Date: Calculus Subject:

Derivative: Instantaneous rate of change of a function

Instantaneous rate of change: A measure of how fast the relation between two variable is changing at any point → Derivative

★ Derivative of a func at a point is the slope of the tangent at that point

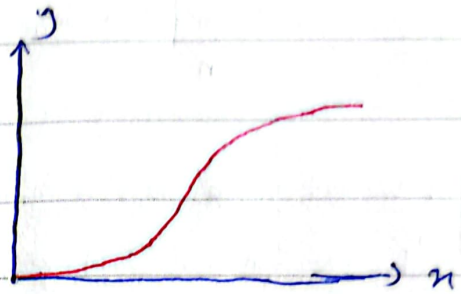
★ Max & Min of a func occurs at one of the points where derivative is zero.

$y = f(x)$

→ Derivative of f is expressed as:

$f'(x)$ → Lagrange's notation

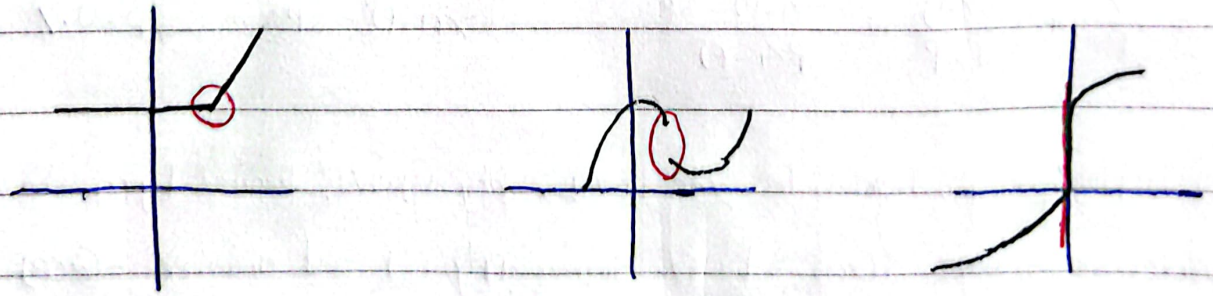
or $\frac{dy}{dx} = \frac{d}{dx} f(x)$ → Leibniz's notation



★ if $g(x) = f^{-1}(x)$ → $g'(x) = \frac{1}{f'(x)}$ (derivative of inverse funcs)

★ $f(x) = e^x$ $f^{-1}(y) = \log(y)$ (with e as base) $\frac{d}{dy} f^{-1}(y) = \frac{1}{y}$

Non-Differentiable functions



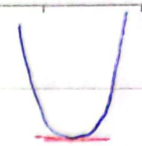
Corners / Cusps | Jump Discontinuity | Vertical tangents

★ Derivatives are used for optimization (finding max or min value of a func) → Useful for finding the model that best fits your dataset

SAMANI

Subject: Date:

Ex: Find X that would minimize this func $(x-a)^2 + (x-b)^2$



$\frac{d}{dx}\left[(x-a)^2 + (x-b)^2\right] \Rightarrow 2(x-a) + 2(x-b) = 0 \rightarrow 2x = a+b \rightarrow x = \frac{a+b}{2}$

The square loss: Minimize $(x-a_1)^2 + (x-a_2)^2 + \dots + (x-a_n)^2$

Sol: $x = \frac{a_1 + a_2 + \dots + a_n}{n}$

Coin Toss: we are gonna toss a coin 10 times. If the results are seven heads followed by three tails we win ow we lose. We have a biased coin (model) with probability of P for head and (1-P) for tails. Find P? (dataset)

$g(P) = P^7(1-P)^3 \rightarrow$ Maximize $g(P) \rightarrow \frac{dg}{dP} = 7P^6(1-P)^3 + 3P^7(1-P)^2(-1)$

$= P^6(1-P)^2(7-10P) = 0 \rightarrow P_1 = 0 \Rightarrow$ ✗ coin would alway land in tails

$P_2 = 1 \Rightarrow$ ✗ " " " " heads

$P_3 = 0.7$ ✓

another way: if $g(P)$ is maximized $\leftrightarrow$ $\log(g(P))$ is maximized

$\log(g(P)) = \log(P^7(1-P)^3) = 7\log(P) + 3\log(1-P) = G(P)$

$\frac{dG(P)}{dP} = \frac{7}{P} + \frac{3(-1)}{1-P} = \frac{7(1-P) - 3P}{P(1-P)} = 0 \rightarrow 7(1-P) - 3P = 0 \rightarrow P = 0.7$

* $-G(P)$ is logloss - a useful loss func in ML. Why negative? because $\log(x)$ when $0 < x < 1$ is negative so we want $G(P)$ to be pos so we apply (-) and minimize $-G(P)$

Why log?

1- Derivative of products is hard, derivative of sums is easy

2- Product of lots of tiny things is tiny

Date: Subject:

Functions of two variables: We have tagent plane in 3-d $f(x,y) = x^2 + y^2$

Partial Derivatives: Find Partial derivative of f with respect to x and y

$\frac{\partial f}{\partial x}$ → Treat y as a constant → $\frac{\partial f}{\partial x} = f_x = 2x$

$\frac{\partial f}{\partial y}$ → Treat x as a constant → $\frac{\partial f}{\partial y} = f_y = 2y$

Gradient: Collection of all the partial derivatives with respect to all the variables in the function → nabla $\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} 2x \\ 2y \end{bmatrix}$

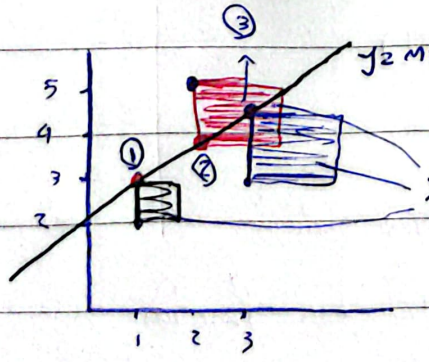
*In functions with two variables, the minimum point is where the tangent plane is parallel to the "floor". It's the same with maximum.

*If we are working with Partial derivatives, the min point (or max) is where both slopes are zero

e.g: $f(x,y) = x^2 + y^2$ → find min/max → $\frac{\partial f}{\partial x} = 0$ so → $2x = 0$ so → $x = 0$

$\frac{\partial f}{\partial y} = 0$ so → $2y = 0$ so → $y = 0$ $(0,0)$ → min

Linear Regression: Analytical approach



$y = mx + b$ → Initial line is represent where to pass connection on x-y plane

connecting

→ cost of connection to the Powerline

① = (1, m+b)

cost = $(m+b-2)^2$

② = (2, 2m+b)

cost = $(2m+b-5)^2$

③ (3, 3m+b)

cost = $(3m+b-3)^2$

Goal: Find m, b such that you minimize sum of squares cost

Subject: Date:

Cost func = $(m+b-2)^2 + (2m+b-5)^2 + (3m+b-3)^2$

→ $E(m,b) = 14m^2 + 3b^2 + 3 + 12mb - 42m - 20b$

$\frac{\partial E}{\partial m} = 28m + 12b - 42 = 0$

→ $m = 0.5 \quad b = \frac{7}{3} \quad E(\frac{1}{2}, \frac{7}{3}) \approx 4.167$

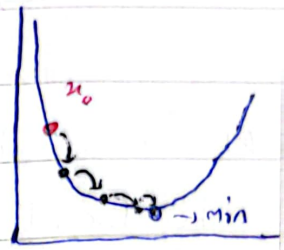
$\frac{\partial E}{\partial b} = 6b + 12m - 20 = 0$

Gradient Descent: Func: $f(x)$ goal: Find min of $f(x)$

a: Define a learning rate α, choose a starting point x_0

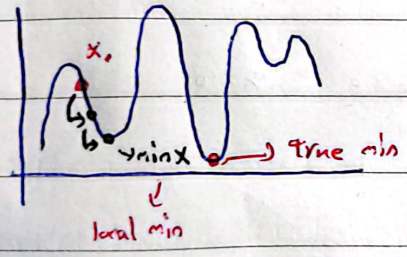
b: Update: $x_k = x_{k-1} - \alpha f'(x_{k-1})$

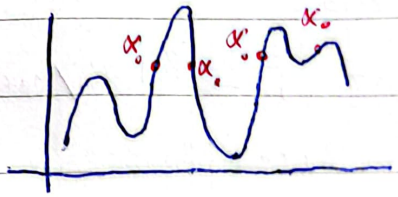
c: Repeat step 2 until you are close enough to the true min



Drawbacks: Find min!

Solution: Take multiple initial starting points





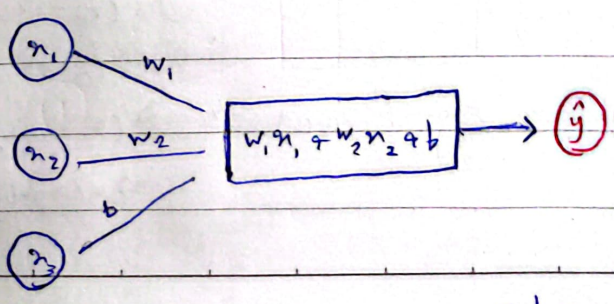
GD with two vars:

a: Define a learning rate α. choose a starting point (x_0, y_0)

b: Update $\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix} - \alpha \nabla f(x_{k-1}, y_{k-1})$

c: Repeat step 2 until you are close enough to the true minimum

Regression with a Perceptron



Prediction Function

$\hat{y} = w_1x_1 + w_2x_2 + b$

Loss function (error)

$L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$

Main goal

Find w_1, w_2, b that gives the least error

SAHANI

Date: Subject:

Using gradient descent $L(y,\hat{y}) = \frac{1}{2}(y-\hat{y})^2$ $\hat{y} = w_1 x_1 + w_2 x_2 + b$

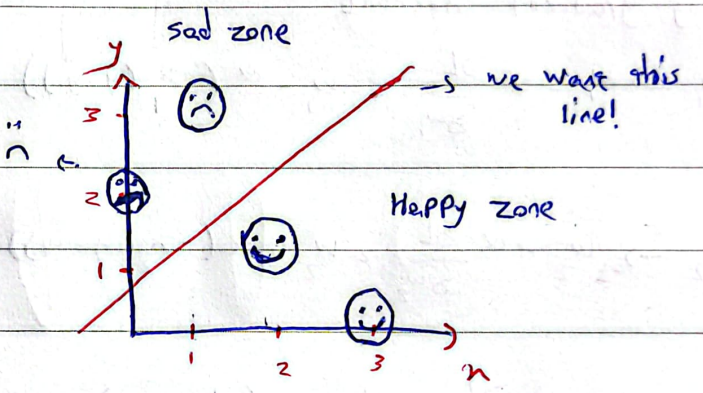
$w_1 \longrightarrow w_1 - \alpha \frac{\partial L}{\partial w_1} \longrightarrow w_1 - \alpha(-x_1(y-\hat{y}))$ $\frac{\partial L}{\partial w_1} = -(y-\hat{y})x_1$

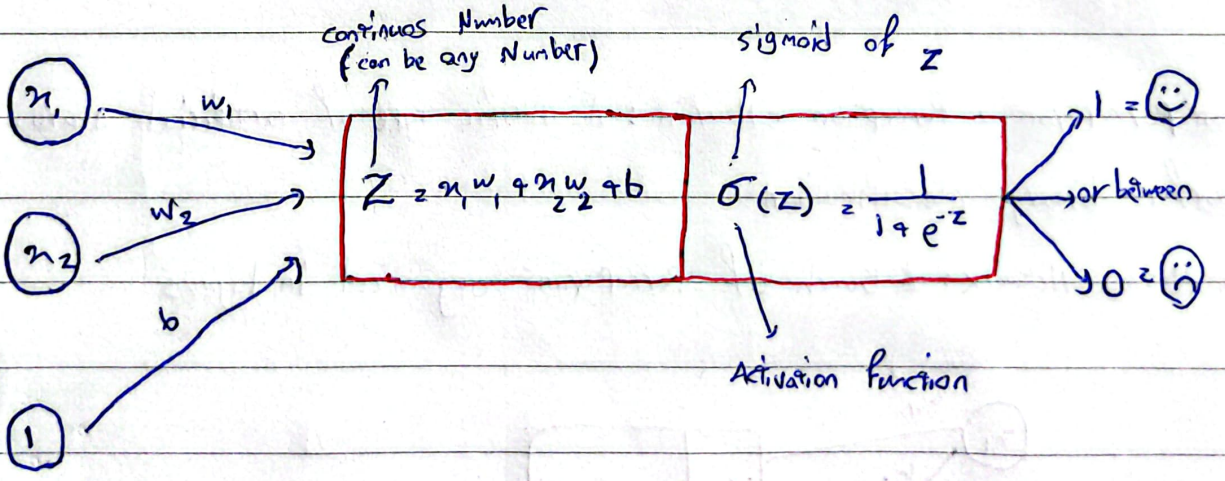
$w_2 \longrightarrow w_2 - \alpha \frac{\partial L}{\partial w_2} \longrightarrow w_2 - \alpha(-x_2(y-\hat{y}))$ $\frac{\partial L}{\partial w_2} = -(y-\hat{y})x_2$

$b \longrightarrow b - \alpha \frac{\partial L}{\partial b} \longrightarrow b - \alpha(-(y-\hat{y}))$ $\frac{\partial L}{\partial b} = -(y-\hat{y})$

Classification with a Perceptron

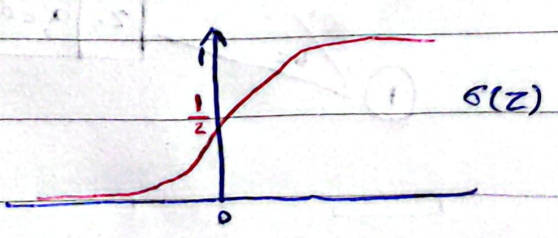
sentence	x	y	Mood
xxx	3	0	☺
yy	0	2	☹
xyyy	1	3	☹
xyx	2	1	☺





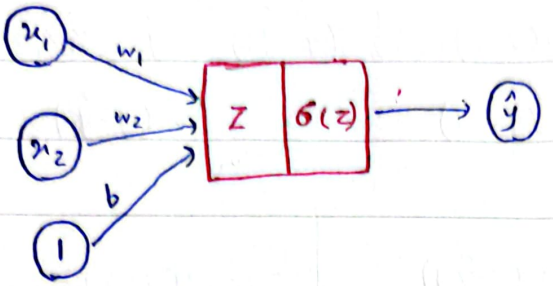
★ sigmoid func takes the entire number line and crunches into interval zero, one

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



$\frac{d}{dz}\sigma(z) = \sigma(z)(1-\sigma(z))$

Subject: Date:



Prediction func:

$\hat{y} = \sigma(w_1 x_1 + w_2 x_2 + b)$

Loss func:

$L(y, \hat{y}) = -y \ln(\hat{y}) - (1-y) \ln(1-\hat{y})$

★ we use Log-loss:

- the Math works out really nicely
- classification has a Probabilistic nature

Goal:

Find w_1, w_2, b that give $\hat{y}$ with the least error

using gradient descent

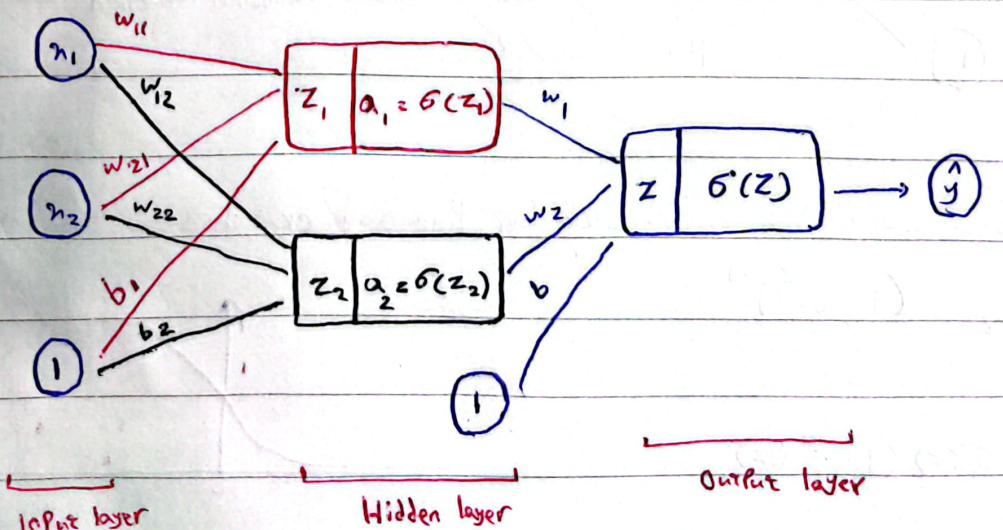
$$w_1 \rightarrow w_1 - \alpha \frac{\partial L}{\partial w_1} = w_1 - \alpha(-x_1(y-\hat{y}))$$

$$w_2 \rightarrow w_2 - \alpha \frac{\partial L}{\partial w_2} = w_2 - \alpha(-x_2(y-\hat{y}))$$

$$b \rightarrow b - \alpha \frac{\partial L}{\partial b} = b - \alpha(-(y-\hat{y}))$$

★ Perceptron = Perception + Neuron: A specific type of artificial neuron designed to perform binary classification

★ Neural Network: A Bunch of Perceptrons organized in layers



Date: Subject:

L1

$$w_{ij} \rightarrow w_{ij} + \alpha x_j w_i a_i (1-a_i)(y-\hat{y}) \qquad b_i \rightarrow b_i + \alpha w_i a_i (1-a_i)(y-\hat{y})$$

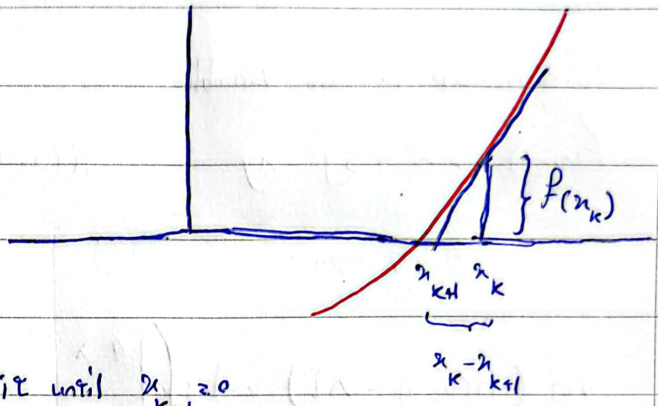
L2

$$w_i \rightarrow w_i + \alpha a_i (y-\hat{y}) \qquad b \rightarrow b + \alpha (y-\hat{y})$$

Newton's Method: is used to find the zeros of a function

$$\frac{f(x_k)}{x_k - x_{k+1}} = f'(x_k)$$

$$\rightarrow \frac{f(x_n)}{f'(x_k)} = x_k - x_{k+1}$$



$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \rightarrow \text{Update it until } x_{k+1} = 0$$

NM for optimization: minimize $g(x)$ = find zeros of $g'(x)$

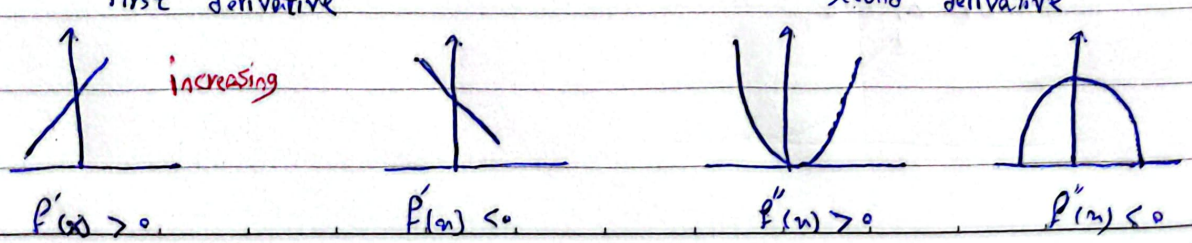
$f(x) = g'(x) \qquad f'(x) = g''(x)$

1) Start with some x_0.

2) Update $$x_{k+1} = x_k - \frac{g'(x)}{g''(x)}$$

3) Repeat until you find the candidate for minimum

Curvature



SAMANI

Subject: Date:

★ $f'(x_0) = 0$ & $f''(x_0) > 0 \longrightarrow x_0$ is a local min

★ $f'(x_0) = 0$ & $f''(x_0) < 0 \longrightarrow x_0$ is a local max

Hessian Matrix : $H(x,y) = \begin{bmatrix} f_{xx}(x,y) & f_{xy}(x,y) \\ f_{yx}(x,y) & f_{yy}(x,y) \end{bmatrix}$

★ in most cases : $f_{xy}(x,y) = f_{yx}(x,y)$

Concave up in Two variable

$f(x,y) = 2x^2 + 3y^2 - xy$ $\qquad H(0,0) = \begin{bmatrix} 4 & -1 \\ -1 & 6 \end{bmatrix}$

finding eigen values

$\det(H(0,0) - \lambda I) = \det\left(\begin{bmatrix} 4-\lambda & -1 \\ -1 & 6-\lambda \end{bmatrix}\right) = \lambda^2 - 10\lambda + 23$ ↗ $\lambda_1 \approx 6.41$ ↘ $\lambda_2 \approx 3.59$

→ both are > 0 → $(0,0)$ is a min

Concave down in Two Variables

$f(x,y) = -2x^2 - 3y^2 - xy + 15$ $\qquad H(0,0) = \begin{bmatrix} -4 & -1 \\ -1 & -6 \end{bmatrix}$

$\det(H(0,0) - \lambda I) = \lambda^2 + 10\lambda + 23$ ↗ $\lambda_1 = -3.59$ ↘ $\lambda_2 = -6.41$

→ both are < 0 → $(0,0)$ is a max

★ if $\lambda_1 > 0, \lambda_2 < 0$ or $\lambda_1 < 0, \lambda_2 > 0$ → we can't conclude anything!

or $\lambda_1 = 0$

SAMANI

Date: Subject:

Newton's method for 2 vars

1 var $x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$ or $x_k - f''(x_k) f'(x_k)$

2 vars $\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \underset{2\times 2}{H^{-1}(x_k, y_k)} \cdot \underset{2\times 1}{\nabla f(x_k, y_k)}$