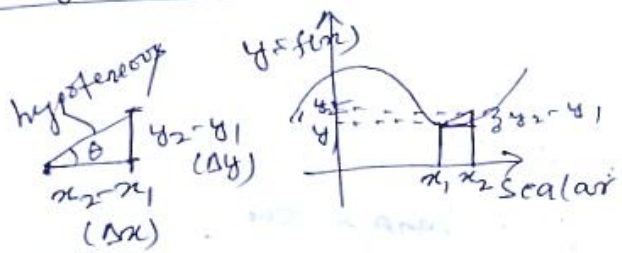
27
Solving optimization problem

27_1 : Differentiation

- optimization problems :- PCA, log. Reg., Linear Regression
- Differentiation is used a lot in ML for optimization.

Single Variable differentiation



$y = f(x)$

$\frac{dy}{dx} = \frac{df}{dx} = y' = f'$

Σ Differentiation of y w.r.t. x

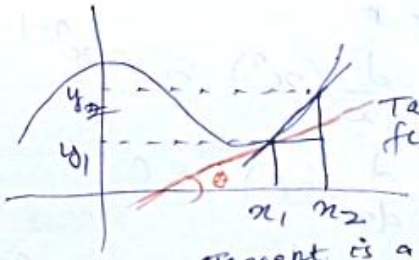
Intuitively

$\frac{dy}{dx}$ means, how much does y change as x changes.
(rate of change of y w.r.t. x)

$$\frac{y_2 - y_1}{x_2 - x_1} = \frac{\Delta y}{\Delta x}$$

$$\boxed{\frac{dy}{dx} = \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x}}$$

Tangent is the hypotenuse that we obtain as $\Delta x \to 0$.



Tangent is a line that touches the plane exactly at one point.

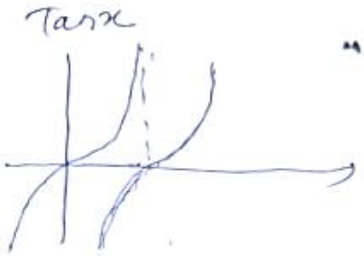
$$\tan\theta = \frac{dy}{dx} \quad \text{as} \quad \Delta x \to 0; \quad \frac{\Delta y}{\Delta x} = \frac{dy}{dx}$$

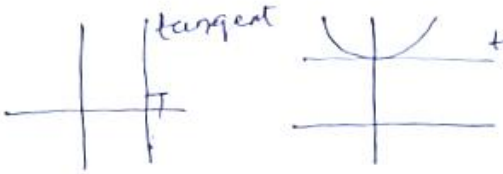
$\frac{dy}{dx}$ = slope of the tangent to f(x)

$\left[\frac{dy}{dx}\right]_{x_1}$ = slope of the tangent to f(x) @ $x = x_1$

Tan θ

angle bet^n Tangent & x-axis

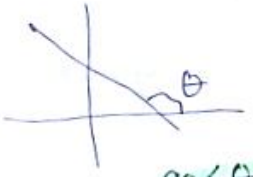




$\tan\theta = \tan 0$.

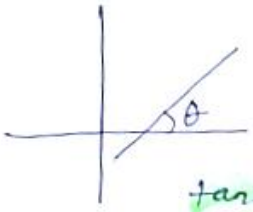
$\tan 0 = \tan 0 = 0$

$\tan(90) =$ undefined



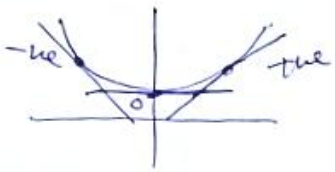
$90 < \theta \leq 180$

$\tan\theta = -ve$



$\tan\theta = +ve$.

$0 < \theta \leq 90$



Slopes tan θ.

Intuitively: $\frac{dy}{dx}$ differentiation is rate of change of y as x changes

Geometrically: It is slope of the tangent to f(x)

Basic ⊕

$\frac{d}{dx}(x^n) = nx^{n-1} \Rightarrow \frac{d}{dx}x^2 = 2x$

$\frac{d}{dx}(c) = 0 \Rightarrow \frac{d}{dx}(3) = 0$

late 1600's
- Newton
- Leibnitz

$\frac{d}{dx}(cx^n) = cnx^{n-1}$

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

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

$\frac{d}{dx}(f(x) + g(x)) = \frac{d}{dx}f(x) + \frac{d}{dx}(g(x))$

Chain Rule $\frac{d}{dx}f(g(x)) = \frac{df}{dg} \cdot \frac{dg}{dx}$

Chain Rule

$f(g(x)) = (a-bx)^2$

$\begin{cases} g(x) = a-bx \\ f(x) = x^2 \end{cases}$

$\frac{d}{dx} f(g(x)) = \frac{df}{dg} \cdot \frac{dg}{dx}$

$\frac{dg}{dx} = \frac{d}{dx}(a-bx) = \frac{d}{dx}(a) - \frac{d}{dx}(bx)$

$= 0 - b = -b$

$\frac{df}{dg} = \frac{dg^2}{dg} = 2g = 2(a-bx)$

$\frac{d}{dx} f(g(x)) = 2 \cdot (a-bx) \cdot -b$

$= -2b(a-bx)$

Online Differentiation tools

www.derivative-calculator.net

$\log(1+\exp(ax))$ [go] ↵ ln: natural logarithm

$f(x)$

$f'(x)$: gradient function

www.wolframalpha.com/input@/

derivative of $x^4 \sin x$ ↵

27-2 Online differentiation tools

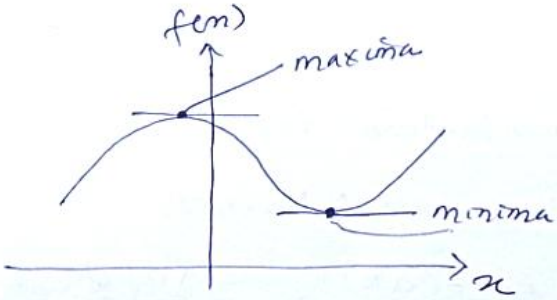
→ www.derivative-calculator.net

→ Derivative calculator with steps (www.numberempire.com/derivativecalculator.php)

www.symbolab.com/...

www.wolframalpha.com/...

27-3 Maxima and minima



At minima and maxima, slope = 0

$f(x) = x^2 - 3x + 2$, find maxima and minima.

$$\text{slope} = \frac{df}{dx} = 0.$$

$$\frac{df}{dx} = 2x - 3 + 0 = 2x - 3 = 0 \Rightarrow x = \frac{3}{2} = 1.5$$

At $x = \frac{3}{2}$; slope = 0.

Q// Is this a maxima or minima.

$f(1.5) = -0.25$

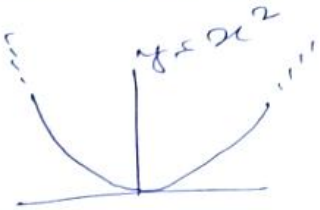
$f(1) = 0$

$\Rightarrow f(1.5) < f(1)$ ⇒ 1.5 can't be maxima.

This means we have minima at $x = 1.5$.

Example 2

$f(x) = x^2$

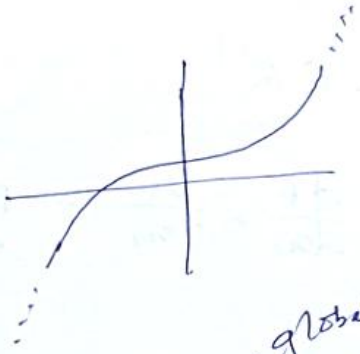


This has no maxima.
but minima at $x = 0$.

Has no minima

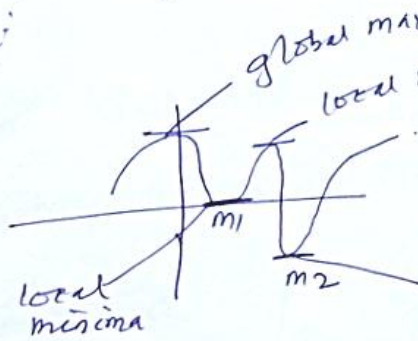
Note # The function may or may not have maxima and minima.

Example -3



no minimum
no maximum

Example -4



A function has multiple minima and maxima.

Example 5

$$f(x) = \log(1 + \exp(ax))$$

$$\frac{df}{dx} = \frac{a \cdot \exp(ax)}{1 + \exp(ax)} = 0$$ } solving this is not trivial / easy.

→ $\frac{df}{dx} = 0$, to find the minima or maxima ~~they~~ may ~~not~~ not be always possible. (for complex functions).

→ Solution: Gradient Descent to find minima & maxima.

22-4

Vector calculus : Grad

So far we have used
x : Scalar

What if x is vector.

$$y = \sum_{i=1}^{d} a_i x_i = a^T x_i$$

Example fun: $y = a^T x$ $\quad x = \langle x_1, x_2 \cdots x_d \rangle$

$a = \langle a_1, a_2 \cdots a_d \rangle$ constant

$$\frac{dy}{dx} = \nabla_x f$$ (Del or grad of w.r.t. x)

vector

$$\nabla_x f = \begin{bmatrix} \frac{df}{dx_1} \\ \frac{df}{dx_2} \\ \vdots \\ \frac{df}{dx_d} \end{bmatrix}$$ vector $\in \mathbb{R}^d$

$\frac{df}{dx_1} = \frac{\partial f}{\partial x_1}$ → partial differentiation.

$$= \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_d} \end{bmatrix}$$

$$f(x) = y = a^T x = \sum_{i=1}^{d} a_i x_i = a_1 x_1 + a_2 x_2 \cdots + a_n x_n$$

$$\nabla_x f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_d} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_d \end{bmatrix} = a$$

$\frac{\partial f}{\partial x_1} = a_1$

$\frac{\partial f}{\partial x_2} = a_2$

⋮

$$\frac{d}{dx}(ax) = a \quad | \quad \nabla_x (a^T x) = a$$

a: scalar; a: vector

Example

$$\mathcal{L}(w) = \sum_{i=1}^{n} \log\left(1 + \exp(-y_i w^T x_i)\right) + \lambda w^T w$$

$\langle x_i, y_i \rangle$: constant (come from D_{train})

variable is w.

using chain rule

$$\nabla_w \mathcal{L} = \frac{(-y_i x_i)\left(\exp(-y_i w^T x_i)\right)}{1 + \exp(-y_i w^T x_i)} + 2\lambda w = 0$$

∇_w: vector; derivation of numerator

27/5

Gradient descent algorithm

While solving maxima & minima, we were using

$$\frac{df}{dx} = 0 \quad | \quad \nabla_x f = 0$$

($\frac{df}{dx}$: scalar; $\nabla_x f$: grad, vector)

But it is not easy for complex function.

Alternative sol^n: Gradient Descent Algo. (for minima)

→ This is an iterative algorithm

$x_0 \leftarrow$ first guess of x^* (optimal)

our problem is $x^* = \arg\min_x f(x)$

Initialization $x_0 \leftarrow$ first Guess of x^*

$x_1 \leftarrow$ Iteration 1

$x_2 \leftarrow$ Iteration 2

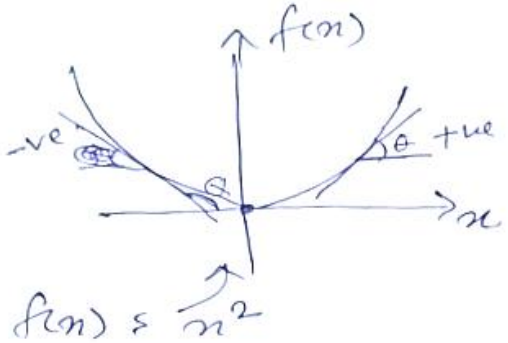
⋮

$x_k \leftarrow$ iter. k.

$\downarrow$

x^*

Gradient Ascent Algo. (for maxima):



$x^* = \underset{x}{\text{argmin}}\ f(x)$

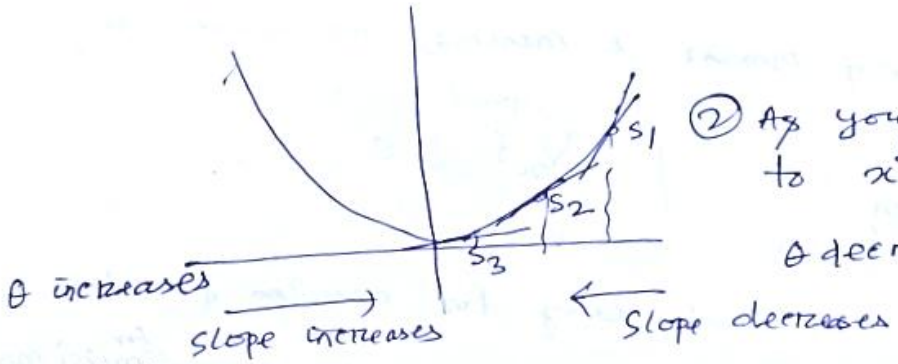
$\min f(x) = \max -f(x)$
$\max f(x) = \min -f(x)$

nd minima : slope become zero.

one side ; slope is +ve

otherside ; slope is -ve

Geometric observation ① slope changes its sign from +ve to -ve at minima.



② As you move closer to x^* ; slope reduces.

How does Gradient Descent algo. work ?

① pick an initial point (let θ x_0) at random

② find x_1 s.t. x_1 is closer x^* than x_0

Update function: make $x_1 = x_0 - \gamma \left[\frac{df}{dx}\right]_{x_0}$

γ : step size

Let step size = 1

$\Rightarrow x_1 = x_0 - 1.[\text{some +ve value}]$

$\Rightarrow x_1 < x_0$

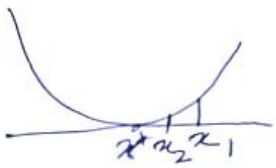
$$x_1 = x_0 - \gamma \left[\frac{df}{dx}\right]_{x_0}$$

$$= x_0 - 1 * (-\text{ve value})$$

$$= x_0 + 1$$

$$\Rightarrow x_1 > x_0$$

③ $$x_2 = x_1 - \gamma \cdot \left[\frac{df}{dx}\right]_{x_1}$$



At any iteration,

$$x_{k+1} = x_k - \gamma \cdot \left[\frac{df}{dx}\right]_{x_k}$$

— Update function of Gradient desc

$x_0, x_1, x_2 \ldots\ldots x_k$.

$$x_k = x_{k-1} - \gamma \cdot \left[\frac{df}{dx}\right]_{x_{k-1}}$$

if $(x_{k+1} - x_k)$ is very small then terminate the loop. and return. $x^* = x_k$

If x is a vector instead of scalar, replace $\frac{df}{dx}$ with $\nabla_x f$

$$x_{i+1} = x_i - \gamma \left(\nabla_x f\right)_{x_i}$$

Gradient Descent Algo.

Initialize $x^{(0)}$, $\eta > 0$

until convergence do

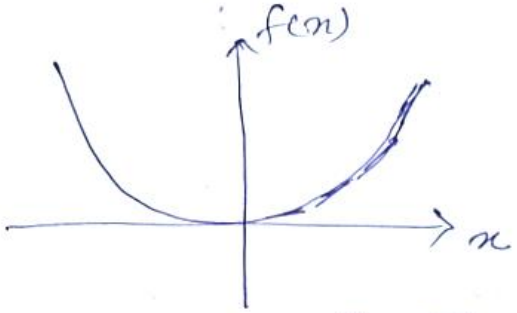
$$x^{(k+1)} = x^{(k)} - \eta \nabla f(x_k)$$

$x^* = x^{(k+1)}$

return x^*

Problem:

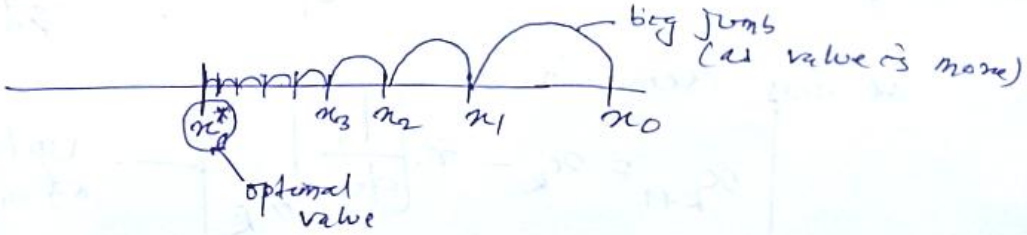
$$x^* = \underset{x}{\text{argmin}}\ f(x)$$



$$x_{k+1} = x_k - r \cdot \left[\frac{df}{dx}\right]_{x_k}$$

At every iteration the slope is decreasing.

$$\left[\frac{df}{dx}\right]_{x_0} > \left[\frac{df}{dx}\right]_{x_1} > \left[\frac{df}{dx}\right]_{x_2} > \cdots\cdots$$



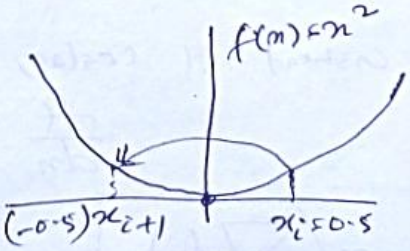
27-6

Learning rate or step size

update equation

$$x_i = x_{i-1} - r \cdot * \left[\frac{df}{dx}\right]_{x_{i-1}}$$

step size / learning rate.



let $r = 1$

$\frac{df}{dx} = 2x$

$$x_{i+1} = x_i - r \cdot \left[\frac{df}{dx}\right]_{x_i}$$

$x_{i+1} = 0.5 - 1 * (2 \times 0.5) = -0.5$

we simply jumped over x^*.

$x_{i+2} = -0.5 - 1 * (2 * -0.5) = -0.5 - 1(-1) = 0.5$

Oscillation problem

Oscillating between $x = 0.5$ and $x = -0.5$.

This happen because we made the r as constant.

So we will never converge to the optimal point.

Remedy for oscillation

① One technique is to reduce r with each iteration.

(r = function of iteration number)

$r = h(i)$ s.t. as $i \uparrow$; $r \downarrow$

i — iteration