

Application of Calculus in Linear Algebra

"Create a cost function then minimize it" lol

⇒ Derivatives are used to optimize functions

⇒ A Derivative is an instantaneous rate of change of a function.

⇒ The Derivative is defined by slope which is $\left[\frac{\text{rise}}{\text{run}} \right]$.

⇒ The Derivative of a function at a point is the slope of the tangent at that particular point.

Zero slope.

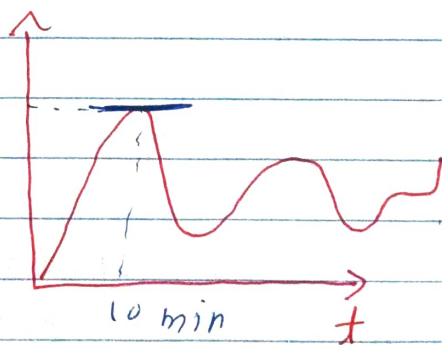
⇒ when the tangent is horizontal or $\frac{\text{rise}}{\text{run}}$ is 0.

⇒ It means there is no change at in that run interval.

Maxime →

imagine the graph on your right is a graph btwn distance & time of a ~~car~~ car, so the question is at what point the car was farthest from the initial position

& the answer is 10 min, which is also its maxima, notice that a point is maxima where the slope is zero it signifies that the car stopped otherwise if the slope is not zero the car is still running



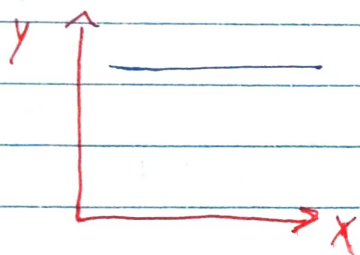
& can achieve higher maxima, so to achieve maxima or minima the slope has to be zero at the point or the derivative has to be zero.

Leibniz's notation $\Rightarrow \frac{d}{dx} f(x)$

Lagrange's notation $\Rightarrow f'(x)$

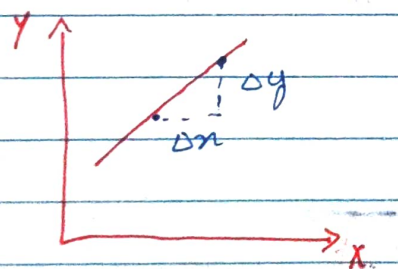
Some common derivatives

$\rightarrow$ Derivative of a constant



$\Rightarrow$ Here the line has no slope it is horizontal so its derivative is zero

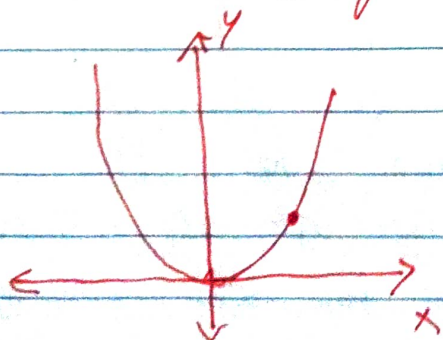
$\rightarrow$ Derivative of a line



$$f(x) = ax + b, \Rightarrow f'(x) = a$$

$$\frac{\Delta y}{\Delta x} = \frac{\text{rise}}{\text{run}} = a$$

$\Rightarrow$ Derivative of quadratic functions
 $y = x^2$



$$\text{Slope: } \frac{\Delta f}{\Delta x} = \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

$$\frac{df}{dx} = \frac{(x + \Delta x)^2 - x^2}{\Delta x}$$

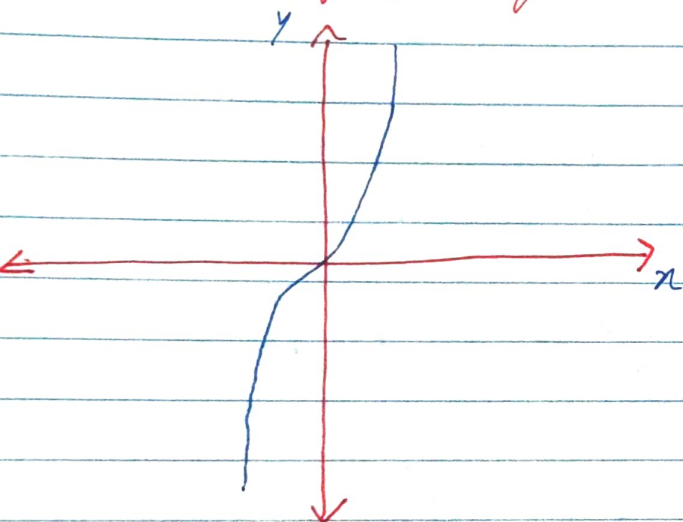
$$\Rightarrow \frac{x^2 + 2x\Delta x + \Delta x^2 - x^2}{\Delta x}$$

$$\Rightarrow \frac{2n\Delta n + \Delta n^2}{\Delta n} \Rightarrow \frac{\Delta n(2n + \Delta n)}{\Delta n} \Rightarrow \boxed{2n + \Delta n}$$

applying applying limit,

for very small change in n $\Rightarrow$ limit $\Delta n \rightarrow 0$ $2n + \Delta n \Rightarrow \boxed{2n}$ $\Rightarrow ds$

Derivative of cubic functions



cubic: $y = f(x) = x^3$

Slope: $\frac{\Delta f}{\Delta x} = \frac{(x + \Delta x)^3 - (x)^3}{\Delta x}$

$$\frac{df}{dx} = \frac{x^3 + 3x^2\Delta x + 3x\Delta x^2 + \Delta x^3 - x^3}{\Delta x}$$

$$\Rightarrow \frac{\Delta x^3 + 3x^2\Delta x + 3x\Delta x^2}{\Delta x}$$

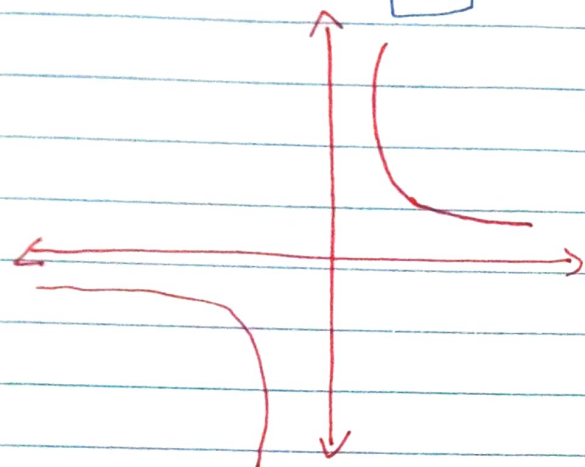
$$\Rightarrow \frac{\Delta x(\Delta x^2 + 3x^2 + 3x\Delta x)}{\Delta x}$$

$$\Rightarrow \Delta x^2 + 3x^2 + 3x\Delta x$$

applying limits $\lim_{\Delta x \rightarrow 0} 3x^2 + 3x\Delta x + \Delta x^2 \rightarrow 0$

$$\Rightarrow \boxed{3x^2}$$

Derivative of $\frac{1}{x}$.



$$\text{Inverse: } y = f(x) = x^{-1} = \frac{1}{x}$$

$$\text{Slope: } \Rightarrow \frac{\Delta f}{\Delta x} = \frac{(x + \Delta x)^{-1} - x^{-1}}{\Delta x}$$

$$\Rightarrow \frac{\frac{1}{x + \Delta x} - \frac{1}{x}}{\Delta x}$$

$$\Rightarrow \frac{\frac{x - (x + \Delta x)}{x(x + \Delta x)}}{\Delta x}$$

$$\Rightarrow \frac{\frac{-\Delta x}{x^2 + x\Delta x}}{\Delta x}$$

$$\Rightarrow \frac{1}{\Delta x} \cdot \frac{-\Delta x}{x^2 + x\Delta x}$$

$$\Rightarrow -\frac{1}{x^2 + x\Delta x}$$

applying limit's $\Delta x \rightarrow 0$

$$\Rightarrow \lim_{\Delta x \rightarrow 0} -\frac{1}{x^2 + x\Delta x \rightarrow 0}$$

$$\Rightarrow \boxed{-\frac{1}{x^2}} \quad \text{Ans}$$

Inverse function

$\Rightarrow$ an inverse function undoes the thing by the done by the function.

$g(x)$ & $f(x)$ are inverses, so, $g(x) = f^{-1}(x)$
 $\Rightarrow g(f(x)) = x$

$$\Rightarrow g'(y) = \frac{1}{f'(x)}$$

$$g'(x) = \frac{1}{f'(x)}$$

Derivative of trigonometric functions

Proof is tough but i got it.

$$f'(\sin x) = \cos x$$

$$f(x) = \sin(x)$$

$$f'(x) = \cos(x)$$

$$g(x) = \cos(x)$$

$$g'(x) = -\sin(x)$$

Meaning of exponential (e)

→ Euler number

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = 2.718 \dots$$

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

Logarithms of Derivative

$$e^? = 3$$

$$? = \log(3)$$

log of (3) will be the power of e
& it will give you (3).

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

Existence of the Derivative

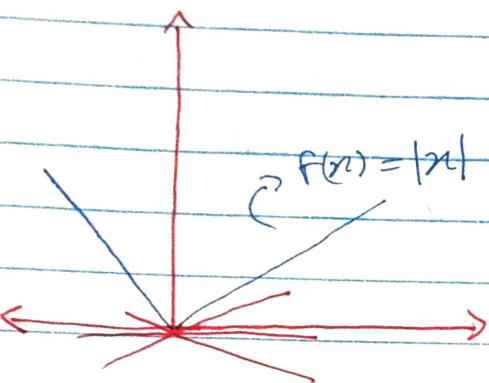
The function which aren't differentiable at every point
are called non-differentiable function

differentiable function

for a function to be differentiable at a point:
The derivative has to exist for that point, meaning you
can draw a ^(well defined) tangent at that point.
for a fn to be differentiable at an entire interval

The derivative has to exist for every point in the interval

Non-differentiable example:-



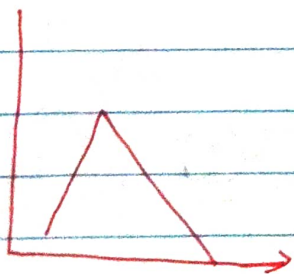
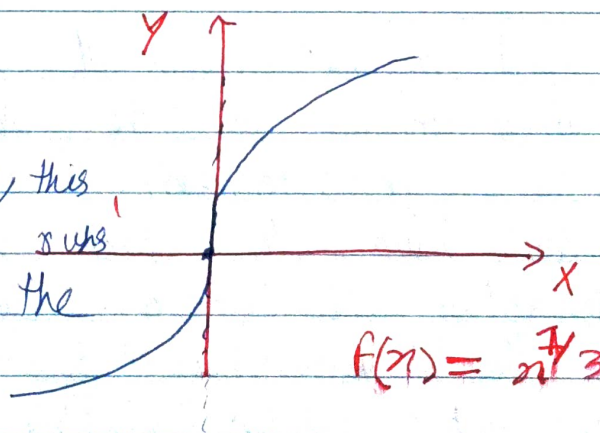
The absolute value function

$$f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ -x, & \text{if } x < 0 \end{cases}$$

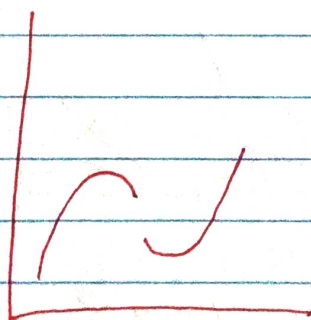
at $x=0$, the derivative doesn't exist because the tangent can't be defined there

$\Rightarrow$ A vertical line doesn't have a well defined slope so when a tangent is vertical or perpendicular to y -axis it is non-differentiable.

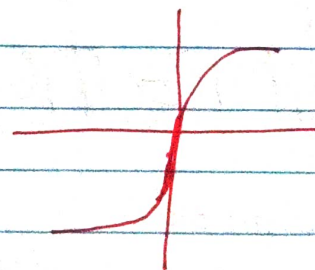
$\Rightarrow$ for the graph, at $x=0$, this graph has a tangent line that runs straight up parallel to the y -axis.



corner's



cusps



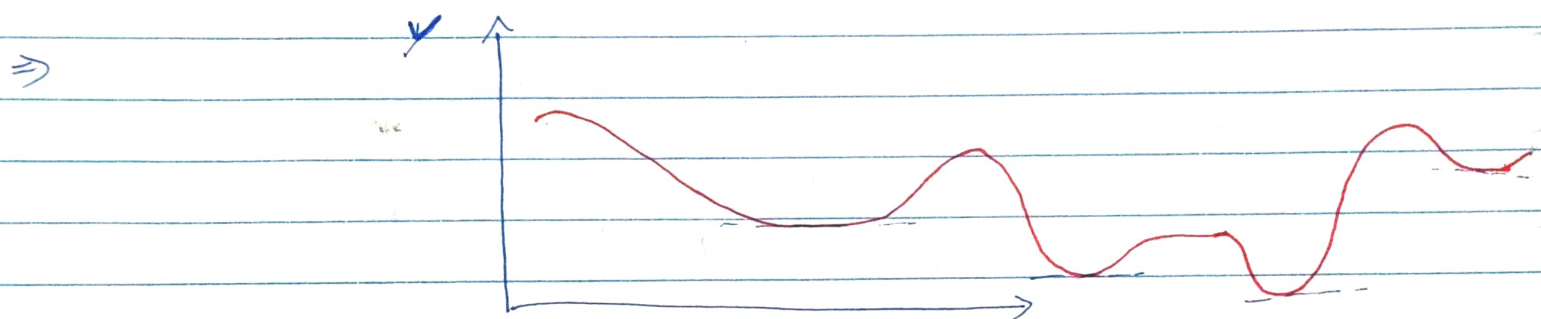
non-vertical tangent's

$\Rightarrow$ These all f's are non-differentiable.

Introduction to Optimization:

Optimization is finding max & min value of a function.

⇒ Let's say you want to find out a model that best fits your dataset, so what you do is calculate error fn that tells you how far are you from an ideal model. & when you minimize this error fn then you have the best model.



⇒ local minima is all the minima^x

⇒ global minima is the absolute ~~closest~~ close we can go to zero

⇒ You have to check at each point where the derivative is zero whether it's minima or maxima & then you have to compare all the minimas to get global minima

Optimization of squared loss

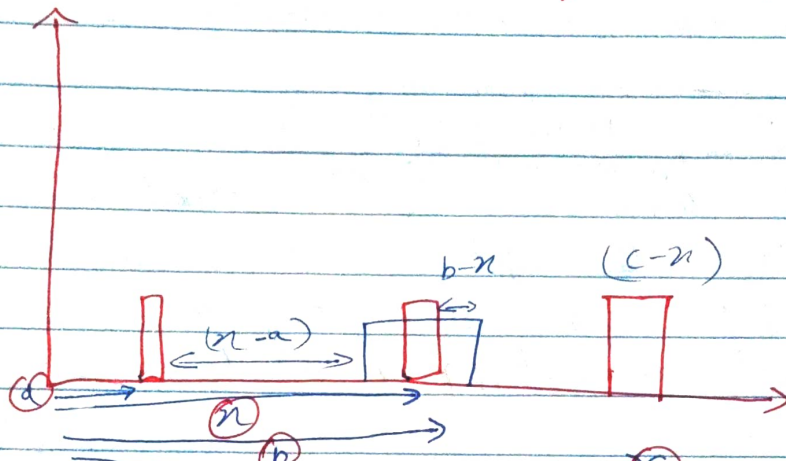
minimize

$$(x-a_1)^2 + (x-a_2)^2 + \dots + (x-a_n)^2$$

Just derivate the above eqn for solⁿ

$$\text{Soln, } x = \frac{a_1 + a_2 + \dots + a_n}{n}$$

let's say you have to find a place where you can place your home & you will get in a way you get charged minimum from all of the power lines.



→ all the red squares are power lines & the blue one is the house

⇒ So let's say you get charged squared of the distance so

⇒ So, the cost function will be.

$$\Rightarrow (n-a)^2 + (n-b)^2 + (n-c)^2$$

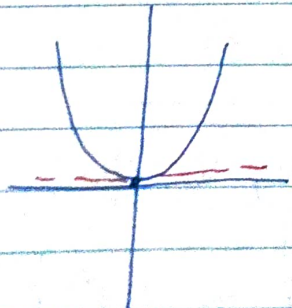
→ here we take $(n-c)$ instead of $(c-n)$ because when we are squaring -ve doesn't matter.

So to find the best place for the house

we do, derivation of cost function to find minima

we set the derivative at zero to show that is the min point.

$$\frac{d}{dn} ((n-a)^2 + (n-b)^2 + (n-c)^2) = 0$$



$$\Rightarrow 2(n-a) + 2(n-b) + 2(n-c) = 0$$

$$\Rightarrow (n-a) + (n-b) + (n-c) = 0$$

$$3n = a+b+c$$

$$n = (a+b+c)/3$$

Optimization of log-loss

⇒ imagine you have to toss a coin & you can choose the coin {The coin is biased} to get the output.

H	H	H	H	H	H	H	T	T	T
---	---	---	---	---	---	---	---	---	---

Then what will be the perfect coin with perfect bias to get to the output,

let's say prob of getting a head is P

& Probability of getting a tails is $(1-P)$.

So,

$$\Rightarrow P \times P \times P \times P \times P \times P \times P \times (1-P) \times (1-P) \times (1-P) = g(P)$$
$$\Rightarrow P^7 (1-P)^3 = g(P)$$

$$\Rightarrow \frac{d}{dP} (g(P)) = \frac{d}{dP} (P^7 (1-P)^3)$$

to get to the derivative easily let's take log loss.

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

$$\text{let's say } \log(g(P)) \Rightarrow G(P) \Leftrightarrow \log P^7 + \log (1-P)^3$$

because if $(g(P))$ is maximum the so

as $\log(g(P))$

$$\Rightarrow \boxed{7 \log P + 3 \log (1-P)}$$

$$\frac{d G(P)}{dP} \Rightarrow \frac{d}{dP} (7 \log P + 3 \log (1-P))$$

$$\Rightarrow 7 \frac{1}{P} + 3 \frac{1}{(1-P)} (-1) = 0$$

$$\Rightarrow \frac{7(1-P) + (-3P)}{(P)(1-P)} = 0$$

$$\Rightarrow 7(1-p) - 3p = 0$$

$$\Rightarrow 7 - 7p - 3p = 0$$

$$\Rightarrow -10p = -7$$

$$\Rightarrow \boxed{p = 0.7}, (1-p) = 1 - 0.7 \Rightarrow \boxed{0.3}$$

So the best value coin is the one with Prob of heads being (0.7) & prob of tails being (0.3)

In ML ~~here~~, we take " $-G(p)$ " as logloss because log is negative btw 0-1.

Tangent Plane

$\Rightarrow$ If a function has 2 variables then, ~~the~~

$$f(x, y) = x^2 + y^2 \quad \text{or} \quad z = x^2 + y^2,$$

it will be plotted in 3d,

& Now the tangent's are not lines they are planes in Three dimensions.

Using the above function, ① let's fix $y=4$,
② let's fix $x=2$,

①

$$f(x, 4) = x^2 + (4)^2$$

$$\boxed{\frac{d}{dx} (f(x, 4)) = 2x}$$

②

$$f(2, y) = 2^2 + y^2$$

$$\boxed{\frac{d}{dy} (f(2, y)) = 2y}$$

Here after fixing two points we have two tangent lines that cross each other & any two lines that cross each other ~~are~~ represent a unique plane.

Partial derivative

Notations $\Rightarrow f_{xx} = \frac{\partial^2 f}{\partial x^2}$, $f_{yy} = \frac{\partial^2 f}{\partial y^2}$

$f(x,y)$

Partial derivative of f with respect to x , Partial derivative of f with respect to y .

Gradient

$$F(x,y) = x^2 + y^2$$

$\Rightarrow$ It is a vector containing all the partial derivative of a given function,
for the function above,

Gradient $\Rightarrow$

$$\begin{bmatrix} 2x \\ 2y \end{bmatrix}$$

In general

$\nabla f =$

(nabla)

$$\begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

find gradient of $f(x,y) = x^2 + y^2$, at point (2,3).

Solⁿ $\Rightarrow$

$$\nabla f(x,y) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \Rightarrow \begin{bmatrix} 2x \\ 2y \end{bmatrix} \Rightarrow \text{putting the values of } x \text{ \& } y.$$
$$\Rightarrow \begin{bmatrix} 2 \times 2 \\ 2 \times 3 \end{bmatrix} \Rightarrow \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

Gradients & maxima/minima

$$f(x,y) = x^2 + y^2$$

$\Rightarrow$ in fn with 1 variable we have, tangent line with slope = 0, $\parallel$ to the y-axis or floor.
in fn with 2 variable we have, tangent planes (with the partial differentiation of each ~~etc~~ variable being zero) are $\parallel$ to the floor, meaning all the tangent lines in the tangent plane are also $\parallel$ to the floor.

$\Rightarrow$ minimum happens when both of the tangent lines in tangent planes are zero, which is given by partial derivative of each variable.

$$\text{so } \frac{\partial f}{\partial x} = 0, \quad \frac{\partial f}{\partial y} = 0$$
$$2x = 0, \quad 2y = 0$$

$$x = 0, \quad y = 0$$

so at the coordinates given above the fn is minimum

motivation for optimization in 2 variable

$$T = f(x, y) = 85 - \frac{1}{90} x^2(x-6) y^2(y-6)$$

$$\frac{\partial f}{\partial x} = -\frac{1}{90} x(3x-12) y^2(y-6) = 0 \quad \text{--- (i)}$$

$$\frac{\partial f}{\partial y} = -\frac{1}{90} x^2(x-6) y(3y-12) = 0 \quad \text{--- (ii)}$$

from eqn (i)

$$\left. \begin{array}{l} 3x-12=0 \\ y^2=0 \\ y-6=0 \\ x=0 \end{array} \right\} \begin{array}{l} x=4 \\ y=0 \\ y=6 \\ x=0 \end{array}$$

from eqn (ii)

$$\left. \begin{array}{l} x^2=0 \\ x-6=0 \\ y=0 \\ 3y-12=0 \end{array} \right\} \begin{array}{l} x=0 \\ x=6 \\ y=0 \\ y=4 \end{array}$$

Now, we make all the possible combination of coordinate
of $(x, y) \Rightarrow (4, 0), (4, 6), (4, 4),$

$$\Rightarrow (0, 0), (0, 6), (0, 4)$$

$$\Rightarrow (6, 0), (6, 4), (6, 6)$$

These above are all possible combinations.

So, to select the global minimum

Here some points are out of the surge (any digit bigger than 5) so we eliminate that

So, the remaining are

$(4,0)$, $(4,4)$, $(0,0)$, $(0,4)$, ~~$(6,0)$, $(6,6)$~~

& so analysing the temperature values give by the graph we ~~say~~ can say that $(4,4)$ is the minimum.

Linear regression optimization using gradient

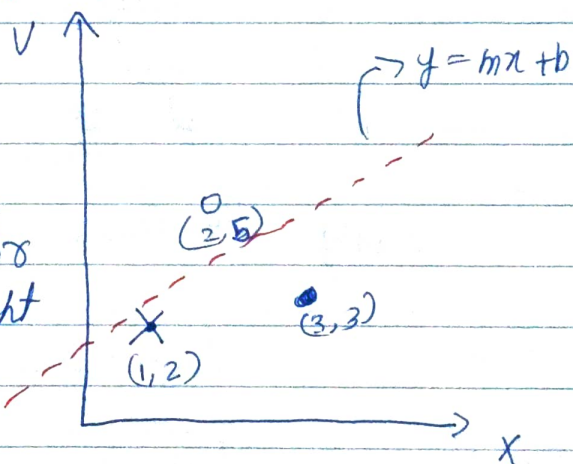
x y •
 $(1,2)$ $(2,5)$ $(3,3)$

⇒ we have to find optimal place for the line that goes in a straight line in such way that you reduce the total cost of connecting to the three points.

⇒ Here the wire that connects the points to the line has to be $\perp$ to Y-axis

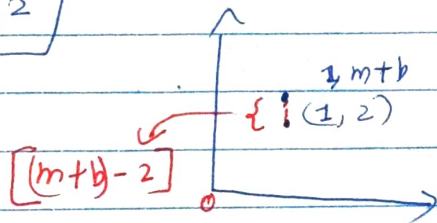
⇒ & the cost of connecting the wire has to be the square of the $\perp$ distance of the point from the line to Y-axis

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



focus on the "x", it is located at the coordinate (1, 2) & it needs to connect to the fibre line, so, the point where it connects is (1, m+b) why? because if you assume $x=1$ then through $y=mx+b$ $y=m+b$

so the distance is $(m+b-2)$ & cost is $(m+b-2)^2$



so, similarly the cost of "0" is $(2m+b-5)^2$ & cost of "•" is $(3m+b-3)^2$

adding the costs we get,

$$E(m, b) \Rightarrow (m+b-2)^2 + (2m+b-5)^2 + (3m+b-3)^2$$

$$E(m, b) \Rightarrow 14m^2 + 3b^2 + 38 + 12mb - 42m - 20b \Rightarrow \text{This is the cost function}$$

To minimize this function we have to calculate

$$\frac{\partial E}{\partial m} = 0$$

$$\frac{\partial E}{\partial b} = 0$$

$$\Rightarrow 28m + 12b - 42 = 0 \text{ (i)}$$

$$\Rightarrow 6b + 12m - 20 = 0 \text{ (ii)}$$

Solving the eqn (i) & (ii) we get

$\Rightarrow$

$$m = 0.5$$

$$b = 2.33$$

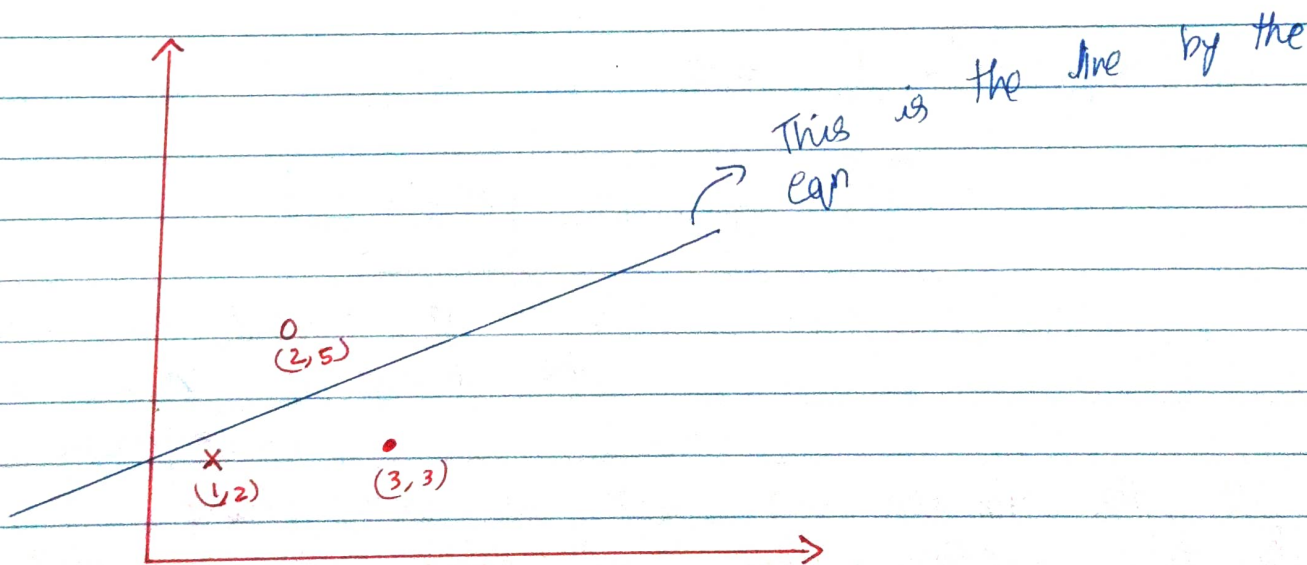
Putting these values in cost fn we get

$$E(m=1/2, b=7/3) \approx \boxed{4.167}$$

$\hookrightarrow$ This is the minimum value of the cost.

Now plot the line on the graph

$\boxed{y = \frac{1}{2}x + \frac{7}{3}}$ $\rightarrow$ This line will give us optimal solutions.



& the distance of the point's to line (note that the distance has to be $\perp$ to y -axis) will give the most optimal solution.

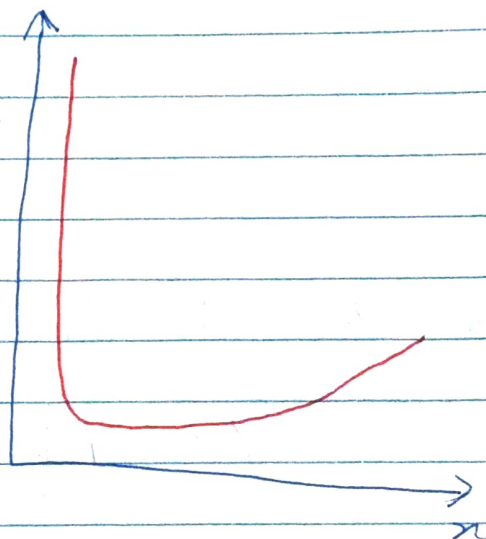
But what if we had many variables & many unknowns, then at that point we use gradient descent.

Gradient descent in one variable

Let's take example of a hard function.

$$F(x) = e^x - \log(x)$$

→ not so accurate graph



find minimum here →

So, derivating w.r.t x & putting it equal to zero

$$\frac{d}{dx}(e^x - \log x) = 0$$

$$\Rightarrow e^x - \frac{1}{x} = 0$$

$$e^x = \frac{1}{x}$$

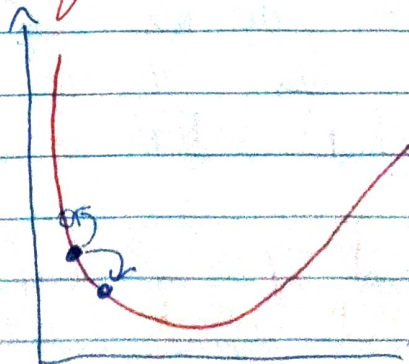
$$x = \frac{1}{e^x}$$

→ it's hard to solve & the answer is $x = 0.5671 \dots$
aka ω constant

So, the minimum we get is very hard to solve & the other way of doing is gradient desc.

method ①

So, we take an initial point which we denote here by " $\bullet$ "; & move it in both the direction & find which is the better one (The better one is the one that has low value)

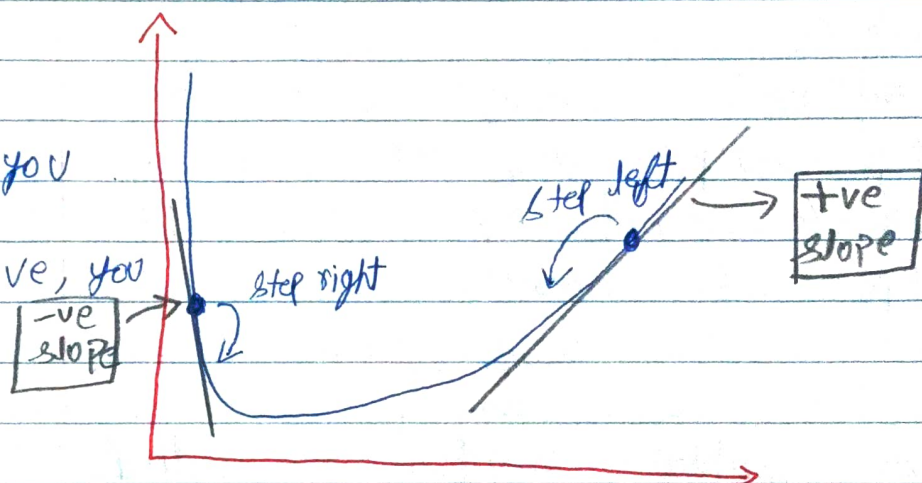


& we iterate this process till we get to the minimum point & on the minimum point if you try to get go either way the value will only increase & so we say we have achieved the minima.

method ②

⇒ if the slope is -ve, you move to the right

⇒ if the slope is +ve, you move to the left.



So, your formula for new point should be

⇒ new point = old point - slope

$$\Rightarrow \boxed{x_1 = x_0 - f'(x_0)}$$

However, there is a caveat,

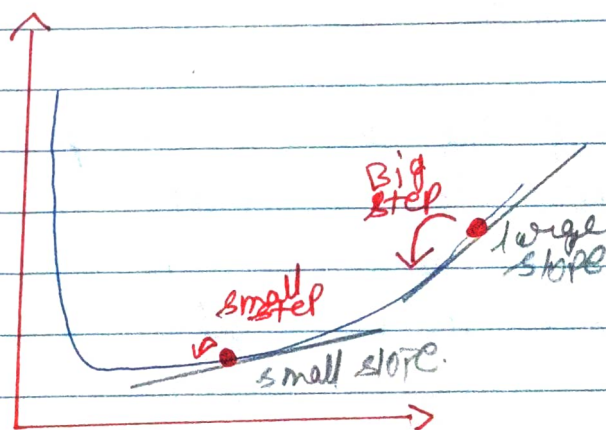
⇒ for a second think if you are at the steep part of the curve, then the value of the derivative will be very large ($f'(x_0)$) so to counter that you can multiply ($f'(x_0)$) with a small number to change its value & make it small. Why large steps are bad well if you take large steps you might miss the minima so you take small steps.

The new equation after correction is

$$\boxed{\pi_1 = \pi_0 - \alpha f'(\pi_0)} \quad \text{or} \quad \boxed{\pi_n = \pi_{n-1} - \alpha f'(\pi_{n-1})}$$

here " α " is called "learning rate" which is a very small number e.g. 0.001 or 0.01.

⇒ it " α " enables us to take large steps when the slope is large & small steps when the slope is small.



This whole method is called **GRADIENT DESCENT**

This is a fast algorithm so it can be iterated many many times quickly

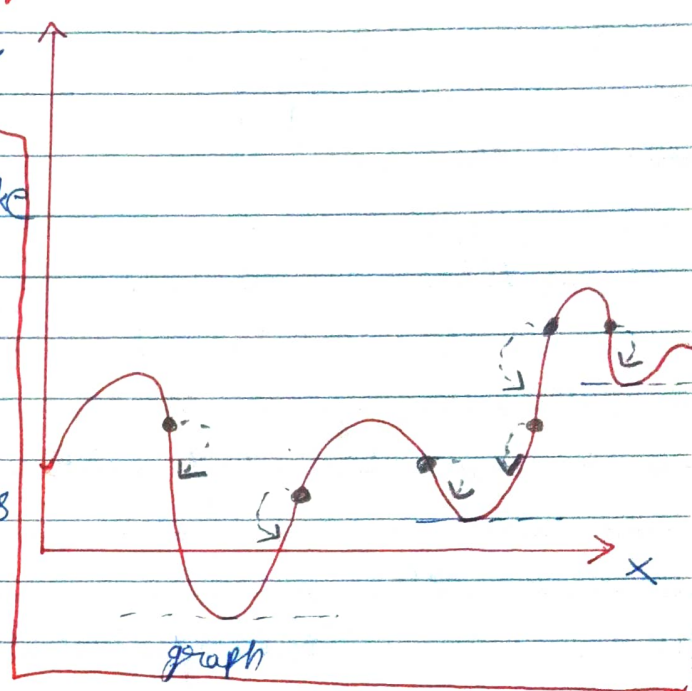
⇒ In gradient descent you never need to solve derivative equals to zero. You only need to know the derivative & apply it in the algorithm

So, what is a good α or learning rate?

it's a research problem & no one knows how to find good learning rate, if you have a big learning rate then you may never find a minima & if you have a very small learning rate you might take a lot of time to actually reach the minima

⇒ what if the fn has a lot of minimas?

⇒ In this case instead of taking one point & minimizing through that one point we take multiple points & we try to find the minima all those points gives us minima & the least value point will be our global minima & others will be local minima.

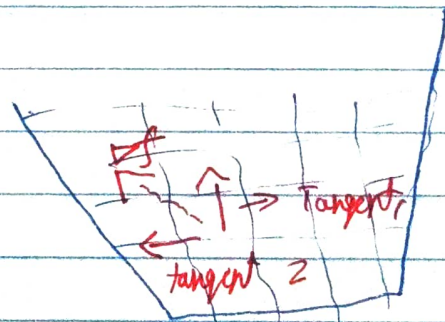


Optimization using Gradient descent with two variables

initial position : (x_0, y_0)

Direction of greatest ascent : ∇f

Direction of greatest descent : $-\nabla f$



Updated position : $(x_0, y_0) - \alpha \nabla f \rightarrow$ This will give new point (x_1, y_1)

let's say you have a cost fn $\rightarrow f(x, y) = 85 - \frac{1}{90} x^2 (x-1) y^2 (y-1)$

[we start at $x = 0.5, y = 0.6$]

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} -\frac{1}{90} x (3x-12) y^2 (y-6) \\ -\frac{1}{90} x^2 (x-6) y (3y-12) \end{bmatrix}$$

$$\nabla f(0.5, 0.6) = \begin{bmatrix} -0.1134 \\ -0.0935 \end{bmatrix}$$

let's say our learning rate " α " is = 0.05

So,

according to the equation,

$$(x_1, y_1) = (x_0, y_0) - \alpha \nabla f$$

$$= \begin{bmatrix} 0.5 \\ 0.6 \end{bmatrix} - 0.05 \begin{bmatrix} -0.1134 \\ -0.0935 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 0.5 \\ 0.6 \end{bmatrix} - \begin{bmatrix} -0.00567 \\ -0.04675 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 0.5057 \\ 0.6047 \end{bmatrix}$$

{ So, the new value of, x is 0.5057
 y is 0.6047 }

& iterate this process the the values don't change
 or change very very little

general
The equation for gradient descent in 2 variable is given by,

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

E.g., From previous lecture you found the loss fn $14m^2 + 3b^2 + 38 + 12mb - 42m - 20b$ of a given problem, now we are finding the right value of m & b using gradient descent
 $\Rightarrow$ finding gradient

$$\nabla E = \begin{bmatrix} 28m + 12b - 42 \\ 6b + 12m - 20 \end{bmatrix}$$

So, take random coordinates for $(m \& b)$ & then apply gradient descent. Let's say random coordinates are (m_0, b_0) .

$$\begin{bmatrix} m_1 \\ b_1 \end{bmatrix} = \begin{bmatrix} m_0 \\ b_0 \end{bmatrix} - \alpha \nabla E(m_0, b_0)$$

iterate it & you will get the value.

Another Example :-

Let's say you have to predict the sales when there is certain budget is allotted, Then the tool that we gonna use is Linear regression.

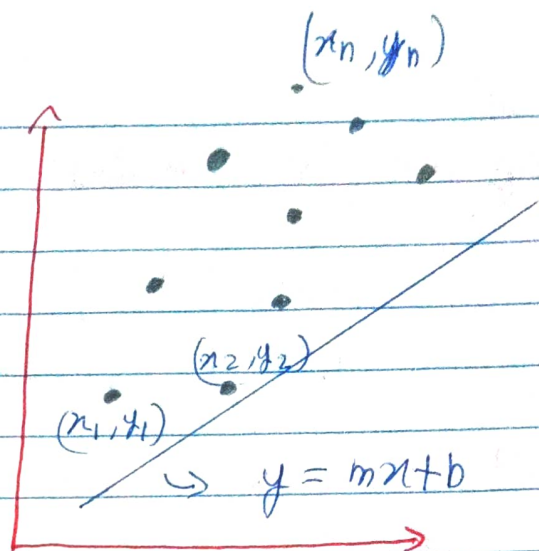
$$[y = mx + b]$$

The standard loss function in linear regression is given by

$$\frac{1}{n} \sum$$

$$\frac{1}{n} \sum$$

$$\frac{1}{n} \sum_{i=1}^n (mx_i + b - y_i)^2$$



for this example

$$L(m, b) = \frac{1}{2n} [(mx_1 + b - y_1)^2 + \dots + (mx_n + b - y_n)^2]$$

Here, we divide the loss by 2 just to make the calculation easy, because after the derivative the two in the power will come down & get cancelled by the 2 we divide & it will have no effect on the minima of the function.

So we start doing gradient descent.

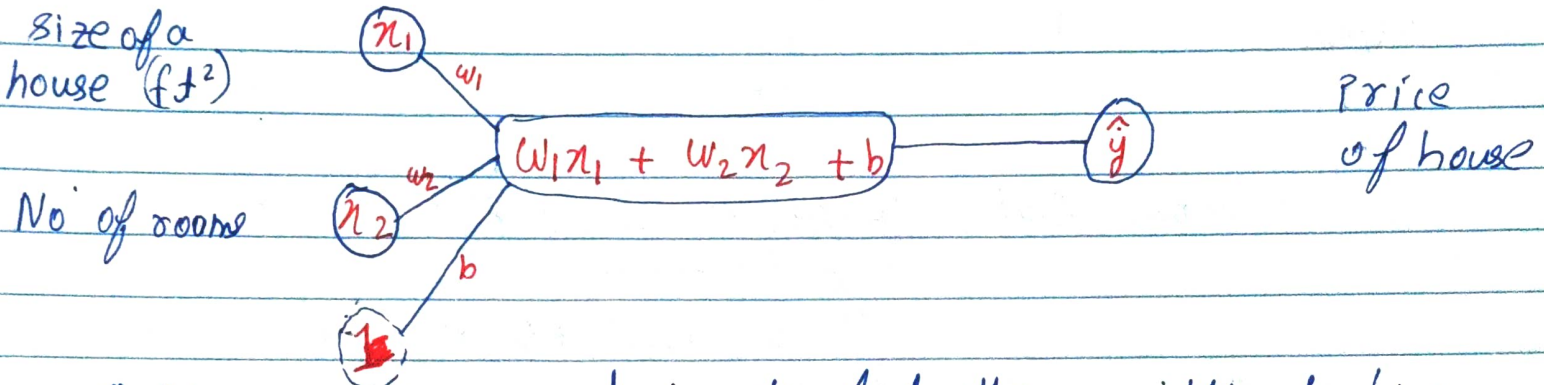
$$\begin{bmatrix} m_N \\ b_N \end{bmatrix} = \begin{bmatrix} m_{N-1} \\ b_{N-1} \end{bmatrix} - \alpha \nabla L_1(m_{N-1}, b_{N-1})$$

we stop the gradient descent when the changes become very very small.

Regression with a Perceptron

inputs

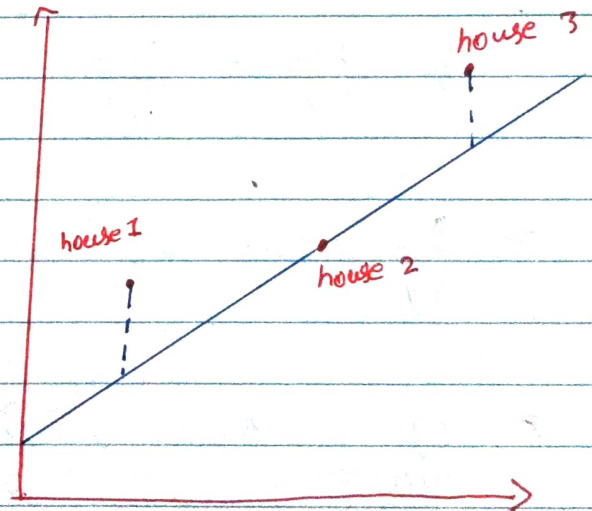
output



Our main goal is to find the weights & biases that will optimise the predictions. & i.e. Reduce the errors in the predictions

Mean Squared Error

	original value y	model predicted value $\hat{y}$	error value $\frac{1}{2}(y - \hat{y})^2$
house 1	20k	15k	error
house 2	30k	30k	error
house 3	50k	45k	error



Prediction function

$$\hat{y} = W_1x_1 + W_2x_2 + b$$

loss function:-

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

To find optimal values for:- W_1, W_2, b you need gradient descent.

$$w_1(\text{new}) \rightarrow w_1 - \alpha \frac{\partial L}{\partial w_1}$$

$$w_2(\text{new}) \rightarrow w_2 - \alpha \frac{\partial L}{\partial w_2}$$

$$b(\text{new}) \rightarrow b - \alpha \frac{\partial L}{\partial b}$$

we know that,

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

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

$$\Rightarrow \frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_1}, \Rightarrow \frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_2} \quad \checkmark$$

$$\Rightarrow \frac{\partial L}{\partial b} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial b}$$

$$\Rightarrow \frac{\partial L}{\partial \hat{y}} = \frac{1}{2} \cdot 2 (y - \hat{y}) \cdot (-1) = -(y - \hat{y})$$

$$\Rightarrow \frac{\partial \hat{y}}{\partial w_1} = x_1, \Rightarrow \frac{\partial \hat{y}}{\partial w_2} = x_2, \Rightarrow \frac{\partial \hat{y}}{\partial b} = 1$$

$$\text{Now, } \frac{\partial L}{\partial w_1} = -(y - \hat{y}) x_1, \frac{\partial L}{\partial w_2} = -(y - \hat{y}) x_2, \frac{\partial L}{\partial b} = -(y - \hat{y})$$

Now,

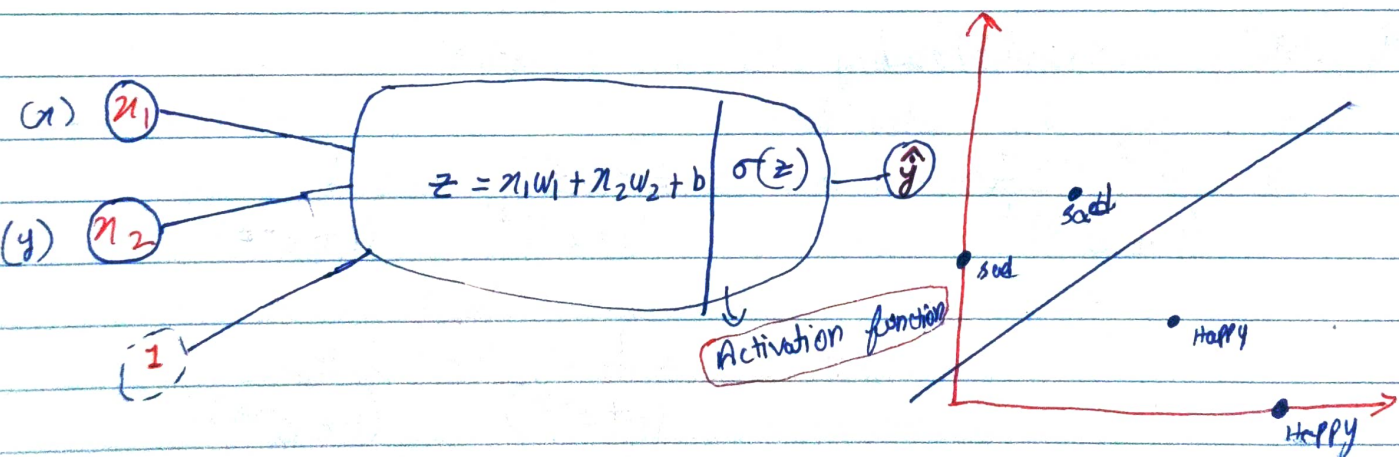
$$\text{New } (w_1) = w_1 - \alpha (-\pi_1 (y - \hat{y}))$$

$$\text{New } (w_2) = w_2 - \alpha (-\pi_2 (y - \hat{y}))$$

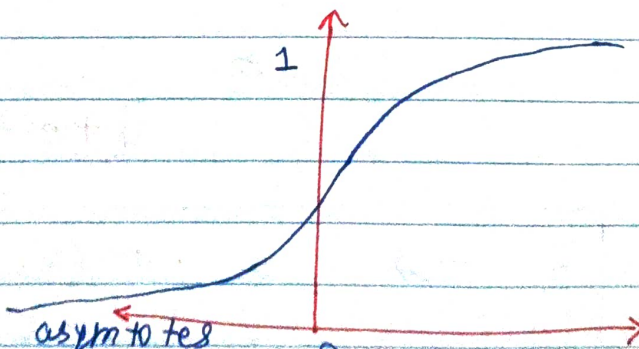
$$\text{New } (b) = b - \alpha (- (y - \hat{y}))$$

Classification with a perceptron.

Words	X	Y	Mood
x x x	3	0	Happy
y y	0	2	Sad
x y y y	1	3	Sad
x y x	2	1	Happy



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



⇒ This functions have asymptotes over zero & one, which means it will never touch these numbers.

Derivative of a sigmoid function

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

$$\Rightarrow \frac{d}{dz} (1+e^{-z})^{-1}$$

$$\Rightarrow -1 (1+e^{-z})^{-2} \cdot \frac{d}{dz} (e^{-z})$$

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

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

$\Rightarrow$ adding & subtracting 1 in numerator

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

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

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

So,

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

Ans

Here in the classification, let's say,

you give (x) $x_1 = 1$ | $w_1 = 4.5$ | & bias = 2
 (y) $x_2 = 3$ | $w_2 = 1.5$

& the output that comes out has some error,

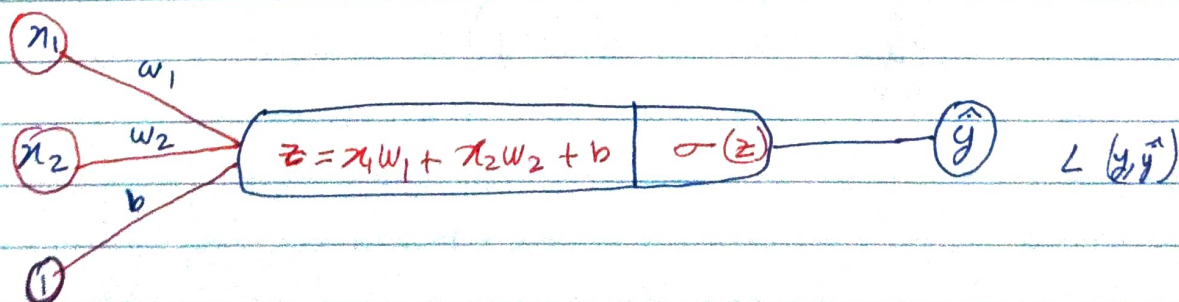
So to fix the error in classification problem we use LOG LOSS.

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

if y is one, then the $\hat{y}$ (hat) if $\hat{y}$ (hat) is close to one the loss function is small & if $\hat{y}$ (hat) is close to zero the loss function is big.

⇒ This function has probabilistic nature to it, & classification problems are highly probabilistic, because you can think of output $\hat{y}$ as a probability.

⇒ let's find the w_1, w_2 & b that give $\hat{y}$ with least error. using gradient descent.



⇒ To get how weight z (w_1) changes the Loss function we do derivative with chain rule → $\frac{\partial L(y, \hat{y})}{\partial w_1} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_1}$

Same for weight 2 (w_2) = $\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_2}$

& bias (b) = $\frac{\partial L}{\partial b} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial b}$

These derivatives tell's us how much the ^{individual} weights & biases effect the output or loss function.

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

$\hat{y} = \sigma(w_1 x_1 + w_2 x_2 + b)$ {Here the output is sigmoid function of the output of prediction function}

So finding the derivatives $\rightarrow$

$$\frac{\partial L}{\partial \hat{y}} = -\frac{(y - \hat{y})}{\hat{y}(1-\hat{y})}$$

$$\frac{\partial \hat{y}}{\partial w_1} = \hat{y}(1-\hat{y})x_1 \quad \left\{ \text{we know that } \frac{d}{dx} \sigma(x) = \sigma(x) \cdot (1-\sigma(x)) \right\}$$

$$\frac{\partial \hat{y}}{\partial w_2} = \hat{y}(1-\hat{y})x_2$$

$$\frac{\partial \hat{y}}{\partial b} = \hat{y}(1-\hat{y})$$

Now,
$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial b} \Rightarrow \frac{-(y - \hat{y})}{\hat{y}(1 - \hat{y})} \cdot \hat{y}(1 - \hat{y}) \Rightarrow -(y - \hat{y})$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_1} \Rightarrow \frac{-(y - \hat{y})}{\hat{y}(1 - \hat{y})} \cdot \hat{y}(1 - \hat{y}) x_1 \Rightarrow -(y - \hat{y}) x_1$$

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_2} \Rightarrow \frac{-(y - \hat{y})}{\hat{y}(1 - \hat{y})} \cdot \hat{y}(1 - \hat{y}) x_2 \Rightarrow -(y - \hat{y}) x_2$$

Now applying gradient descent.

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

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

$$\text{new}(b) \rightarrow b - \alpha \frac{-(y - \hat{y})}{\hat{y}(1 - \hat{y})} \frac{\partial L}{\partial b} \Rightarrow b - \alpha (-(y - \hat{y}))$$

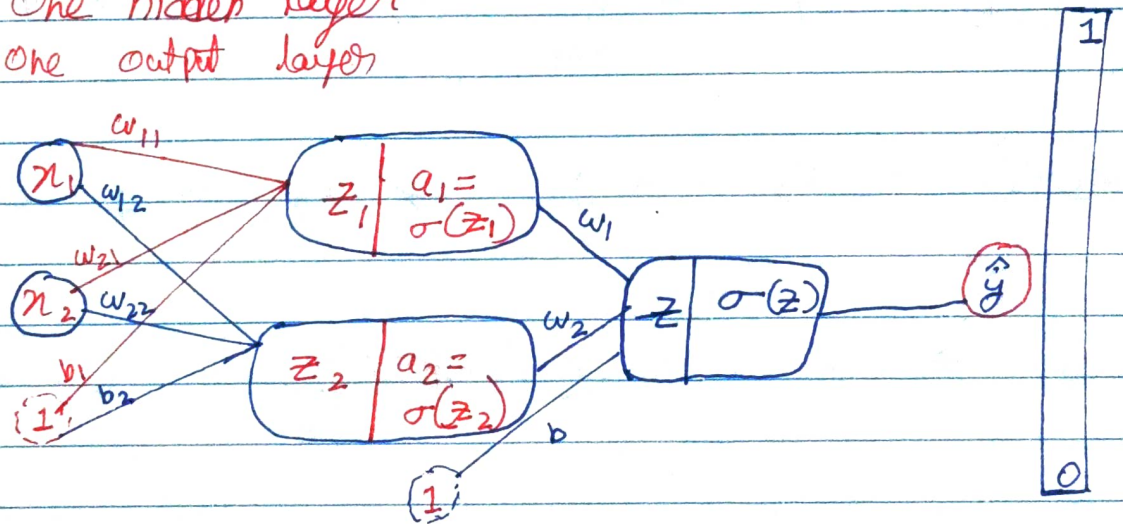
Classification with a neural network.

⇒ A Neural Network is a collection of perceptrons organised in layers.

⇒ We have to take the log loss derivative with respect to each one of the weight's & biases of the neural network.

Neural network of depth 2 given below

- One input layer
- One hidden layer
- One output layer



Here,

$$a_1 = \sigma(z_1)$$

$$z_1 = w_{11}x_1 + w_{21}x_2 + b_1$$

$$a_2 = \sigma(z_2)$$

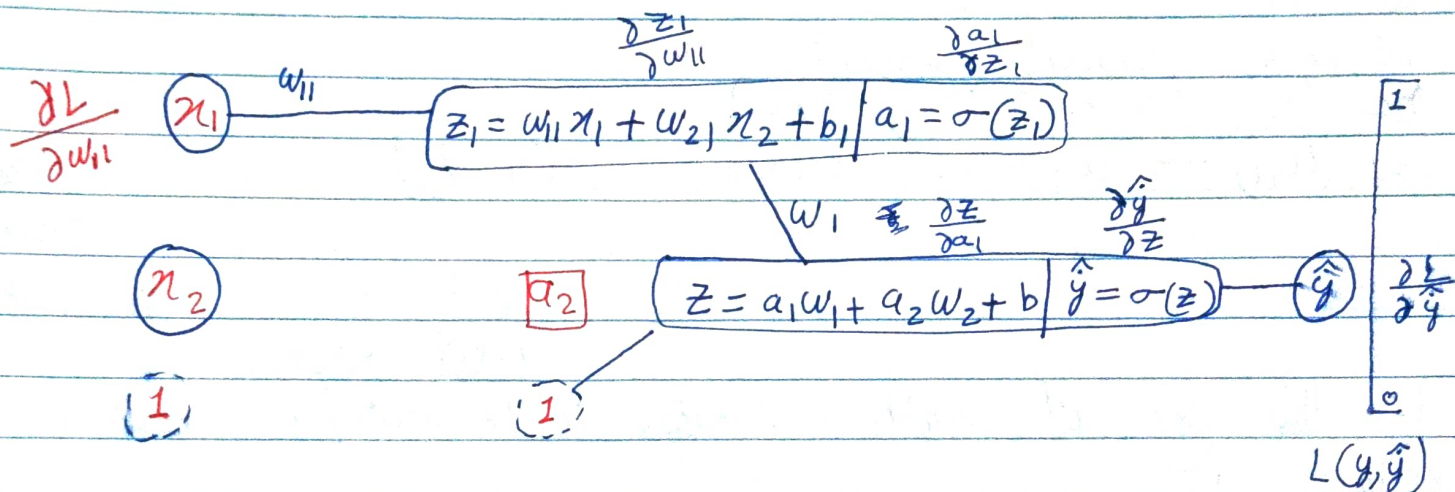
$$z_2 = w_{12}x_1 + w_{22}x_2 + b_2$$

$$\hat{y} = \sigma(z)$$

$$z = w_1a_1 + w_2a_2 + b$$

Classification with a neural network: minimizing log-loss

from the neural net's diagram on the left, ~~off~~ let's find how (w_{11}) effects the log loss $L(y, \hat{y})$.



Now, for $\frac{\partial L}{\partial w_{11}}$, we have to do a very big chain rule.

$$\frac{\partial L}{\partial w_{11}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial a_1} \cdot \frac{\partial a_1}{\partial z_1} \cdot \frac{\partial z_1}{\partial w_{11}}$$

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

$$\hat{y} = \sigma(z), \quad z = a_1w_1 + a_2w_2 + b, \quad a_1 = \sigma(z_1), \quad z_1 = x_1w_{11} + x_2w_{21} + b_1$$

Using the values let's find the derivative,

$$\frac{\partial L}{\partial \hat{y}} = \frac{-y}{\hat{y}} + \frac{1-y}{1-\hat{y}} = -\frac{(y-\hat{y})}{\hat{y}(1-\hat{y})}$$

$$\frac{\partial \hat{y}}{\partial z} = \sigma(z) \cdot (1 - \sigma(z)) \quad \hat{y} \cdot (1 - \hat{y})$$

$$\frac{\partial z}{\partial a_1} = w_1$$

$$\frac{\partial a_1}{\partial z_1} = \sigma(z_1) \cdot (1 - \sigma(z_1))$$

$$\frac{\partial z_1}{\partial w_{11}} = x_1$$

$$\hookrightarrow (a_1)(1-a_1)$$

Now, $\frac{\partial L}{\partial w_{11}} = \frac{-(y - \hat{y})}{\hat{y}(1 - \hat{y})} \cdot \hat{y}(1 - \hat{y}) \cdot w_1 \cdot a_1(1 - a_1) \cdot x_1$

$$= \boxed{-x_1 w_1 a_1 (1 - a_1) (y - \hat{y})}$$

Perform gradient descent with

~~new~~ $w_{11} \rightarrow w_{11} - \alpha (\partial L / \partial w_{11})$

new $w_{11} \rightarrow w_{11} - \alpha (-x_1 w_1 a_1 (1 - a_1) (y - \hat{y}))$, this will find the optimal value of w_{11} that gives the least error.

"Do this for all weights & biases in the neural network to optimize it"

⇒ "The way we adjust the weights & biases for a single gradient descent step also depends on every single example"

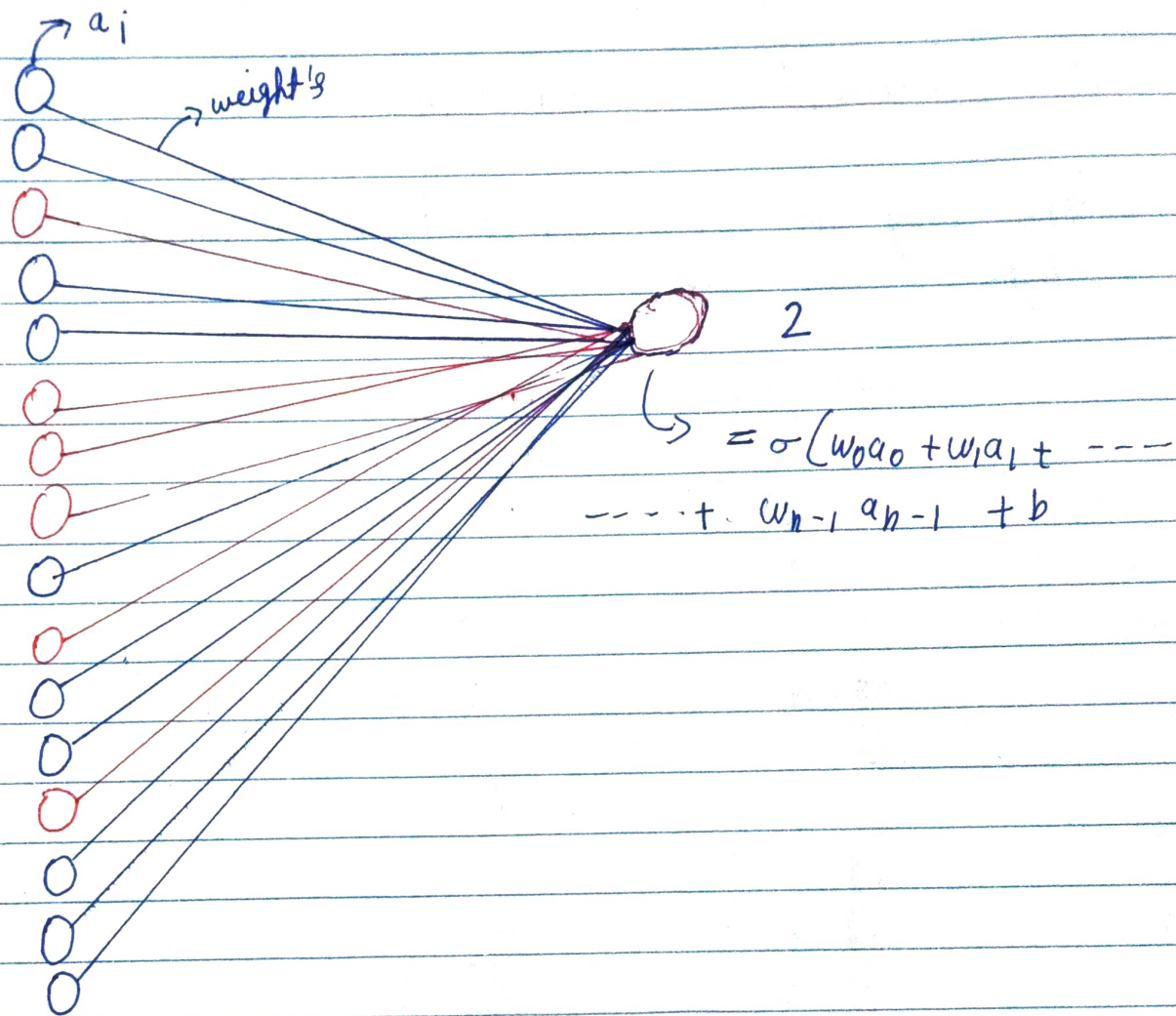
Back Propagation Introduction

Let's say we are going to train a neural network with a single image of 2,

output layer		expected output	
0.0	0	0.0	0
0.2	1	0.0	1
0.2	2	1.0	2
0.6	3	0.0	3
0.5	4	0.0	4
1.0	5	0.0	5
1.0	6	0.0	6
0.8	7	0.0	7
0.5	8	0.0	8
0.3	9	0.0	9

Here in the output layer we would like to nudge down all the values except the value of "2" we need to nudge up that value. The sizes of each nudge should be proportional to how far away each current value is from target value.

Now focusing on the neuron that activates 2.



⇒ Here to increase the value of neuron that activates 2 we can

- ⇒ Increase b
- ⇒ Increase w_i (in proportion to a_i)
- ⇒ Change a_i (in proportion to w_i)

⇒ if everything connected to the '2' neuron with a +ve weight got brighter & every thing connected with a -ve weight got dimmer, the value of digit 2 neuron will increase.

The paragraph above this is what that digit 2 neuron wants there are 9 other neurons in the output layer that want to change the

weights & biases, so in that second to last layer,
so,

so the values of nudging up & down of every neuron in the second last layer as per the requirements of the output layer gets added up.
(every neuron in the)

& you get a list of nudges that you want to happen to this second last layer,
& ~~recursively~~ recursively you can find the nudges to the third last layer & so on till you get to the input layer.

& we should also know that this is all how a single training example 2 wants to change the value of weights & biases;

if we do only this then the network will be incentivised to predict 2.

So what we do is record how each & every single training example wants to change the values of weights & biases

& we average together those desired changes.

& that's how you get the values of weights & biases which are negative gradient of the cost function.

& then you perform gradient descent

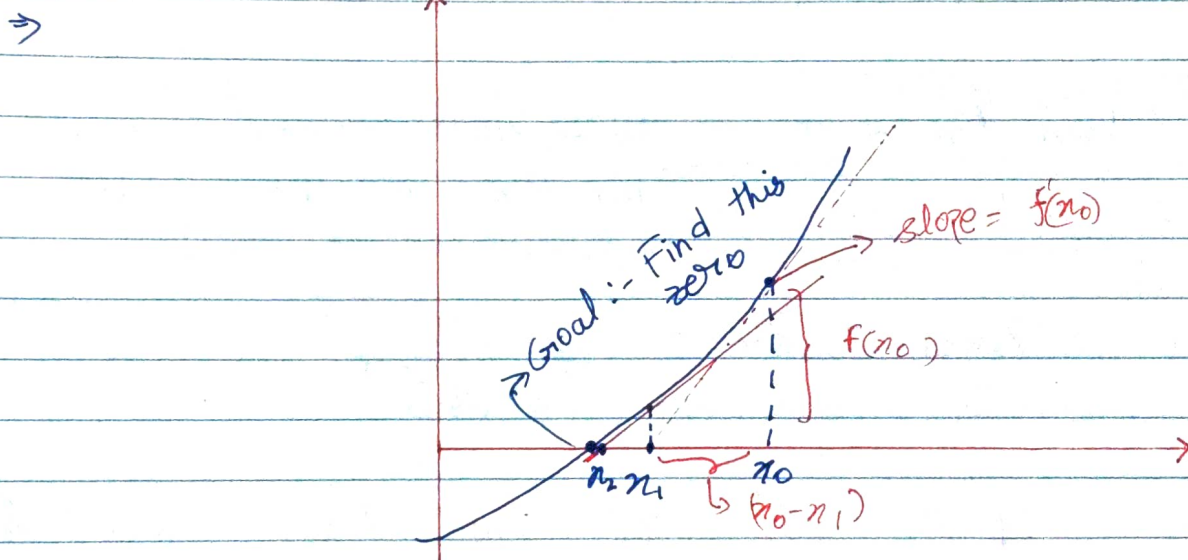
It takes computers large time to compute the influence of every training example every gradient descent step. So instead we do "STOCHASTIC GRADIENT DESCENT"

in stochastic gradient descent you randomly shuffle your training data & divide it into mini batches. let's say each one having a 100 training examples, we compute each step of gradient descent with respect to a mini batch.

Newton's Method

⇒ Newton's Method is used to find the zeros of a function. However, we can tweak it a bit to be used for optimization, it works in many variables as well.

Newton method for one variable.



for Newton's method let's start with some random points say (x_0) & x_0 is not the zero of a function.

& let's take a tangent at this point & see where this tangent hits the x-axis, & let's call that point x_1 , now do the same till you hit the zero or get pretty close to it.

Now, $\text{slope} = \frac{\text{rise}}{\text{run}} \Rightarrow$

$$f'(x_0) = \frac{f(x_0)}{(x_0 - x_1)}, \text{ so}$$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Now to generalize,
we can say

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

Newton's method for optimization.

Goal :- minimize $g(x)$ $\rightarrow$ find zeros of $g'(x)$

Steps

$f(x) \mapsto g'(x)$	$f'(x) \mapsto (g'(x))'$
above statement says that let's say $f(x)$ is single differentiation of $g(x)$	then $f'(x)$ will be double differentiation of $g(x)$

① Start with some x_0

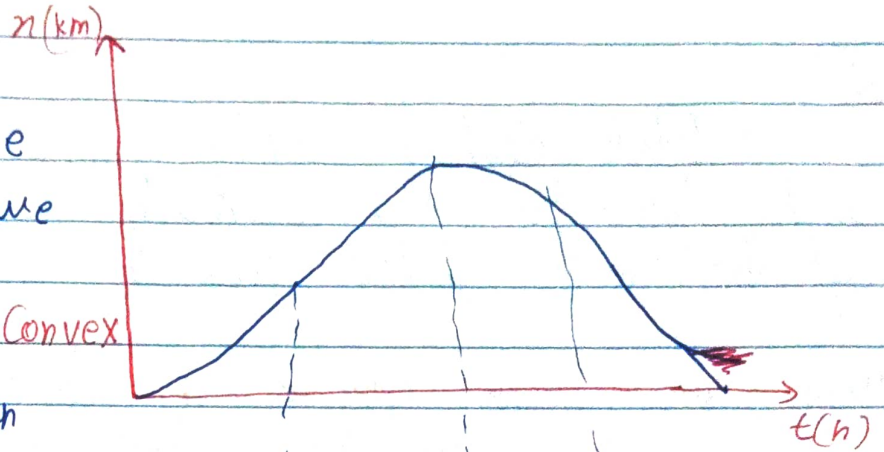
② Update

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

③ Repeat step ② until you find the candidate for minimum.

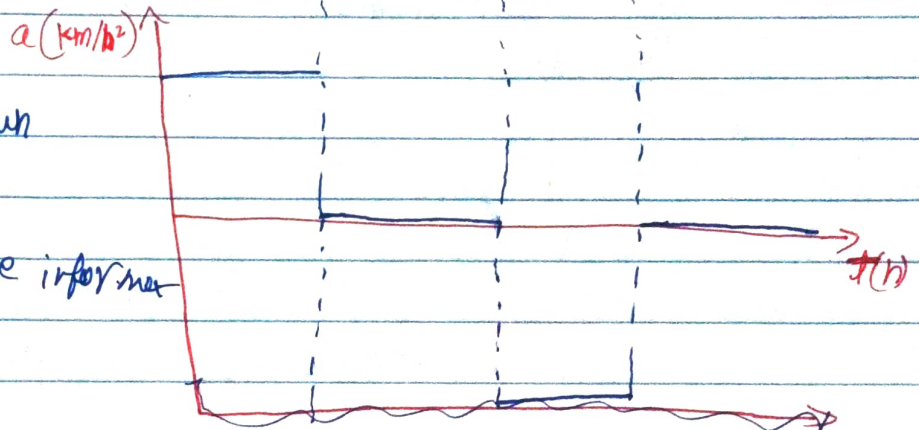
Curvature

⇒ When you have a true second derivative, you have a concave up function
 $\frac{d^2x}{dt^2} > 0$, Concave up or Convex
 & the fn is increasing at an increasing rate.

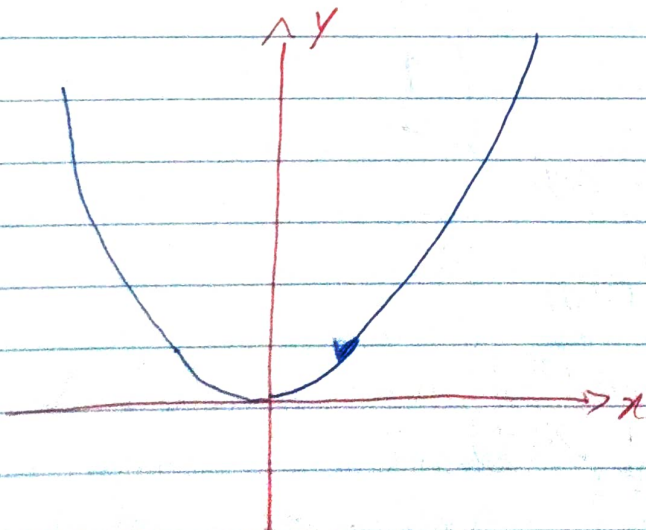


$$\frac{d^2x}{dt^2} < 0 \text{ concave down}$$

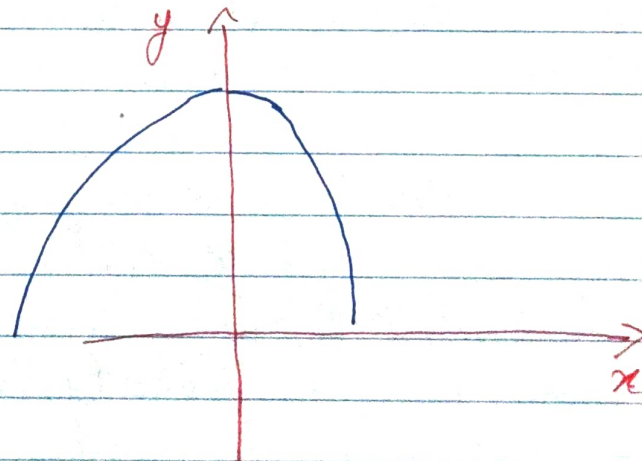
$$\frac{d^2x}{dt^2} = 0, \text{ Need more information.}$$



The second derivative gives a measure of the amount by which a curve deviates from being a straight line. This is called curvature.



Concave up or Convex
 $f''(0) > 0$



Concave down
 $f''(0) < 0$

when,

$$f''(0) > 0 \text{ Local minimum}$$

$$f''(0) < 0 \text{ Local maximum}$$

Hessian

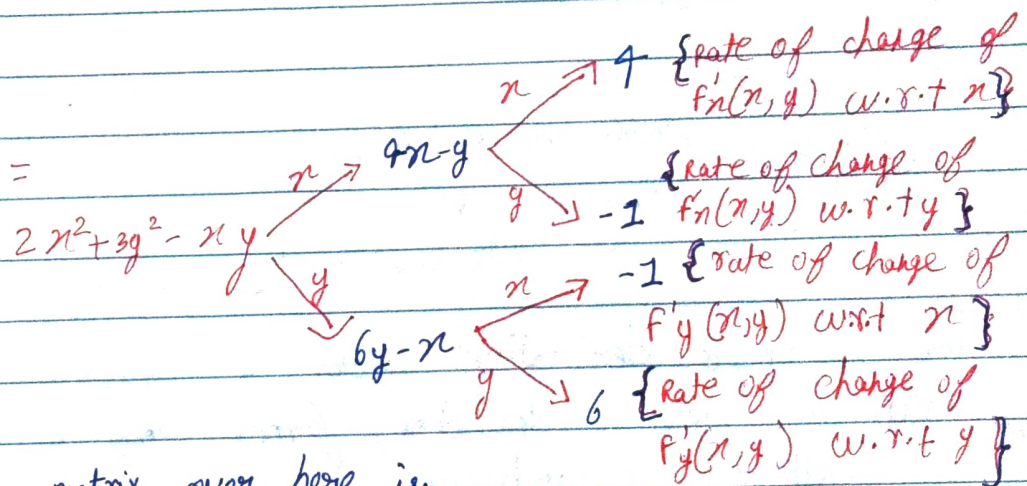
⇒ While dealing with multiple variable the second derivative is actually a matrix full of second derivatives called hessian.

When we wanna optimize a function of many variable we can use multivariable newton's method. & that one uses the Hessian.

Second derivative

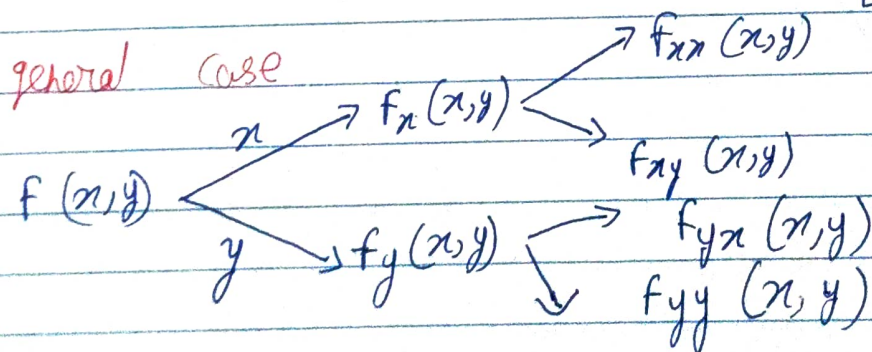
$$F(x, y) =$$

$$2x^2 + 3y^2 - xy$$



so, the hessian matrix over here is

$$\begin{bmatrix} 4 & -1 \\ -1 & 6 \end{bmatrix}$$



$$(\text{Hessian matrix}) \quad H = \begin{bmatrix} f''_{xx}(x, y) & f''_{xy}(x, y) \\ f''_{yx}(x, y) & f''_{yy}(x, y) \end{bmatrix}$$

Hessians & Concavity

For concave up $\rightarrow f(x,y) = 2x^2 + 3y^2 - xy$

$$H(0,0) = \begin{bmatrix} 4 & -1 \\ -1 & 6 \end{bmatrix}$$

In one variable, if we wanted to know if it was concave up we looked at the second derivative & checked if it was +ve, but now we have a matrix, so, How do we check if a matrix is +ve or -ve? well you actually look at the eigenvalues,

So the eigen values of the matrix given above are,

$$\det(H(0,0) - \lambda I) = (4-\lambda)(6-\lambda) - (-1)(-1) \\ \Rightarrow \lambda^2 - 10\lambda + 23$$

$$\text{So, } \lambda_1 = 6.41 \\ \lambda_2 = 3.59$$

Now, both the eigen vector's are +ve. So for a matrix this is equivalent of being a +ve number, it means the matrix is +ve definite, because all the eigen values of this matrix are +ve then the function is concave up & Point 0,0 is minimum.

for Concave Down

$$f(x, y) = -2x^2 + (-3y^2) - xy + 15$$

$$\nabla f(x, y) = \begin{bmatrix} -4x & -y \\ -x & -6y \end{bmatrix}$$

$$H(0, 0) = \begin{bmatrix} -4 & -1 \\ -1 & -6 \end{bmatrix}$$

$$\det(H(0, 0) - \lambda I) = (-4 - \lambda)(-6 - \lambda) - (-1)(-1)$$

$$\Rightarrow \lambda^2 + 10\lambda + 23$$

$$\boxed{\lambda_1 = -3.59, \lambda_2 = -6.41}$$

All The eigen values are -ve ~~so~~, the matrix is negative & then it's concave down & therefore the point 0.00 is a maximum.

Saddle point

$$f(x, y) = 2x^2 - 2y^2$$

$$\nabla f(x, y) = \begin{bmatrix} 4x \\ -4y \end{bmatrix}$$

$$H(0, 0) = \begin{bmatrix} 4 & 0 \\ 0 & -4 \end{bmatrix}$$

$$\det(H(0, 0) - \lambda I) = (4 - \lambda)(-4 - \lambda) - 0$$

$$\lambda_1 = -4$$

$$\lambda_2 = 4$$

Here, not all the eigenvalues are positive & not all the eigenvalues are negative,

So we can't conclude anything, so here $(0,0)$ is actually a saddle point a point that is neither minimum nor maximum

for more than two variables

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

for minima All $(\lambda_i > 0)$

for maxima All $(\lambda_i < 0)$

& when some $(\lambda_i > 0)$ & some $(\lambda_j < 0)$
or At least one $(\lambda_i = 0)$, then it's inconclusive.

⇒ Newton's method for two variables Optimization

1 variable

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

$$x_{k+1} = x_k - \left(f''(x_k)^{-1} \cdot f'(x_k) \right)$$

2 variables

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - H^{-1}(0,0) \cdot \nabla f(x_k, y_k)$$

⇒ The double derivative turns into hessian & Single derivative turns into the gradient.

An example -:

lets find the minimum of a concave up function given by eqn, $\Rightarrow$

$$F(x, y) = x^4 + 0.8y^4 + 4x^2 + 2y^2 - xy - 0.2x^3y$$

$$f(x, y) \begin{cases} x \rightarrow 4x^3 + 8x - y - 0.4xy & \begin{cases} x \rightarrow 12x^2 + 8 - 0.4y \\ y \rightarrow -1 - 0.4x \end{cases} \\ y \rightarrow 3.2y^3 + 4y - x - 0.2x^2 & \begin{cases} x \rightarrow -1 - 0.4x \\ y \rightarrow 9.6y^2 + 4 \end{cases} \end{cases}$$

$$\nabla f(x, y) = \begin{bmatrix} 4x^3 + 8x - y - 0.4xy \\ 3.2y^3 + 4y - x - 0.2x^2 \end{bmatrix}$$

$$H(x, y) = \begin{bmatrix} 12x^2 + 8 - 0.4y & -1 - 0.4x \\ -1 - 0.4x & 9.6y^2 + 4 \end{bmatrix}$$

lets choose a ^{random} start point $\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$

$$\nabla f(4, 4) = \begin{bmatrix} 277.6 \\ 213.6 \end{bmatrix}$$

$$H(4, 4) = \begin{bmatrix} 198.4 & -2.6 \\ -2.6 & 157.6 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \end{bmatrix} - \begin{bmatrix} 198.4 & -2.6 \\ -2.6 & 157.6 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 277.6 \\ 213.6 \end{bmatrix}$$

$$= \begin{bmatrix} 2.58 \\ 2.62 \end{bmatrix} \rightarrow \text{This is a new point}$$

iterate this process again & again

Here we needed $k = 8$ steps, or 8 iterations.

$$\begin{bmatrix} x_8 \\ y_8 \end{bmatrix} = \begin{bmatrix} 4.15 \times 10^{-17} \\ -2.05 \times 10^{-17} \end{bmatrix}$$

The above points are super close to Zero.

so, the optimal point for x & y are

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$