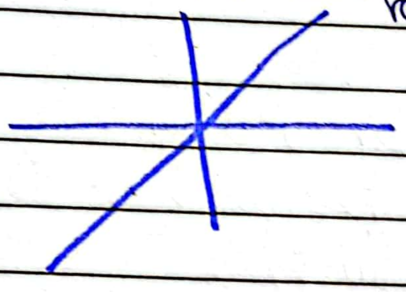
data is an asset
agar kam data toh do:
Cross-validation

iterations ziyada toh minimize with time

plot heuristic for lr to see where error bada (ashkay heuristic)

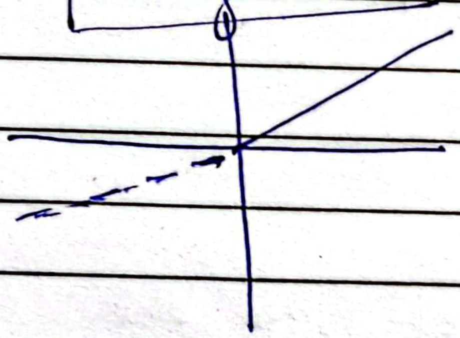


jau function use karletay haga, uska derivative haga



bottleneck nhi hati

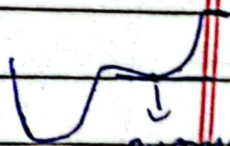
leaky relu



EBPA ka kahan rokna:

stopping condition:

- error gradient kam ho jayein



- epoch ↑
- weight change ziyada nhi ho rahi

high-bias

model itna complex nhi keh it could understand your data

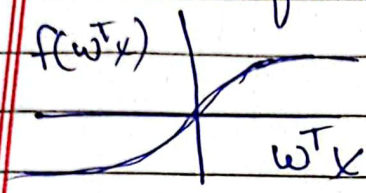
★ - validation keh zariye control karein

Date: __/__/20__

O mean karein

(v) weights ki initialization

rules for initialization



keep weights in range
kch +, - bhi
ho & choti range
ho

eg:
[-1.5, 1.5]
avoid 0

ng bahut zyada
dps likhein

agar bahut small
hoa (near 0)
toh not activation
mein time lagayega

Std of weight $\sigma_w = \frac{1}{\sqrt{m}}$

m → no. of connections to a node

Scikit-learn → documentation check

(vi) learning constant

learning constant ko variable
banaeb
change kr:
outer mein bada,
inner mein chota

Date: __/__/20__ MON TUE WED THS FRI SAT

classes ki tadad = no. of nodes in O/P Layer

binary classification 0 1 0
0 0 1

one hot encoding

binary encoding

input layers = no. of features nodes

hidden layer nodes = b/w I/P & O/P ki range (fixed formula nhi) hum zyada nhi rakhein, overfitting ho jayegi.

generalization behtar ho jati : generalized performance judge ki ja rahi

(iv) Scaling of inputs

use scaling with GD.

scaling kesay karni, numerical scaling, text ko bhi numericals lani pahegi

z-score scaling

min-max scaling (required : decorelated hon) eg: correlated weight & mass

L3 network ki scalability hogi

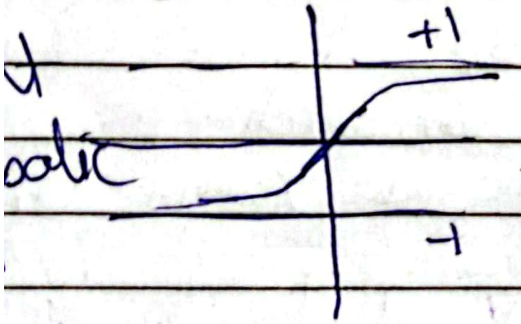
apnay previous weight change ko
is t per weight change calculate
kar rahay

$$\Delta w_t \leftarrow \Delta w_t + \beta \Delta w_{t-1}$$

↓ momentum factor (is a fraction) $[0,1)$ hyperparameter 0.85, 0.9

(ii) Antisymmetric O/P function in hidden nodes → $f(-x) = -f(x)$

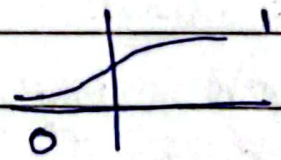
dont use sigmoid.



benefits:
use this
→ symmetric about origin
→ linear range bahi
→ want to avoid saturated ranges
→ we create +1 & -1

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$f'(x) = 1 - (f(x))^2$$



saaray positive in sigmoid

how many nodes in output in MLP
→ o/p → classifier sch aati

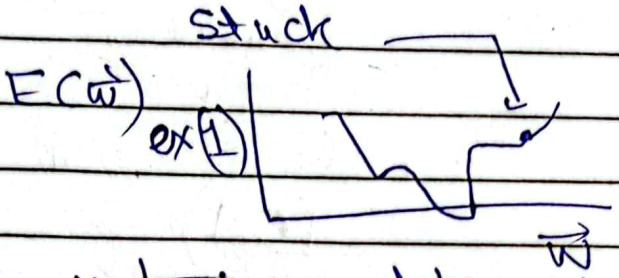
Date: __/__/20__ MON TUE WED THU FRI SAT

batch mein hum phasein gay

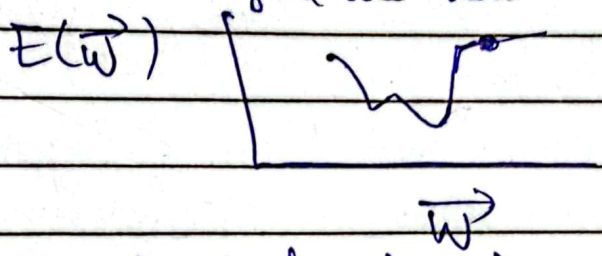
Heuristics

(i) SGD

ho sakta aik error surface, we get stuck



but ho sakta ex 2 keh hisaab sey



complicated network error so SGD is good in this situation

for batch

but agar error func use karna, toh skip GD & use:

Newton's method

use network error wrt one ~~error~~ training example

(ii) momentum

- adam? solver
- line search
  - ↳ improve method: conjugate gradient descent

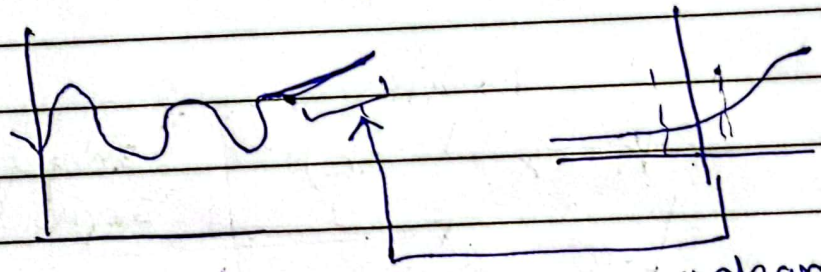
Date: __/__/20__

error surface:
error ko plot karay wrt weight.

hyper surface
3D. ← more than 3D ka sochna
think of it as 3D with billion parameters

Why we take different error surface:

→ error surface flat nhi



non-convex optimization yahan ka smooth
model acha as it works, generate
and is non-linear and kar rahe
has alot of free parameters

error function derived in class wrt weight that we minimized

$$E(w) = \frac{1}{2} \sum_k (t_k - o_k)^2$$

network error represent kar raha hai:

$$E(w) = \frac{1}{2N} \sum_{i=1}^{N} \sum_k (t_k - o_k)^2$$

Date: 31/12/2024

Naive Bayes Classifier

review it today

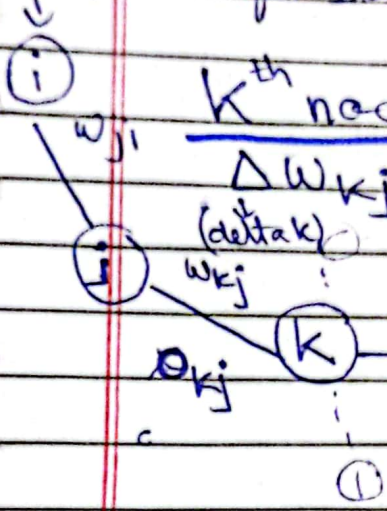
error backpropogation:

Heuristics to improve the performance of EBPA:

(It is basically a discriminative classifier)

(for stochastic implementation)

Single training set kehliye hai



k^{th} node

$\Delta w_{kj} = \eta \cdot \delta_k \cdot O_{kj}$

(δ_k = delta k)

small (δ_k)

error (local gradient error)

$\delta_k = e_k \cdot O_k (1 - O_k)$ (only for sigmoid function)

$O_k(1-O_k)$ → derivative

e_k → error term

$(d_k - O_k)$

↳ error derivation keh baad nikala, no need to square

j^{th} node

→ here we can't find it directly as it has no desired output at hidden node

$\Delta w_{ji} = \eta \cdot \delta_j \cdot x_{ji}$

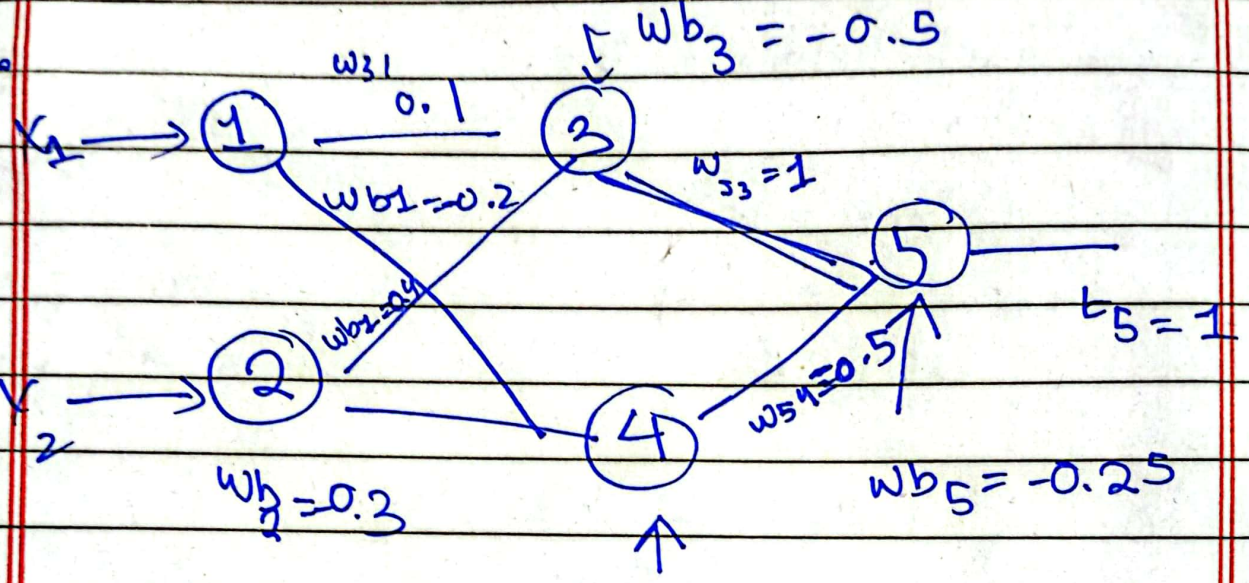
$\delta_j = e_j \cdot O_j (1 - O_j)$

$e_j = \sum_k \delta_k \cdot w_{kj}$

Date:__/__/20__

MON TUE WED THS FRI SAT

Q.

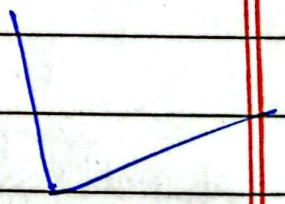


$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$wb_{x4} = 0.3$

(i) $\Delta w_{53} = ?$

(ii) $\delta 4 = ?$



cost

creates non-convex function

inefficiency (it doesn't align with the probability

l

DW it

2. For each hidden node:

$$\Delta w_{ji} = \eta \delta_j . x_{ji}$$

hidden node per delta kesay find kaein?

$$\delta_j = o_j (1 - o_j) \sum_{k \in \text{output}} w_{kj} . \delta_k$$

$$E_d(\vec{w}) = \frac{1}{2} \sum_k (t_k - o_k)^2$$

d^{th} example of pattern

$$E_d(\vec{w}) = \frac{1}{2} \sum_d \sum_k (t_k - o_k)^2$$

for batch mode : $w_{ji} + \sum_d \Delta w_{ji}$

why does stochastic converge faster:

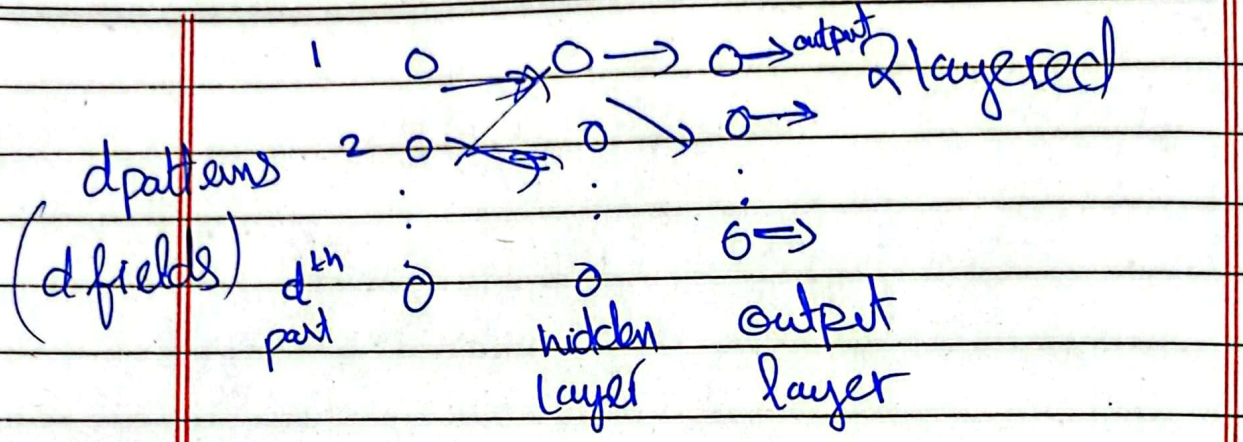
local minima

aik example ki aik weight local minima mein phas jaye but doosray mein na phasay

(stochastic kamyaab hai practically)

Date: __/__/20__

MON TUE WED THS FRI SAT



→ forward propogate the signals

1. For each o/p node:

delta k

error term $\delta_k = (t_k - o_k) \cdot o_k(1-o_k)$

error

derivative term

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

$\sigma'(x) = \sigma(x)(1-\sigma(x))$

$\Delta w_{kj} = \eta \cdot \delta_k \cdot o_j$ or x_{kj}

↓

kth node wein jth connection

sif 2 layers, Sigmoidal, fully connected,

keh liye derived

create non-convex cost function
- inefficiency

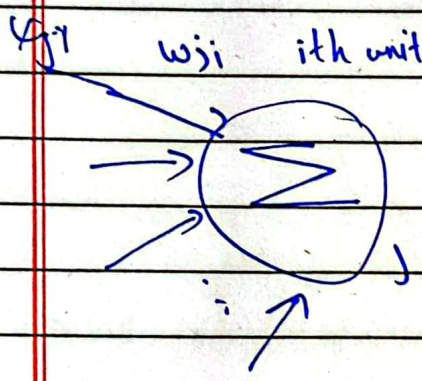
Date:__/__/20__ MON TUE WED THS FRI SAT ○○○○○○○

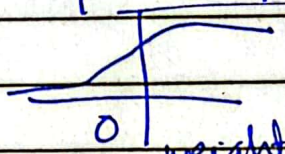
EBPA (n, n_{in}, n_{op}, n_{hd}, D)

→ create a n/w with n_{in}, n_{op}, n_{hd}

→ initialy the n/w with small weights [-0.05, +0.05]

↳ keep weight small



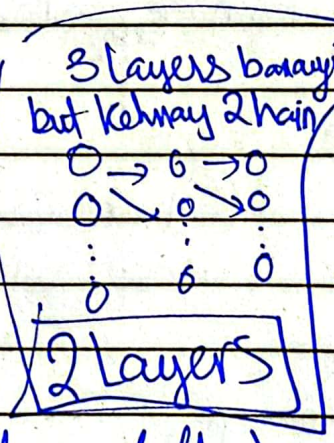


or net_j

$$net_j = \sum_i w_{ji} x_{ji}$$

↑ i will vary

3 Layers banayi but kehnay 2 hain why small seh initialize?



↓ yahan per error change nhi aata toh iss liye 0 seh initialize kia bhai gradient change bhi aaye.

Sigmoid function net per chalta ↓ Sexy always

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

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

nrse based fully connected sigmoidal

Date:__/__/20__

MON TUE WED THS FRI SAT

$$P(y=c|z) = \frac{e^{z_c}}{\sum_{j=1}^{c} e^{z_j}}$$

output are probabilities

$$L(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{c=1}^{c} y_c^{(i)} \ln\left(P(y=c|x^{(i)})\right)$$

predicted prob

minimize the -ve log likelihood

$$\frac{\partial L(\theta)}{\partial \theta_c} = -\frac{1}{m} \sum_{i=1}^{m} \left[y_c^{(i)} - P(y=c|x^{(i)}) \right] x^{(i)}$$

$$\theta_1 = \theta_i - \alpha \frac{\partial L}{\partial \theta_c}$$

$$\frac{e^{z_c}}{\sum_{j=1}^{c} e^{z_j}}$$

MON TUE WED THS FRI SAT

Date:__/__/20__

x	y

$\mathbb{R} \to \mathbb{R}$

$x_1 \cdots x_d$	y

$D : \mathbb{R}^{d+1} \to \mathbb{R}$

$n >> d$

↳ well defined

$y = X\theta$

$$\underset{n \times (d+1)}{X} \quad \underset{(d+1) \times 1}{\theta} = \underset{n \times 1}{y}$$

$$\theta = y \cdot X^{-1}$$

$$= y \cdot (X^T X)^{-1}$$

$$= \cancel{XO}$$

$$X^T (y - \hat{y}) = 0$$

$$X^T y - X^T \hat{y} = 0$$

$$X^T y = X^T \hat{y}$$

$$\hat{y} = X\theta$$

$$X^T y = X^T X \theta$$

$$(X^T X)^{-1} \cdot X^T y = \theta$$

$$\theta_j = \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j} \quad \left(\frac{\partial J(\theta)}{\partial \theta_j} = \theta_j + \alpha \left(y - h_\theta(x)\right) x_j\right)$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \left[\frac{\frac{1}{2} \sum_{i=1}^{n} \left((h_\theta(x)^{(i)} - y^{(i)} \right)^2}{\partial \theta_j} \right]$$

$$= \frac{2}{2} \times \frac{1}{2} \left(h_\theta(x) - y\right) - \frac{2}{2\theta} \left(h_\theta(x) - y\right)$$

$$= \left(h_\theta(x^i) - y^i\right) \cdot x_j$$

d → no. of input vars excluding x_0)

cost func

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)})^2$$

chose θ to minimize $J(\theta)$

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

$$= \theta_j + \alpha (h_\theta(x) - y) x_j$$

y-h
+
h-y
⊖

one training ex:

$$\theta_j = \theta_j + \alpha (y^{(i)} - h_\theta(x^{(i)})) x_j^{(i)}$$

$$\theta_j = \theta_j + \alpha \sum_{i=1}^{n} (y^{(i)} - h_\theta(x^{(i)})) x_j^{(i)}$$ for every j

$\mathbb{R}^{n \times d} \to \mathbb{R}$

$X \to n \times (d+1)$

$$\nabla_\theta J(\theta) = X^T X\theta - X^T \vec{y}$$

$$0 = X^T X\theta - X^T \vec{y}$$

$$\theta = (X^T X)^{-1} X^T \vec{y}$$

$$\theta = (X^T X)^{-1} \cdot X^T \vec{y}$$

$\frac{X^T \vec{y} = 0}{X^T X}$

learning rate not in normal eq

how many times to update parameter? n/k

sequential gd

Date:___/___/20___

MON TUE WED THS FRI SAT

$$(h_\theta(x^{(i)}))^{y^{(i)}} \times (1-h_\theta(x^{(i)}))^{1-y^{(i)}}$$

$$L(\theta) = \log L(\theta) = \sum_{i=1}^{n} y^{(i)} \log(h_\theta(x^{(i)})) + (1-y) \log(1-h_\theta(x^{(i)}))$$

$$\sum_{i=1}^{n} y^{(i)} \log(h_\theta(x^{(i)})) + (1-y) \log((1-h_\theta(x^{i}))$$

$$\frac{\partial}{\partial \theta_j} L(\theta) = y - \frac{1}{g(\theta^T x)} - (1-y)\left(\frac{1}{1-g(\theta^T x)}\right) \frac{\partial}{\partial \theta_j} g(\theta^T x)$$

$$= y\,(1 - g(\theta^T x)) - (1-y)(g(\theta^T x)) \frac{\partial}{\partial \theta_j} \theta^T x$$

$$= (y - h_\theta(x)) x_j$$

$$\theta_j := \theta_j + \alpha \frac{\partial L(\theta)}{\partial \theta_j}$$

$$\frac{\partial L(\theta)}{\partial \theta_0} = \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)}) \right)$$

$$\frac{\partial L(\theta)}{\partial \theta_1} = \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)}) \right) \cdot x^{(i)}$$

(Logistic Sigmoid function)

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}} = \frac{1}{1+e^{-(\theta_0 + \theta_1 x)}}$$

↳ why?

~~to classify~~ predicted y

$\frac{1}{1+e^{-z}}$

$h_\theta(x) = \{0, 1\}$

is b/w 0, 1 & real values

$z \to \infty : h(x) \to 1$

$z \to -\infty : h(x) \to 0$

$\theta^T x = \theta_0 + \sum_{j=1}^{n} \theta_j x_j$

or $\theta_0 + \theta_1 x + \theta_2 x_2 + \dots \theta_j x_j$

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

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

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

$$= g(z)\left(1 - g(z)\right)$$

$\theta_0 = 0.2$

$\theta_1 = 0.5$

$$h_\theta(x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x)}}$$

$$h_\theta(0.5) = \frac{1}{1 + e^{-(0.2 + 0.5(0.5))}}$$

$$\Rightarrow \frac{1}{1 + e^{-0.45}}$$

$$\frac{1}{0.610639} = 0.5866 \simeq 1$$

$$h_\theta(1.0) = \frac{1}{1 + e^{-(0.2 + 0.5(1))}}$$

$$= \frac{1}{1 + e^{-0.7}}$$

$$\frac{1}{0.668187} = 0.6225 \simeq 1$$

$$h_\theta(1.5) = \frac{1}{1 + e^{-(0.2 + 0.5(1.5))}}$$

$$= \frac{1}{1 + e^{-0.95}}$$

$$= 0.6791 \simeq 1$$

$$\frac{1}{0.7211} = \ln(\quad)$$

a) $L(\theta) = y \cdot \log(h_\theta(x)) + (1-y) \cdot \log(1 - h_\theta(x))$

$= 1 \cdot \log(0.610) + (1-1) \cdot \log(1 - 0.610)$

$= -0.7131 \qquad 0 \cdot \log(1 - 0.610)$

$= 0.7131 \qquad \ln(0.49) = -0.494$

b) $L(\theta) = y \cdot \log(h_\theta(x)) + (1-y) \cdot \log(1 - h_\theta(x))$

$= 0 \cdot \log(0.668) + (1-0) \cdot \log(1 - 0.668)$

$= 0 + \log(0.332)$

$= -1.590744$

c) $L(\theta) = y \cdot \log(h_\theta(x)) + (1-y) \cdot \log(1 - h_\theta(x))$ $\quad -1.295$

$= 1 \cdot \log(0.750) + (1-1) \log(1 - 0.750)$

$= -0.415037 + 0 \qquad x = -0.415$

Date: __/__/20__

MON TUE WED THS FRI SAT SUN

→ Back propogate errors

- O/P layer

error Term $\delta_j = O_j(1-O_j)(t_j - O_j)$

1 - output is derivative of sigmoid

gradient of o/p function (derivative of sigmoid)

prediction error

O_j → hypothesis

whole structure is the hypothesis

$= O_j(1-O_j) - (t_j - O_j)$

$\Delta w_{ji} = \eta \delta_j x_j$

↳ for output layer

3 things: learning const term · delta · signal strength

Sequential learning
stochastic learning (SGD)

EBPA (η, n_{in}, n_{hidden}, $n_{o/p}$)

↓ input kaitnay perceptron

↓ hidden layer mein kitnay perceptron

Sirf 2 layers ke liye hai

→ create the network

(design choice : deciding how many nodes we need)

hidden layer most important layer

transformation iss layer ki zimmedari

→ initialize weights between small interval

[-0.05 to 0.05] input layer hidden layer

isko GD sah optimize kiye jaye ga

why?

→ forward propogate inputs

how are we going to forward propogate?

weighted + threshold sum

for all neurons in hidden layer then in output layer

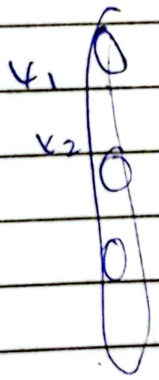
start off with first hidden layer

layer by layer propogation

use a non-linear continuous threshold
sigmoidal unit how atleast
tangent sigmoid
multinomial logistic regression
Softmax function

x. back propogation algorithm
→ direction of finding errors & adjusting weight
→ transfered errors ko back propogate & signals ko forward propogate

also called forward propogation



input layer

signals ki propogation keh liye use hoti
(forward propogation

0 ek scaler | 0 0 0 6 vector

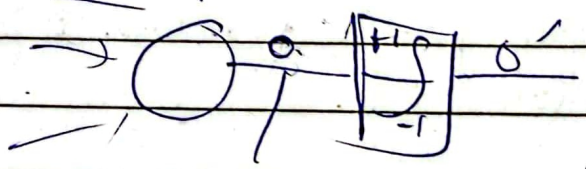
also called: Error back propogation algo

Date: __/__/20__

this will converge till t-0 doesn't matter

this will converge, but doesn't mean it will converge give correct classification, but will give you minimum error classification, but normal perceptron doesn't converge. converge

what is the difference b/w this & a perception?

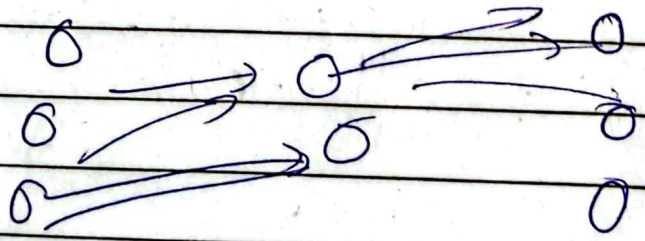


GD

eg: squared features for XOR in logistic regression

how can we make powerful structures from it

perception always outputs a bool.



input layer

hidden layer

galat naam as perception in MLP after this

Date: __/__/20__

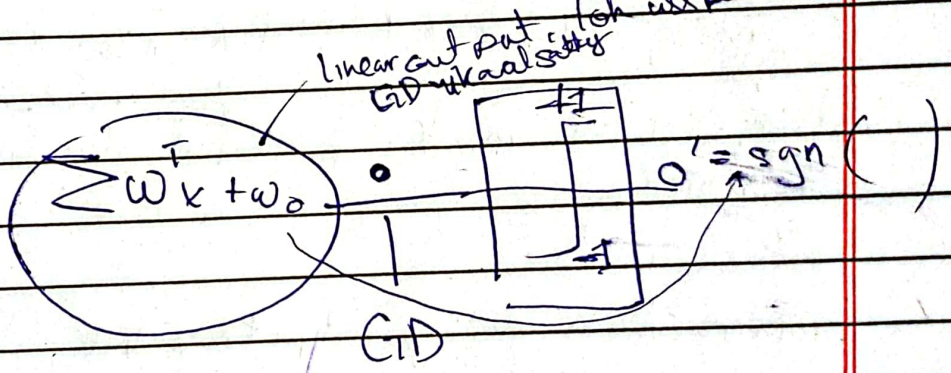
MON TUE WED THS FRI SAT

take a higher degree polynomial
square etc

$\sum w^T x + w_0$ ← degree of polynomial

agar sigmoid mein perceptron ko manipulate
kar bhi toh na sakta?
wow bhi line draw kar rahe ga
but in a higher dimension

can't apply GD to perceptron, derivative
bhi bekaar pada, can't take derivative
of step function



$$E(w) = \frac{1}{2} \sum_{i=1}^{n} (t^{(i)} - o^{(i)})^2$$

→ simplify $w^T x - w_0$

o → is a real value
keep it near +1 or -1

$\frac{\partial E}{\partial w_i} = ?$

philosophy: $\Delta w_j \propto -\frac{\partial E}{\partial w_i} = -\frac{1}{2} \sum_{i=1} (t^{(i)} - o^{(i)}) \cdot x_j$

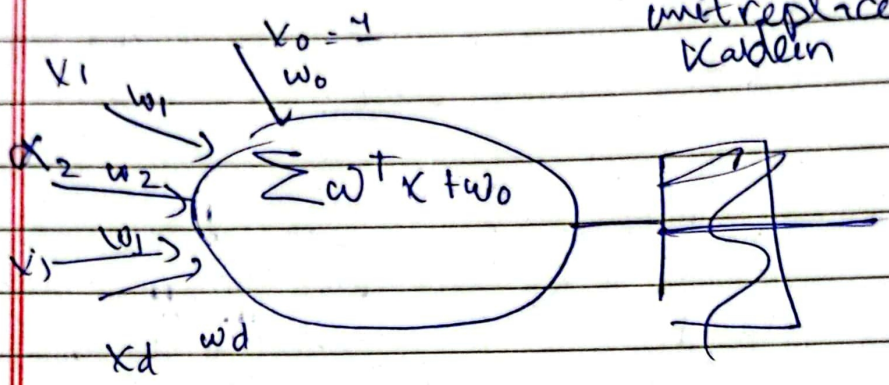
now, what is the difference b/w perceptron
& logistic regression? difference in error
kar rahe

MON TUE WED THS FRI SAT

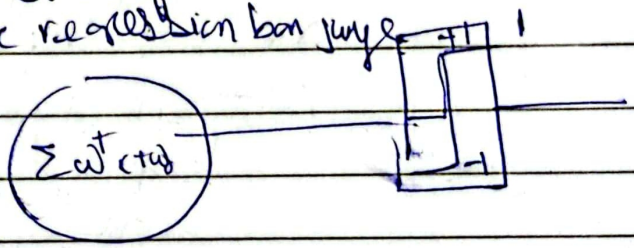
Date: __/__/20__

$\vec{w} \leftarrow \vec{w} + \Delta\vec{w}$ x_j → input vector

agar regression kahi toh perceptron kehsaath kal sakte ay, agar thresholding unit replace kardein



for classification:
yeh logistic regression ban jaye



hyperbolic tangent function

→ you can only draw a linear boundary

while logistic regression can draw a non linear boundary:

how can we draw a non linear boundary in logistic regression

perceptron needs multiple perceptrons to make non-linear boundary

Date: __/__/20__

2 layer kehsaath you can recognise all patterns. with perception

$\|x\|^2 + 1$ → adding this to weight jab galti hoti model seh

↓ input vector

$e = (t - o)$

↓ target ↳ actual output

ch4 → textbook

$\Delta w_j \propto e$ error zyada

$\Delta w_j \propto x_j$ input strength zyada

$\Delta w_j \propto (t-o) x_j$

$\Delta w_j = \eta (t-o) x_j$

↑ eta

way ①: update after every iteration → $\Delta \vec{w} = \eta (t-o) \cdot \vec{x}$

(instantaneous adjustment / online / offline) immediately apply it

calculated got

e $+1 - (-1) = 2$

$-1 - (+1)$

$e = -2$

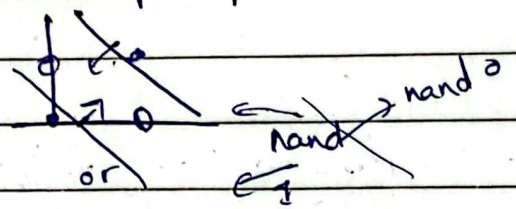
$\eta = (0, 1]$

↑ learning constant

way ②: batch adjustments — accumulate karay dan, ek hat aik ka vector banaeyein & aggregate karadain $\Delta w^{\rightarrow(i)} + \Delta w^{\rightarrow(i,1)} + \dots \Delta w^{\rightarrow(n)}$

Date: __/__/20__

need 3 perceptron to create make XOR.

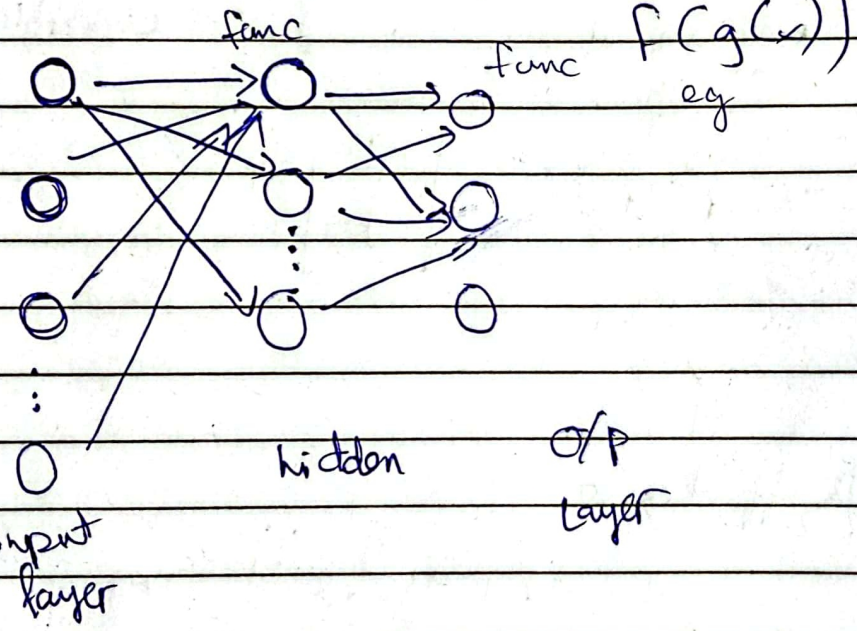


in dono ke firing ko kon decide karay ga? another perceptron (and)

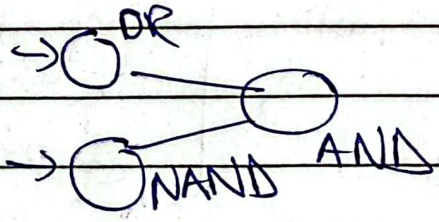
Multi Layered Perceptron

→ we'll study one only

→ **feed forward**



Deep network: multiple hidden layers



We need a multilayered perceptron with learning algorithm.

Perceptron (Dn, T)

$w = [0\ 0 \dots 0]^T$; $w_0 = 0$

for $t = 1$ to T # of epochs (# of generations in GA)

  for $i = 1$ to n

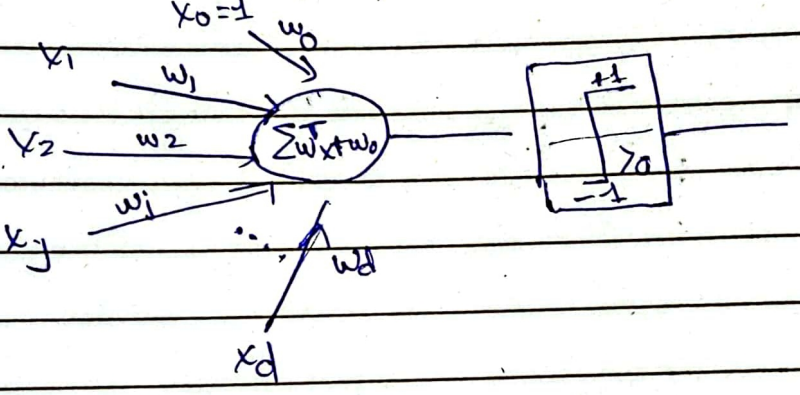
    if $y^{(i)}(w^T x + w_0) \leq 0$

      $w \leftarrow w + y^{(i)} \cdot x^{(i)}$ (vector updation)

      $w_0 \leftarrow w_0 + y^{(i)}$ (scaler updation)

return w, w_0

linear classification



starts off at zero

$w \leftarrow w + y^{(i)} \cdot x^{(i)}$ → vector

$w_j \leftarrow w + x_j^{(i)}$] → scalar

w_1	w_2	..	w_d	w_0
x_1	x_2		x_d	x_0

$$\begin{bmatrix} = \\ \vdots \\ = \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + 1 \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

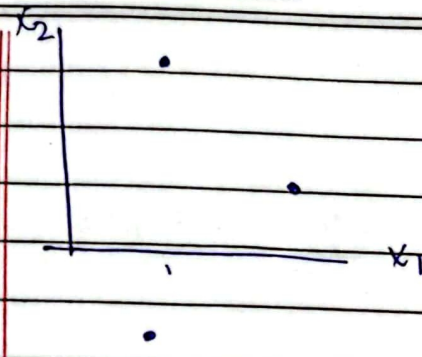
epochs

If a pattern cannot fit the data, toh jab tak saari iteration expire na ho jaye toh dega, & in the end will give misconfiguration.

Date:__/__/20__

always include it is 1

MON TUE WED THU FRI SAT



x_0	x_1	x_2	y
1	1	3	1
1	3	1	1
1	-1	1	-1
1	1	-3	1

$w = [1 \quad -1 \quad 1]$

1) $\text{sgn}(w^T x)$

$$[1 \quad -1 \quad 1]\begin{bmatrix}1\\3\\1\end{bmatrix} = 1 - 3 + 1 = -1$$

$\text{Sgn}(w^T x) = -1$

signum function

write this way

x_1	x_2	x_0

(2) $1 \cdot -1 \leq 0$

(3) w.

$$\begin{bmatrix}1\\-1\\1\end{bmatrix} + 1 \cdot \begin{bmatrix}1\\3\\1\end{bmatrix} = \begin{bmatrix}2\\2\\2\end{bmatrix}$$

new weight vector is $[2 \quad 2 \quad 2]$

iter 2

$$[2 \quad 2 \quad 2]\begin{bmatrix}3\\1\\1\end{bmatrix} \qquad [2 \quad 2 \quad 2]\begin{bmatrix}-1\\1\\1\end{bmatrix} = \begin{bmatrix}2\\2\\2\end{bmatrix}$$

completely run it
check where it stops
if it does

$= 2$

$-1(2) \leq 0$

-2

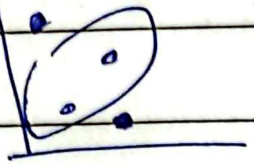
0 → w now

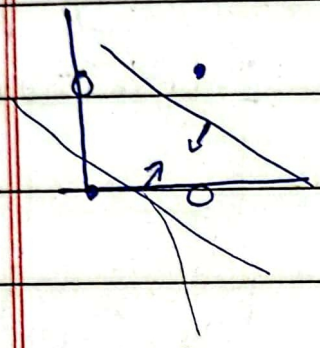
in reference book

Mon Tue Wed Thu Fri Sat

Date: __/__/20__

you can use logistic regression to solve this XOR problem, as we can create non-linear boundaries.





can solve it easily with 2 perceptions

firing decision jab done 1,1 toh fire and this class → 0

→ multiple perceptions seh hum koi bhi boundary bana saktay, like circular can make any pattern, specially 2 layer walay perceptron:

for XOR

need 2 layer, ak hidden.

practical difficulty.

↳ phir emerge hua deep neural networks

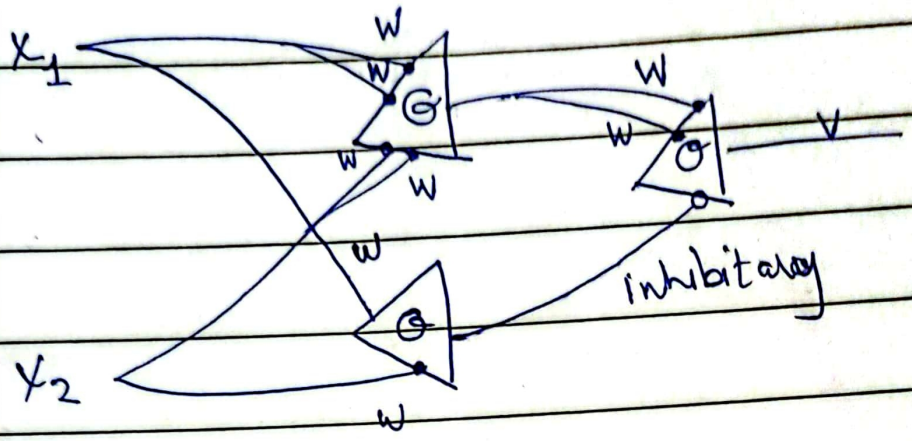
x_1	x_2	y
1	3	1
3	1	1
-1	1	-1
1	-3	1

$$W = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} \begin{matrix} w_1 \\ w_2 \\ w_0 \end{matrix}$$

x_2

Mon Tue Wed Thu Fri Sat

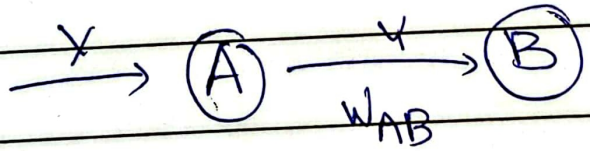
Date: __/__/20



all weights 'W' = 1 & $\theta = 2$

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

* special condition, inhibitory lock kardega.



hebbian theory?

$W_{AB} \leftarrow W_{AB} + \alpha X Y$

problem of this model:

* learning hai, but learning is unbounded

* Neuron Networks ki problem: learning nhi hoti.

Perceptron - Neural Model
- animal eye demonstration

Mon Tue Wed Thu Fri Sat

Date: 12/12/2024

Mc Culloch Pitt Neuron

Solution for XOR

M-P characteristics:

(i) They are binary devices, i.e.

O/P $\in \{1, 0\}$

(ii) Each neuron has a fixed threshold 'θ'

(iii) The neuron receives inputs from excitatory synapses, all with 'identical' wts.

There may be more than one weight from a single source.

(iv) Inhibitory inputs have an absolute veto power over any excitatory input

(v) At each time step, the neurons are synchronously updated by summing the weighted excitatory inputs & outputting as follows:

$$V = \begin{cases} 1 & \sum W x_j \geq \theta \text{ and No INHIBITATION} \\ 0 & \text{otherwise} \end{cases}$$

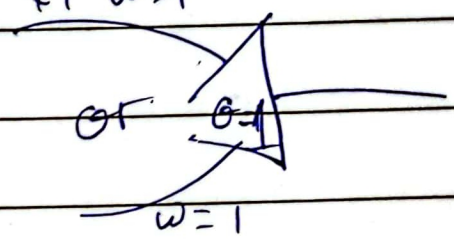
Mon Tue Wed Thu Fri Sat

Date: __/__/20__

M-P $y = \begin{cases} 1 & \sum_j w x_j \geq \theta \text{ and there is no inhibition} \end{cases}$

It is equivalent to boolean gate (NOT)

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

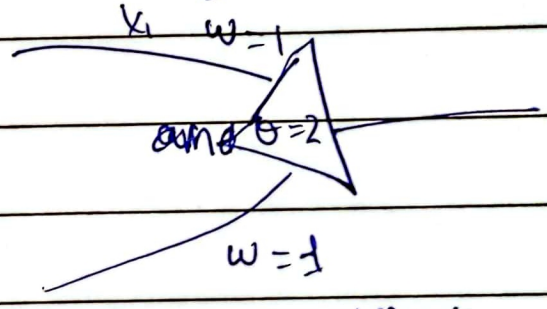


multinomial logistic regression

XOR ka bana ka dikhayen ~~next time~~

~~selection~~

convert to and



Quiz: Linear regression + Logistic regression at Thursday

softmax ka doosra derivative dekhna hai.

Loss function ka derivative dekhna hai.

Mon Tue Wed Thu Fri Sat

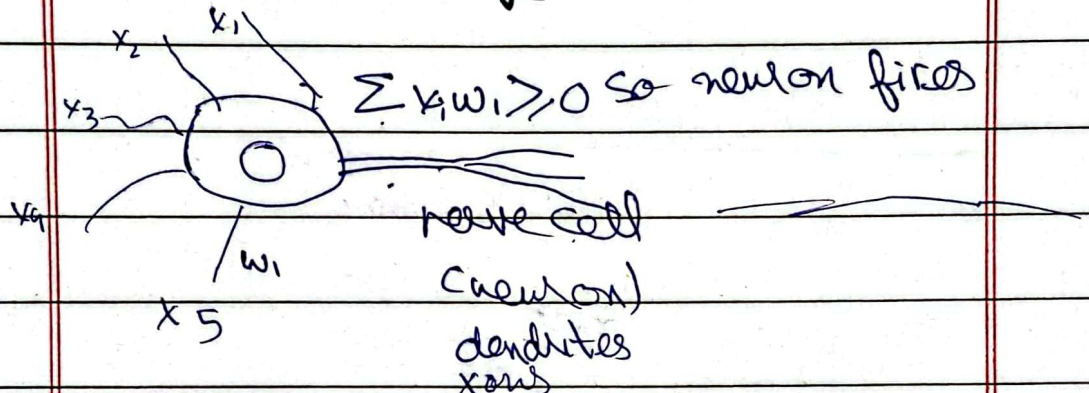
Date: __/__/20__

$\frac{\partial L(\theta)}{\partial \theta_i} = \theta_i - 1$ $\quad \vec{\theta} - \vec{1}$ $\quad \vec{1} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

1 - C / 1 - K

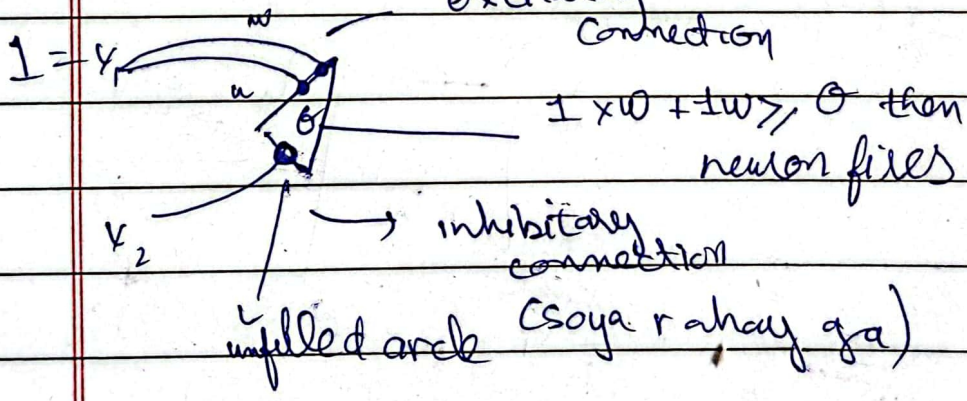
1 - Hot coding

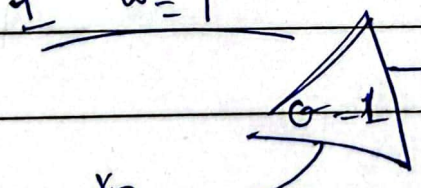
Neural Computing



Linear combination = signal x weight

Mechalic Pitts





$V \in \{1, 0\}$

x_2	V
0	1
1	0

$1 \times 1 \geq 1$

if x_2 is 1 now matlb what the excitation is it will always produce 0.

difference b/w stochastic & incremental

↓ randomly ~~activation~~, you do the next val

↓ first row based, 2nd dachtay phr 3rd

to extend to K classes:

$y \in \{1, 2, \dots k\}$

$\vec{\theta}_1, \vec{\theta}_2, \dots, \vec{\theta}_K$ } vectors

$\theta_i \in \mathbb{R}^{d+1}$

→ SOFTMAX $(\theta_1^T x, \theta_2^T x, \dots, \theta_K^T x)$ — K scalers

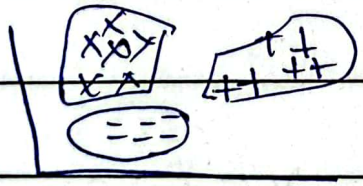
$f: \mathbb{R}^K \to \mathbb{R}^K$

θ_1
θ_2
$\vdots$
θ_K } K entries

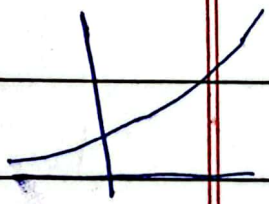
$\frac{e^{\theta_i^T x}}{\sum_j e^{\theta_j^T x}}$

$\sum_i \phi_i = 1$

exponential function



3 classes jiski highest probability, yeh wo wali class



Mon Tue Wed Thu Fri Sat Date: __/__/20__

$$= \frac{1}{1+e^{-x}} - \left(\frac{1}{1+e^{-x}}\right)^2$$

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

now back to binary cross-entropy:

$$= -\left(\frac{y}{h_\theta(x)} - \frac{(1-y)}{1-h_\theta(x)}\right) h_\theta(x)(1-h_\theta(x))$$

$$= -\left(\frac{y}{h_\theta(x)} - \frac{(1-y)}{1-h_\theta(x)}\right) h_\theta(x)(1-h_\theta(x)) \cdot x_j$$ x is a vector

$$J'(\theta) = -(y(1-h_\theta(x)) - (1-y)h_\theta(x))\, x_j$$

$$= -(y - yh_\theta(x) - h_\theta(x) + yh_\theta(x))\, x_j$$

$$J'(\theta) = -(y - h_\theta(x))\, x_j$$

$$J'(\theta) = (h_\theta(x) - y)\, x_j$$

$\frac{\partial \theta^T x}{\partial \theta_j}$ & $\frac{\partial}{\partial \theta_j}(\theta_d x_d + \theta_{d-1} x_{d-1} + \ldots + \theta_1 x_1 + \theta_0) = x_j$ (saaray 0s no jitay except j wala term)

derivative term is exactly same as linear reg, but this has different cost function

GD: $\theta_j := \theta_j - \alpha J'(\theta)$

$$\theta_j = \theta_j - \alpha(h_\theta(x) - y)\, x_j$$

$$\theta_j = \theta_j + \alpha(y - h_\theta(x))\, x_j$$

$$\theta_j = \theta_j + \alpha\left(y^{(i)} - h_\theta(x)^{(i)}\right) x_j^{(i)}$$

batch mode mein Σ summation:

$$\theta_j = \theta_j + \alpha \sum_{i=1}^{n} \left(y^{(i)} - h_\theta(x)^{(i)}\right) \cdot x^{(i)}$$

$\therefore \frac{d}{dx} \log x = \frac{1}{x} \frac{d(x)}{dx}$

Mon Tue Wed Thu Fri Sat

Date: __/__/20

$$= -\left(\frac{\partial (y \log(h_\theta(x))}{\partial \theta_j} + \frac{\partial (1-y)(\log(1-h_\theta(x))}{\partial \theta_j} \right)$$

$$= -$$

$$\frac{\partial \log(h_\theta(x))}{\partial \theta_j} = \frac{1}{h(\theta)^x} \cdot \frac{\partial (h_\theta(x))}{\partial \theta_j}$$

$$= -\left(\frac{\partial (y \log(h_\theta(x))}{\partial \theta_j} + \frac{\partial (1-y)(\log(1-h_\theta(x))}{\partial \theta_j} \right) \left\{ \begin{array}{l} (1-y) \cdot \frac{1}{1-h_\theta(x)} \\ \cdot \frac{\partial}{\partial \theta_j} (1-h_\theta(x)) \end{array} \right.$$

$$= -\left(y \cdot \frac{1}{h_\theta(x)} - (1-y) \frac{1}{1-h_\theta(x)} \right) \frac{\partial h_\theta(x)}{\partial \theta_j}$$

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

$$\frac{d}{dx} \sigma(x) = \frac{d}{dx} \left(\frac{1}{1+e^{-x}} \right)$$

$$\frac{d}{dx}(1+e^{-x}) = \frac{d}{dx}(1) + \frac{d}{dx}(e^{-x}) = e^{-x}$$

$$= \frac{(1+e^{-x}) \cdot \frac{d(1)}{dx} - (1) \frac{d}{dx}(1+e^{-x})}{(1+e^{-x})^2}$$

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

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

Mon Tue Wed Thu Fri Sat

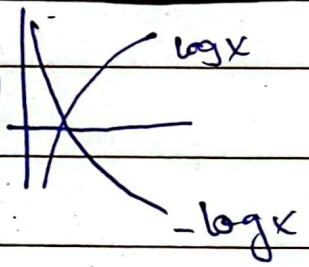
Date: __/__/20

cost function: kitni misclassification (not good)

- correct category seh kitna door, hmmm continuous label banana hai.

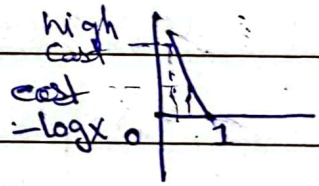
Binary cross entropy

$$J(\theta) = -y(\log h_\theta(x)) - (1-y)\log(1-h_\theta(x))$$



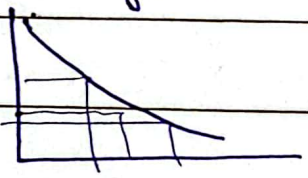
L is composite function, done ka combine kal raha

ab gradient descent lagasaktay

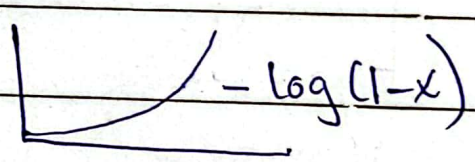


log x graph ka itna portion le lein

near 0, cost ziyada, aaniteh I chahiye thi



y=0 keh liye use this



$$\theta_j' = \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

$$J(\theta) = -y(\log h_\theta(x)) - (1-y)\log(1-h_\theta(x))$$

$$J'(\theta) = \frac{\partial (J(\theta))}{\partial \theta_j}$$

Mon Tue Wed Thu Fri Sat

Date: __/__/20

$h_\theta(x) = \theta^T x$

$y, \vec{x}, \vec{\theta}$

$J(\theta) = \frac{1}{2}\sum_{i=1}^{m} (h_\theta(\vec{x}_i;\theta) - y)^2$ ask Sample each type

$\theta = \theta + \alpha(y - h(\vec{x};\theta))\vec{x}$

→ we can use squared errors for classification, but ask no.

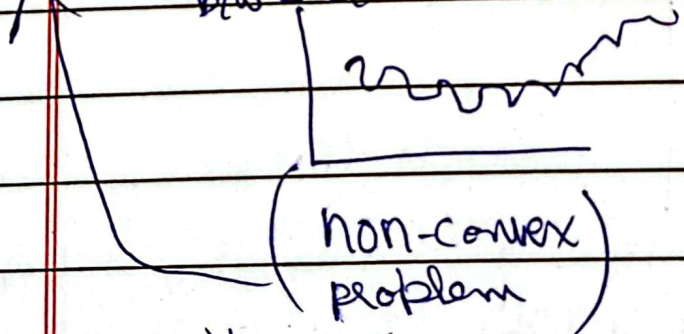
$h_\theta(x) = \theta^T x$ (regression)

$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$ → for logistic regression (classification)

$J(\theta) = \frac{1}{2}\sum_{i=1}^{m} (h_\theta(x) - y)^2$

→ non-linear

$h_\theta(x) \in [0, 1]$ & it is continuous



(non-convex problem)

can't use it

wiggly little functions, can't lower it with gradient descent

→ it is not guaranteed to converge, or converge into the best possible way

Mon Tue Wed Thu Fri Sat

Date: __/__/20

$\theta^T x \geq 0$

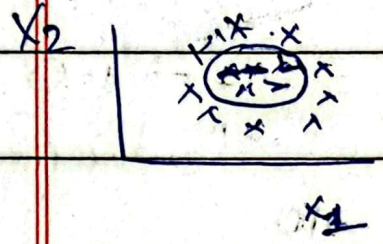
decision boundary

if $g(\theta^T x) \geq 0.5 \rightarrow y = 1$

else $g(\theta^T x) < 0.5 \rightarrow y = 0$

$\downarrow$

$\theta^T x < 0$



$\theta^T x = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_2^2 + \theta_4 x_2^2$

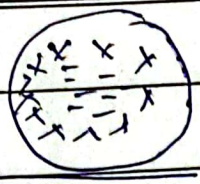
→ We can get confused in logistic regression with Linear Regression.

→ can draw all sorts of decision boundaries

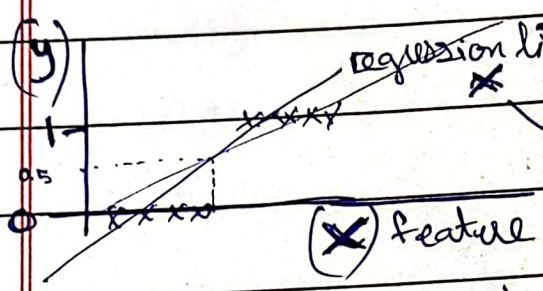
sum of sq. errors won't work here.

Date: __/__/20

Mon Tue Wed Thu Fri Sat



classification ki tareef aani chahiye.



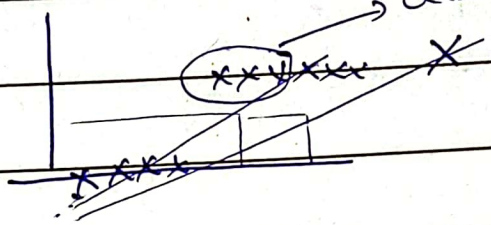
$h_\theta(x) \geq 0.5 \rightarrow y=1$

$h_\theta(x) < 0.5 \rightarrow y=0$

is it an outlier?

no

luxury house or not



classifies these as economy although it is luxury

hypothesis galat hogaya

we can draw a line but not a trend line but a seperator.

$\theta^T x > 0$

; wont be a trend line but a seperator

$\theta^T x < 0$

line pe agar koi value aaye toh uski value kya hogi?

$\theta^T x = 0$

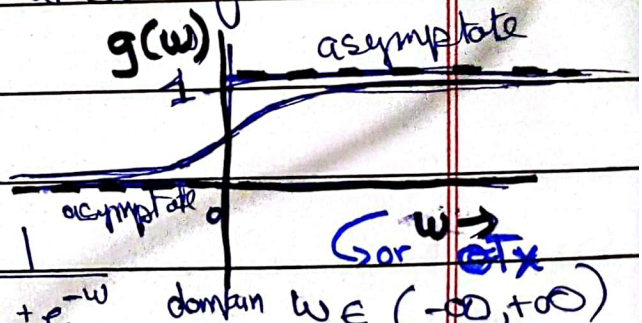
Logistic Regression

→ naam regression but kaam classification

$$h_\theta(x) = g(\theta^T x)$$

function

$$g(w) = \frac{1}{1+e^{-w}}$$



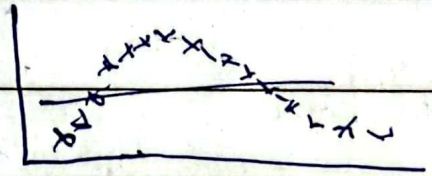
domain $w \in (-\infty, +\infty)$

range $0 \leq g(w) \leq 1$

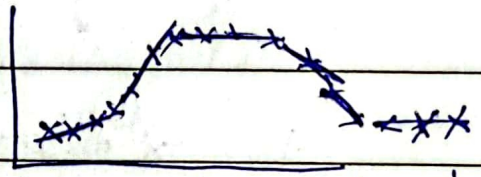
→ can take it as probability

Agar trend kuch aur ho jaye toh learning rate acha nhi

→ trend dekh kar, we can set the best learning rate, otherwise change it.



Linear regression ka algo karlega, but we know trend isn't fit.



Logally administered linear regression? google the term

Lazy learning

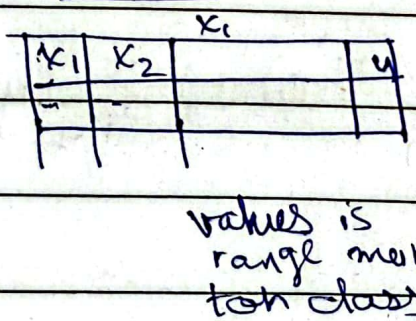
Linear Regression vs Classification?

feature space

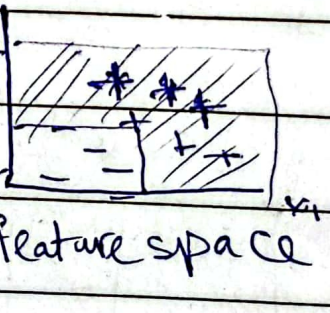
what is essence of classification?

piecewise approximation

classification mein discrete values
regression mein continuous values



values is range mein ton class



feature space

decision tree is narrowing your search

- You can use decision trees for regression; Regression trees.

decision trees classification.

Mon Tue Wed Thu Fri Sat

Date: __/__/20

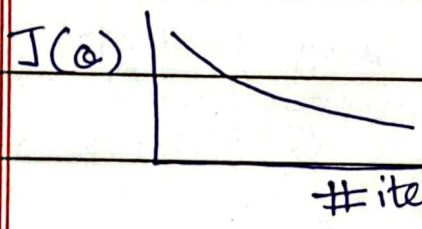
$\alpha \in (0,1]$

→ bahut chota alpha
bahut ~~bada~~ chota convergence

→ bahut bada alpha, bai convergence ~~kar~~

hyperparameter → α

why is it a hyperparameter? ↳ you set it.
why not parameter? ↳ learned from your algorithem.



↳ x aesay nhi jaana

$J(\theta)$ = sum of squared residuals

$h_\theta(\theta^T x)$ → hypothesis (predicted)
y → known value

$(y - \theta^T x)^2$

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{n} (y - \theta^T x)^2$$

kitnay feature vectors? n

x_1	x_2	...	x_d	x_0
—	—	—	—	1
—	—	—	—	1

Mon Tue Wed Thu Fri Sat Date: __/__/20__

Artificial Intelligence

types? or

2 Linear regression solutions:

- normal eq
- gradient descent

types of gd:
- sequential
- stochastic
- mini batches

$\frac{nk}{5}$? $\left[\frac{n}{k}\right]$ → tells this times we have to update parameter number

Scaling only matters in GD based solution, not normal equations

Learning rate kitna hona chahiye:

→ used in GD, not in normal eq

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

-ve sign to go downwards, opposite to gradient

$\frac{\partial J(\theta)}{\partial \theta_j}$ rate of change

α = Step size

convergence in GD? what does it mean? it is a problem of GD

Solution to it:

$\left(\underset{\text{range}}{1 \to 0}\right)$ $x_{new} = \frac{x_{old}}{max(x)}$

or

$\left(\underset{\text{range}}{-1 \to +1}\right)$ $x_{new} = \frac{x_{old} - x_{mean}}{max(x)}$

min max normalization: $x_{new} = \frac{x_{old} - x_{mean}}{max(x) - x_{mean}}$

$x_{zs} = \frac{x_{old} - \mu}{\sigma}$

z-score

scale our features!

why do we normalise?
→ to make convergence faster

Mon Tue Wed Thu Fri Sat

Date: __/__/20__

$\nabla_\theta J(\theta) = 0$

$(X\theta - y)(X\theta - y)$

gradient symbol (nabla)

$$\theta_j = \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

$$\theta = \theta + \alpha \sum_{i=1}^{n} \theta^T x^{(i)} - y^{(i)} \cdot x^{(i)}$$

(vector)

(scaler)

$\Delta\theta$

$$\theta = \theta + \alpha \sum_{i=1}^{n} (y^{(i)} - \theta^T (x^i)\, x^{(i)}$$

batch gradient descent

purana theta

cumulative adjustment

Sequential GD (online, instantious, —)

while

for i=1 to n

$\theta = \theta \times \alpha (y^{(i)} - \theta^T x^{(i)}) \cdot x^{(i)}$

end

end

flavours of GD —

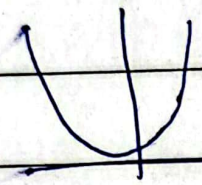
Stochastic GD

Mon Tue Wed Thu Fri Sat

Date: __/__/20__

agar maximizing ki problem thi toh max lagatay

$$\theta^* = \arg\min_{\theta} J(\theta)$$



R[a, b]
↳ find min max over this interval

$$\theta^* = (X^T X)^{-1} X^T y$$

$(d+1 \times n)(n \times d+1)$ | $n \times 1$ dimensions

$d+1 \times d+1$ $\quad (d+1) \times n$

after inv still: $d+1 \times d+1$ $\quad (d+1) \times 1$

we got this formula (least squared) from 1A ch6

$d+1 \times 1$

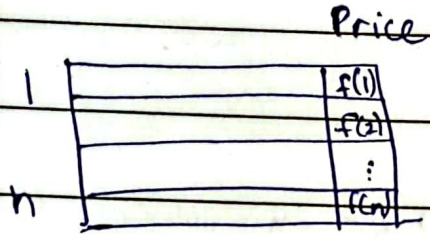
X matrix → design matrix

→ Problem : agar d+1 x d+1 big matrix so inverting is hard.

Mon Tue Wed Thu Fri Sat

Date: __/__/20__

polynomial regression → general term
linear regression → specific term
zaroori nhi keh sirf line guzaarein



Which hypothesis is best/good for linear regression?

→ jisman cost function ki value kam aaye
θ value... jo minimum hon/optimal

→ sum of sq residuals criteria batayen gaey

$J(\theta) = \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

$J: \mathbb{R}^{n+1} \to \mathbb{R}$

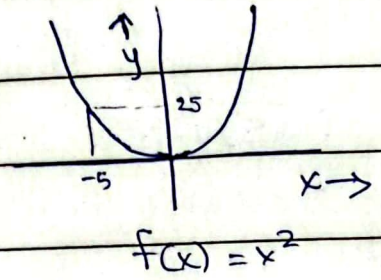
$h_\theta(x^{(i)})$ → vector
$y^{(i)}$ → scaler
$J(\theta)$ → vector
$\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$ → scaler

dot product is scaler

$x^{(i)} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \end{bmatrix}$

Mon Tue Wed Thu Fri Sat

Date: __/__/20__



$f(x) = x^2$

$x_0 = -5$

$\frac{d}{dx}(x^2) = 2x$

	x	f(x)
$x^{(0)} \to 1$	$x^{(1)} = -5$	25
$x^{(1)} \to 2$	$x^{(2)} = -4$	16

GD:

$x_{new} = x_{old} - \alpha(f'(x_{old}))$

$= -5 - 0.1(2(-5))$

$= -5 - \frac{1}{10}(-10)$

$= -5 + 1$

$= -4$

gradient descend:
opposite to the direction of gradient

gradient ascend:

$x_{new} = x_{old} + \alpha(f'(x_{old}))$

d+1 dimensional
↳ input

$\theta^T x + \theta_0 = \sum_{j=0}^{d} \theta_j x_j = \theta^T x$

inner product

$[\theta_1 \ \theta_2 \ \theta_3 \ \dots \ \theta_d \ \theta_0] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$

Mon Tue Wed Thu Fri Sat

Date: __/__/20__

$x_3 = -1.25$:

$x_4 = -1.25 - 0.25(-1.25) = 0.625$

$x_4 = 0.625$:

$x_5 = 0.625 - 0.25(3.75) = -0.3125$

x	f(x)
-5	75
2.5	18.75
-1.25	4.6875
0.625	1.171875
-0.3125	0.29296875
0.15625	0.073242

$x_5 = 0.625 - 0.3125$

$x_6 = -0.3125 - 0.25(-0.3125)$

$= 0.15625$

$6x$

Mon Tue Wed Thu Fri Sat

Date: __/__/20

$$\theta_j = \theta_j + \alpha \sum_{i=1}^{n} \left(y^{(i)} - h_\theta(x)^{(i)}\right) \cdot x_j^{(i)}$$

for complete vector

(all samples)

$$\theta_j = \theta_j + \left[\begin{matrix} \\ \\ \\ \end{matrix} \right] \cdot x^{(i)}$$

batch gradient descent

Q. $f(x) = 3x^2$ minimize using gradient descent iteratively

	x	f(x)
$x^{(1)}$	-5	75

GD ; $\alpha = 0.25$

$x = x - \alpha f'(x)$

$$f'(x) = \frac{d}{dx}(3x^2) = 6x$$

$x^{(5)}$

$x_1 = -5$: $x_{n+1} = x_n - \alpha f'(x_n)$

$= -5 - \alpha\, 6(-5)$

$= -5 - 0.25(-30)$

$= -5 + 7.5$

$= 2.5$

$x = 2.5$:

$x_3 = 2.5 - 0.25(6(2.5))$

$= 2.5 - 3.75 = -1.25$

Mon Tue Wed Thu Fri Sat

Date: __/__/20

$$\theta_j = \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

α → alpha (learning rate/const) (stepsize)

$[0, 1]$

$\alpha > 0$

0.1, 0.2

20% consider kara chah rahay

0.8 → 80% consider kara chah rahay

$$\frac{\partial J(\theta)}{\partial \theta_j} = ?$$

$$= \frac{\partial \left[\frac{1}{2} \sum_{i=1}^{n} (h_\theta(x)^{(i)} - y^{(i)})^2 \right]}{\partial \theta_j}$$

consider we have single example

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{2\left(\frac{1}{2}(h_\theta(x) - y)^2\right)}{\partial \theta_j}$$

$$= \frac{1}{2} \times 2 (h_\theta(x) - y) \cdot \frac{2(h_\theta(x) - y)}{\partial \theta_j}$$

$$= (h_\theta(x) - y) \cdot x_j$$

$$\frac{\partial (h_\theta(x))}{\partial \theta_j} = \frac{\partial \theta^T x}{\partial \theta_j}$$

$$= \frac{\partial (\theta_d x_d + \theta_{d-1} x_{d-1} + \theta_j x_j + \dots + \theta_1 x_1 + \theta_0)}{\partial \theta_j}$$

$$\frac{\partial (h_\theta(x))}{\partial \theta_j} = x_j$$

$$\theta_j = \theta_j - \alpha (h_\theta(x) - y) \cdot x_j$$

jth component $\theta_j = \theta_j + \alpha (y - h_\theta(x)) \cdot x_j$

hamnay salay vector mein update find karna

Mon Tue Wed Thu Fri Sat

Date: __/__/20__

$X^T \hat{y} = X^T y$

$\hat{y} = X\theta$

$\boxed{X^T X\theta = X^T y}$

LA Ch6 ↓

normal equation

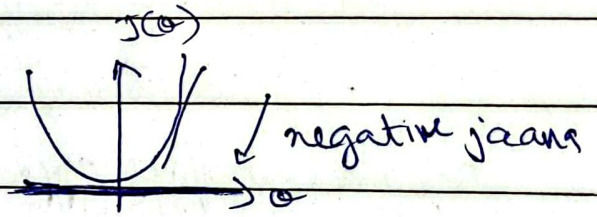
(sys of linear eq)

$\frac{\partial J(\theta)}{\partial \theta}$

$\theta = (X^T X)^{-1} X^T y$

Revisit LA chap 6

Assume a vector of 'θ' having specific values and then iteratively improve it to minimize the cost function for $J(\theta)$



$J(\theta)$ is a quadratic func

$\theta_j^{new} = \theta_j^{old} \ominus \alpha \frac{\partial J(\theta)}{\partial \theta_j}$

↓ opposite direction to gradient

gradient descent

move in op direction of gradient...

matrix X

$X_{n \times (d+1)}$ $\theta_{(d+1)\times 1} = y_{n\times 1}$

rows cols

matrix . vector =

$X\theta = y$ $\theta = ?$ can't find θ in $\theta = X^{-1}y$ ✗

$Ax = b$ (LA) as $n \gg d$

$x = A^{-1}b$

X sq matrix hi nhi hai

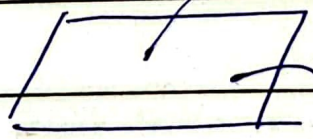
is it underdetermined or overdetermined system?

underdetermined hai since:

Q overdetermined system banay ga. Uska ~~solution~~ kya?

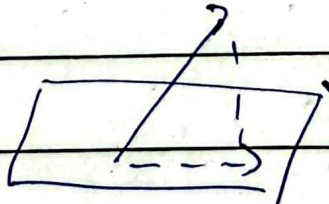
in general, no sol

$x = A^{-1}b$



so we will find approximate solution.

(least squared approximation)



$x^T(y - \hat{y}) = 0$

$x^T y - x^T \hat{y} = 0$

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \\ \theta_0 \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_0 \end{bmatrix} \rightarrow \text{constant } 1$$

$$y = \sum_{j=0}^{d} \theta_j x_j = \theta^T x$$

$$h_\theta(x) = \theta^T x \leftarrow f(x) = x^2$$

parameterized by θ — like

compact hypothesis representation

Cost function → subscript

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{n} \left(h_\theta(x)^{(i)} - y^{(i)} \right)^2$$

$h_\theta(x)^{(i)}$: predicted; $y^{(i)}$: actual label

mean sq error / sum of sq. residuals

$$\theta^* = \arg\min_\theta J(\theta)$$

Mon Tue Wed Thu Fri Sat

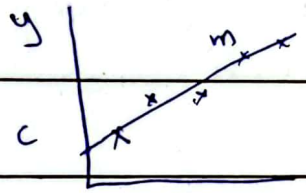
Date: __/__/20__

Lec 1

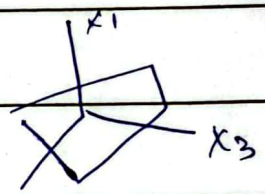
$D: \mathbb{R} \to \mathbb{R}$ $\quad D: \mathbb{R}^{d+1} \to \mathbb{R}$

	x	y
$x^{(1)}$		
$x^{(2)}$		
$x^{(n)}$		

	x_1	x_2	x_3	... x_d	y
$x^{(1)}$					$y^{(1)}$
$x^{(2)}$					
$x^{(n)}$					$y^{(n)}$



$y = mx + c$



Residuals ka sum of square errors minimize karein

Sum of sq. residuals

$y = mx + c$

$\downarrow$

$y = \theta_1 x + \theta_0$ new notational convienence

agar zyada values keh liye likhni:

$y = \theta_d x_d + \theta_{d-1} x_{d-1} + \dots + \theta_1 x_1 + \theta_0$

$= \sum_{j=0}^{d} \theta_j x_j$