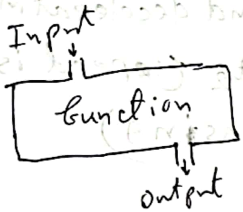
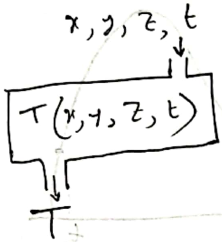
Multivariate Calculus

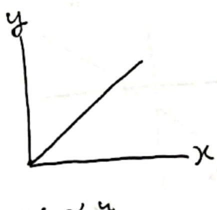
Functions



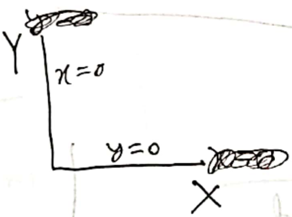


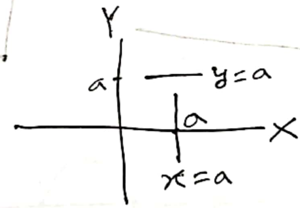
temperature at a certain co-ordinate

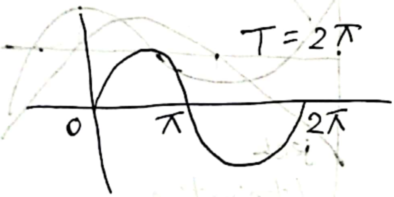
Leibniz		Lagrange		Newton		Euler
$\frac{dy}{dx}$	$\equiv$	$y'(x)$	$\equiv$	$\dot{y}$	$\equiv$	Dy



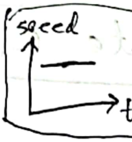
$x \propto y$



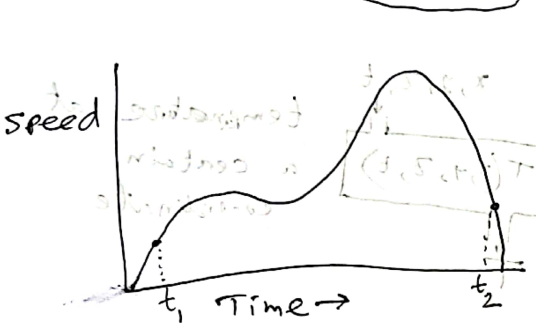




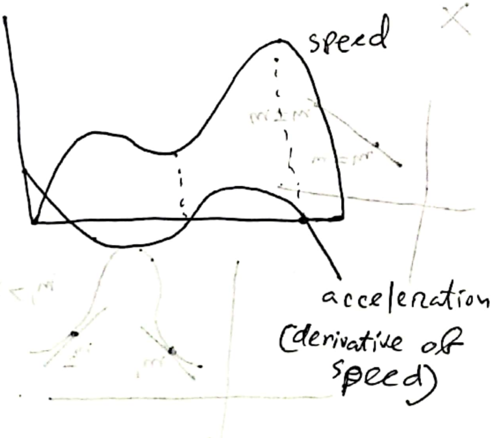
Rise over Run

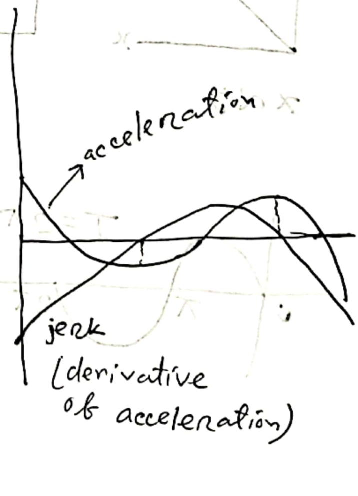


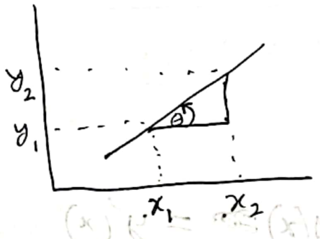
the car is moving at constant speed



the car is accelerating (speed is increasing) at $t = t_1$ and decelerating at $t = t_2$ (speed is decreasing)

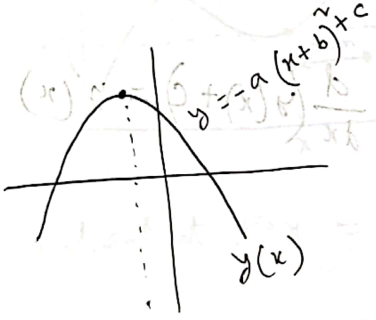


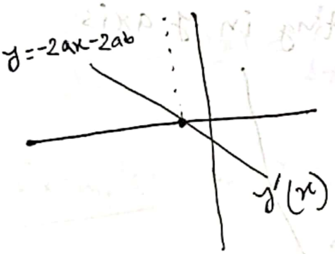


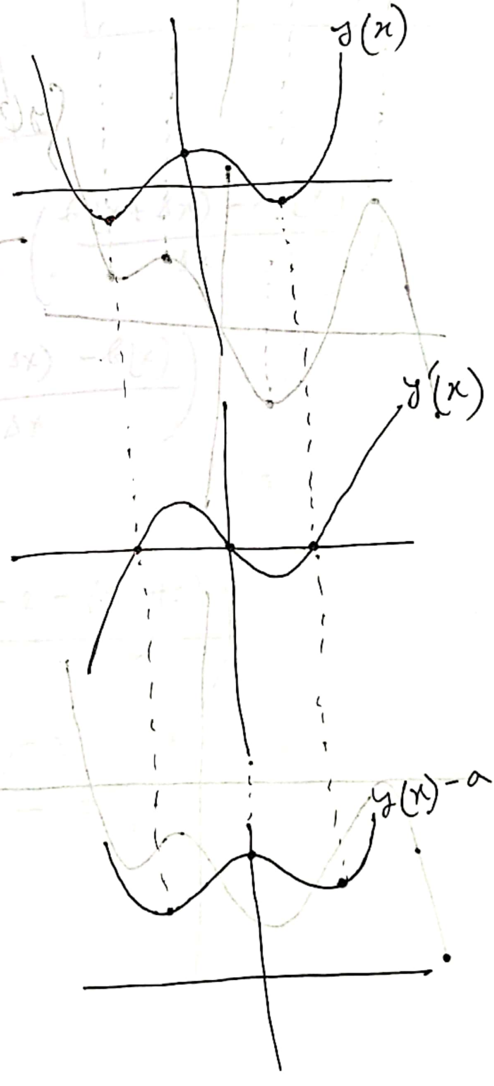


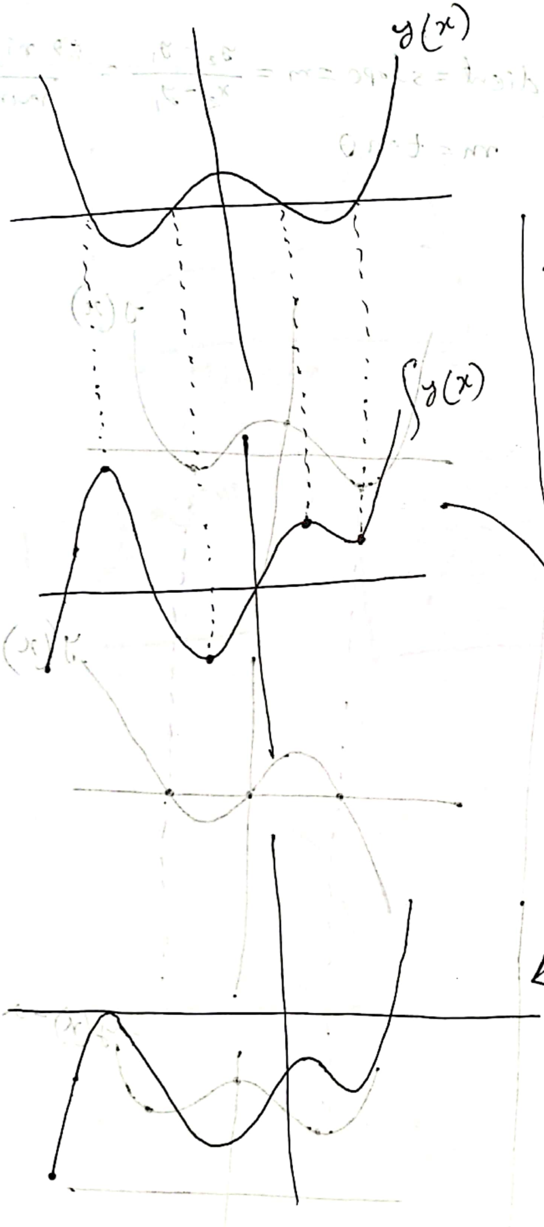
$$\text{gradient} = \text{slope} = m = \frac{y_2 - y_1}{x_2 - y_1} = \frac{\text{rise}}{\text{run}}$$

$$m = \tan\theta$$









$$\frac{d}{dx} y(x) = y'(x)$$

$$\frac{d}{dx}\left(y(x)+c\right) = y'(x)$$

shifting in y axis

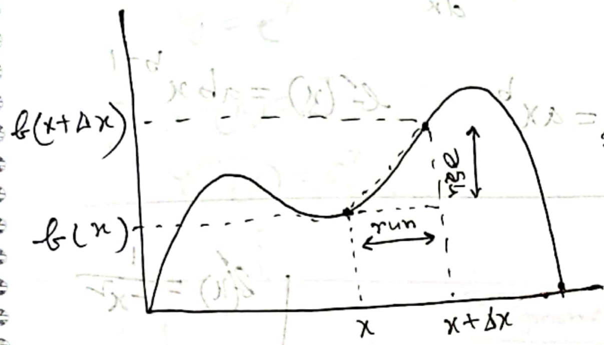
Definition of Derivative

$f(x)$ is a cont^s function

$$\text{gradient} = \frac{\text{rise}}{\text{run}}$$

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

$$\boxed{\frac{f(x+\Delta x) - f(x)}{\Delta x}}$$



$$\text{gradient at } x = \lim_{\Delta x \to 0}\left(\frac{f(x+\Delta x) - f(x)}{\Delta x}\right)$$

$$\frac{df}{dx} = f'(x) = \lim_{\Delta x \to 0}\left(\frac{f(x+\Delta x) - f(x)}{\Delta x}\right)$$

<u>Example:</u> $f(x) = 3x + 2$

$$f'(x) = \lim_{\Delta x \to 0}\left(\frac{3(x+\Delta x) + 2 - (3x+2)}{\Delta x}\right)$$

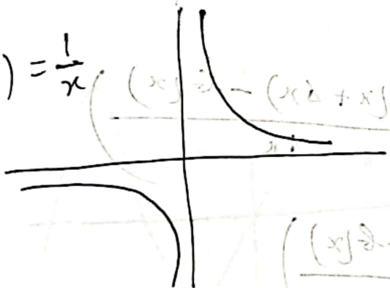
$$= \lim_{\Delta x \to 0}\left(\frac{\cancel{3x} + 3\Delta x \cancel{+2} \cancel{-3x} \cancel{-2}}{\Delta x}\right) = \lim_{\Delta x \to 0}\left(3\frac{\cancel{\Delta x}}{\cancel{\Delta x}}\right) = 3$$

$\boxed{\text{Sum rule}}$

$$\frac{d}{dx}\left(f(x)+g(x)\right) = \frac{df(x)}{dx} + \frac{dg(x)}{dx}$$

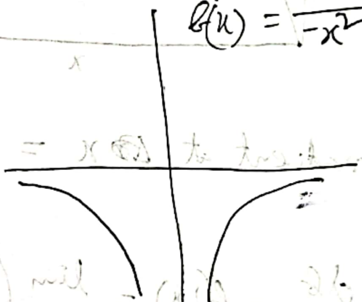
$\boxed{\text{Power rule}}$ $\quad f(x) = ax^b \qquad f'(x) = abx^{b-1}$

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



$\xrightarrow{\frac{d}{dx}}$

$f'(x) = \frac{1}{-x^2}$



undefined at $x=0$

negative slope at all points

undefined at $x=0$

gradient is negative

---

now we want a function that's derivative is always ~~zero~~ the same.

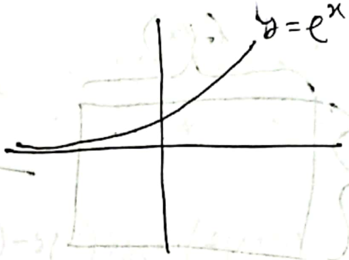
a function that satisfies our demand is $x=0$

another function that satisfies this property is

$$y = e^x$$

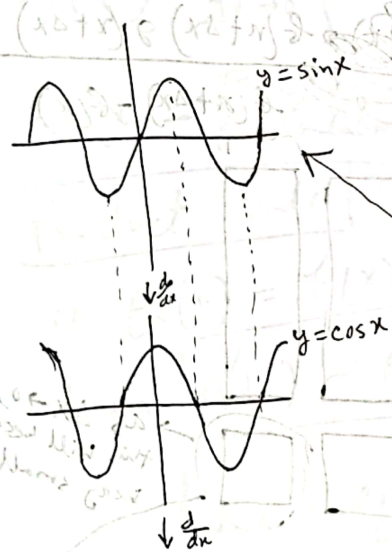
$$y''(x) = e^x$$

$$y'''(x) = e^x$$



e = Euler's number $= 2.71828\ldots$

$\boxed{\text{Trigonometric functions}}$



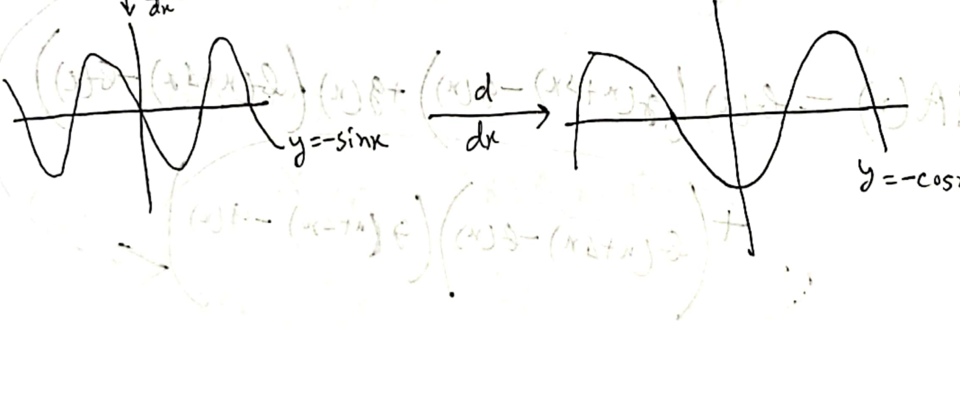
$f(x) = \sin(x)$

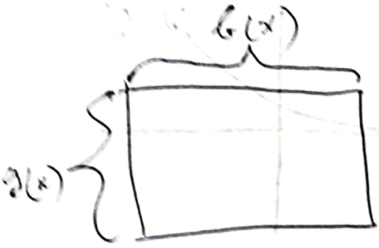
$f''''(x) = \sin x$

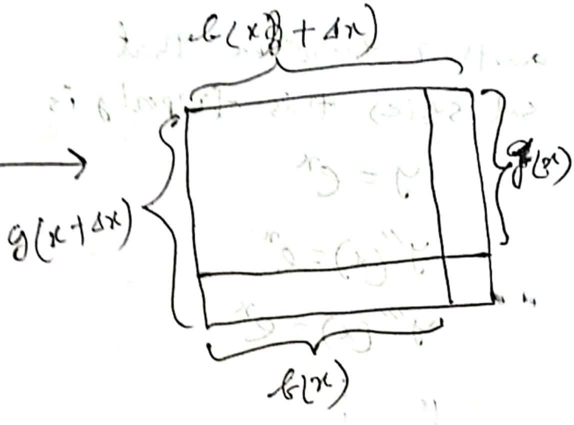
because $\sin x$ is e^x in disguise

$$\sin x = \frac{e^{ix} - e^{-ix}}{2i}$$

$\frac{d}{dx}$



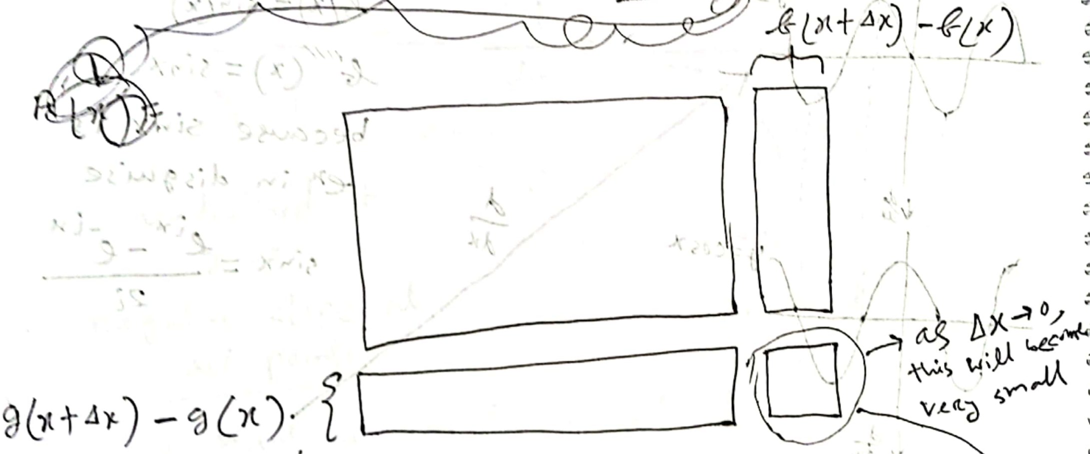




$A(x) = f(x)\, g(x)$

area

new area $\rightarrow$ $A(x) = f(x+\Delta x)\, g(x+\Delta x)$



$$\Delta A(x) = f(x)\big(g(x+\Delta x) - g(x)\big) + g(x)\big(f(x+\Delta x) - f(x)\big)$$

$$+ \big(f(x+\Delta x) - f(x)\big)\big(g(x+\Delta x) - g(x)\big)$$

$$\lim_{\Delta x \to 0} \left(\Delta A(x)\right) = \lim_{\Delta x \to 0} \left(f(x)\left(g(x+\Delta x) - g(x)\right) + g(x)\left(f(x+\Delta x) - f(x)\right)\right)$$

$$\Rightarrow \lim_{\Delta x \to 0} \frac{\Delta A(x)}{\Delta x} = \lim_{\Delta x \to 0} \frac{f(x)\left(g(x+\Delta x) - g(x)\right) + g(x)\left(f(x+\Delta x) - f(x)\right)}{\Delta x}$$

$$\Rightarrow A'(x) = \lim_{\Delta x \to 0} \left(f(x) \frac{\left(g(x+\Delta x) - g(x)\right)}{\Delta x} + g(x) \frac{f(x+\Delta x) - f(x)}{\Delta x} \right)$$

$$\Rightarrow A'(x) = f(x)\, g'(x) + g(x)\, f'(x)$$

Product rule

if $A(x) = f(x)\, g(x)$

then $\boxed{A'(x) = f(x)\, g'(x) + g(x)\, f'(x)}$

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

$$u(x) = f(x)\, g(x)\, h(x)$$

$$u'(x) = \left(f(x)\, g(x)\right)' h(x) + h'(x)\, f(x)\, g(x)$$

$$= f'(x) g(x) h(x) + f(x)\, g'(x)\, h(x) + f(x) g(x) h'(x)$$

$\boxed{\text{chain rule}}$

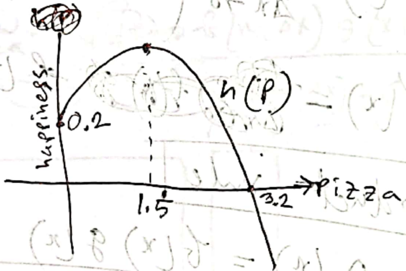
let's consider a nested function $h(p(m))$.

$h = h(p) \rightarrow$ describes how happy I am

$p = p(m) \rightarrow$ how many pizzas I eat

$m \rightarrow$ how much money I make

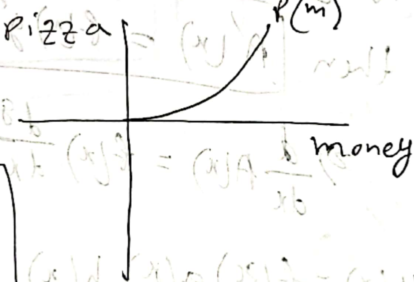
$$h(p) = -\frac{1}{3}p^2 + p + \frac{1}{5}$$



$$p(m) = e^m - 1$$

$$h(p(m)) = -\frac{1}{3}(e^m-1)^2 + (e^m-1) + \frac{1}{5}$$

$$\frac{dh}{dm} = \frac{1}{3}e^m(5-2e^m)$$



$$\boxed{\frac{dh}{dp} \times \frac{dp}{dm} = \frac{dh}{dm}}$$

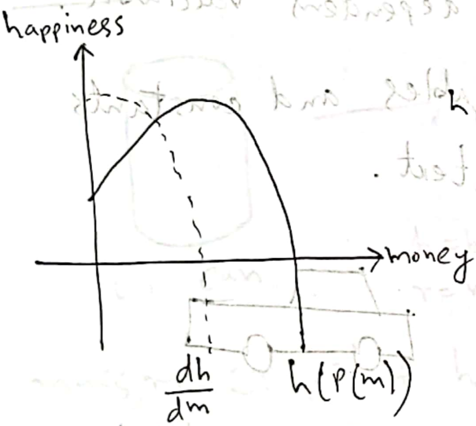
if $h = h(p)$ and $p = p(m)$

$$\frac{dh}{dp} = 1 - \frac{2}{3}p \qquad \frac{dp}{dm} = e^m$$

$$\frac{dh}{dp} \times \frac{dp}{dm} = \left(1 - \frac{2}{3}p\right)e^m$$

$$= \left(1 - \frac{2}{3}(e^m - 1)\right)e^m$$

$$= \frac{1}{3}e^m\left(5 - 2e^m\right) = \frac{dh}{dm}$$



$$h(p(m)) = e^m - \frac{1}{3}(e^m - 1) - \frac{4}{5}$$

$$\frac{dh}{dm} = \frac{1}{3}e^m\left(5 - 2e^m\right)$$

---

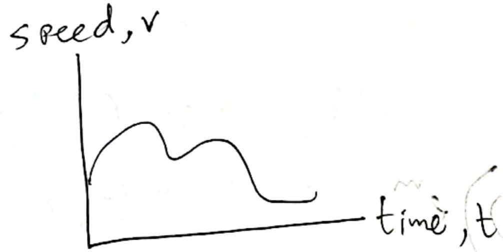
if $f(x) = g(h(x))$

then $f'(x) = g'(h(x))\,h'(x)$

$$\frac{df(x)}{dx} = \frac{d}{dx}g(h(x))$$

$$= g'(h(x)) \cdot h'(x)$$

Variables, constants | content



$v = v(t)$

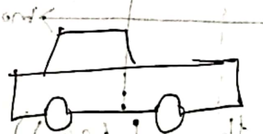
v: dependent variable

t: independent variable

the speed of a car changes ~~~~ at different points in times, so v is dependent variable.

but the choice of variables and constants might depend on the context.

F = force generated by a car's engine



m = mass of car

$$F = ma + dv^2$$

a = acceleration

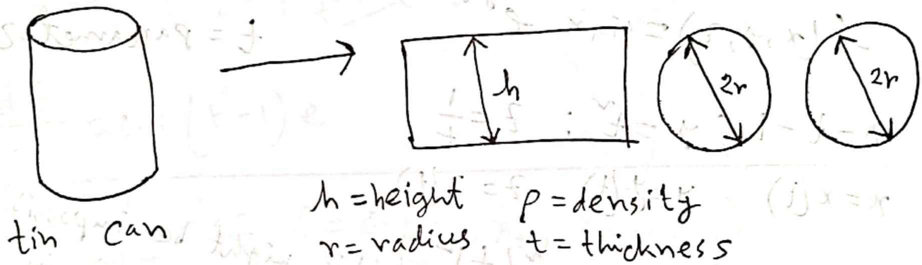
d = aerodynamic drive

v = velocity

for a driver, his car is already built, so m, d are constants.

but for the car designer, we might have to alter m, d to achieve a target a, v. so for him, m, d are variables.

<u>another example</u>



h = height　　ρ = density
r = radius　　t = thickness

mass, $m = 2\pi r^2 \rho t + h \cdot 2\pi r \rho t$

$\frac{\partial m}{\partial h} = 2\pi r t \rho$ (mass varies linearly with height when all else is kept constant)

$\frac{\partial m}{\partial r} = 4\pi r t \rho + 2\pi h t \rho$

$\frac{\partial m}{\partial t} = 2\pi r^2 \rho + 2\pi r h \rho$

$\frac{\partial m}{\partial \rho} = 2\pi r^2 t + 2\pi r h t$

$$f(x,y,z) = \sin x \, e^{yz^2}$$

$$\frac{\partial f}{dx} = \cos x \, e^{yz^2}$$

$$\frac{\partial f}{dy} = \sin x \, e^{yz^2} z^2$$

$$\frac{\partial f}{dz} = \sin x \, e^{yz^2} \, 2yz$$

$$f(x,y,z) = \sin x \, e^{yz^2}$$

t = parameter

$$x = t-1 \; ; \; y = t^2 \; ; \; z = \frac{1}{t}$$

$$x = x(t) \quad y = y(t) \quad z = z(t)$$

$$f(t) = \sin(t-1) \, e^{t^2 \left(\frac{1}{t}\right)^2}$$

$$f(t) = \sin(t-1) \, e$$

$$\frac{df(t)}{dt} = \cos(t-1) \, e$$

this might be impossible/ too difficult in some cases, so we use the chain rule

$$\frac{df(x,y,z)}{dt} = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{dy}\frac{dy}{dt} + \frac{\partial f}{\partial z}\frac{dz}{dt}$$

$$\frac{\partial f}{\partial x} = \cos x \; e^{yz^2}. \qquad \frac{\partial f}{\partial y} = z^2 \sin x \; e^{yz^2} \qquad \frac{\partial f}{\partial z} = 2yz \sin x \; e^{yz^2}$$

$$\frac{dx}{dt} = 1 \qquad \frac{dy}{dt} = 2t \qquad \frac{dz}{dt} = -t^{-2}$$

$$\frac{df(x,y,z)}{dt} = \cos x e^{yz^2} + z^2 \sin x e^{yz^2} \cdot 2t - 2yz \sin x e^{yz^2} t^{-2}$$

putting $x = t-1$, $y = t^2$, $z = \frac{1}{t}$,

$$\Rightarrow \frac{df|_{x,y,z}}{dt} = \left(\cos(t-1) e + \cancel{\frac{1}{t^2} \sin(t-1) e \cdot 2t} \; \cancel{- 2t^{-1} \sin(t-1) e} \right)$$

$$\Rightarrow \frac{df}{dt} = \cos(t-1)\, e$$

$\boxed{\text{Jacobian}}$ if $f = f(x_1, x_2, x_3 \ldots)$

the $J = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3}, \ldots \right]$ ← row vector

$$f(x,y,z) = x^2 y + 3z$$

$$\frac{\partial f}{\partial x} = 2xy$$

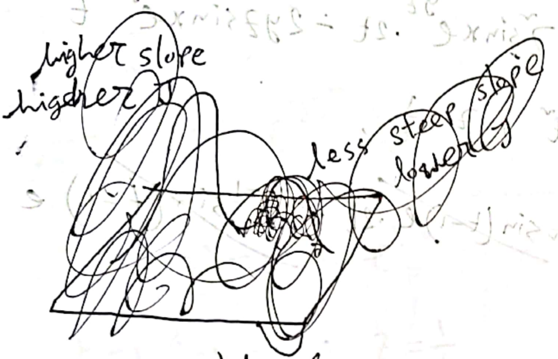
$$\frac{\partial f}{\partial y} = x^2 \qquad J = \left[2xy, x^2, 3 \right]$$

$$\frac{\partial f}{\partial z} = 3$$

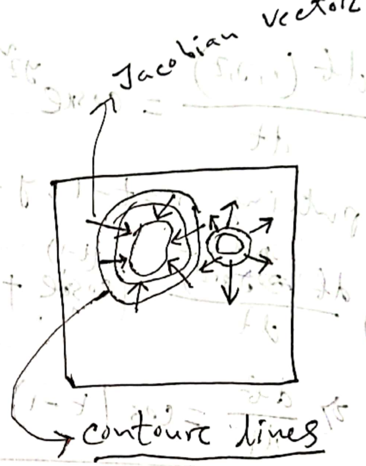
J gives steepest slopes of f for a given x, y, z

$$J = \left[2xy,\ x^2,\ 3 \right]$$

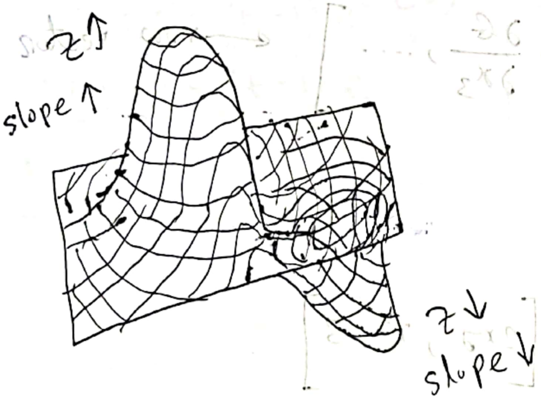
$$J(0,0,0) = \left[0,\ 0,\ 3 \right]$$



3D plot of arbitrary function



the jacobian vectors are pointing inwards to higher values and outwards for lower values of z.



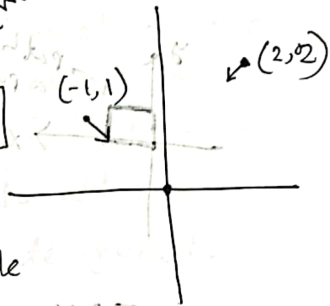
Jacobians applied

$$f(x,y) = e^{-(x^2+y^2)}$$

$$J = \left[-2x e^{-(x^2+y^2)}, -2y e^{-(x^2+y^2)} \right]$$

higher magnitude

$$J(-1,1) = \left[2e^{-2}, -2e^{-2} \right] = [0.27, -0.27]$$

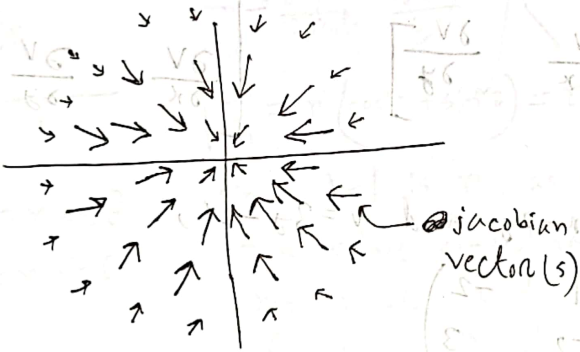


$$J(2,2) = [-0.001, -0.001]$$

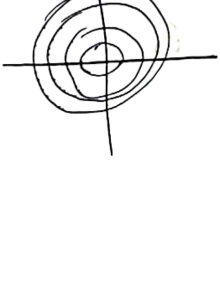
lower magnitude

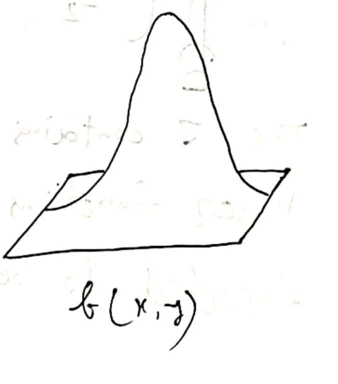
$$J(0,0) = [0,0]$$

maxima / minima / saddle point



so the origin must be maxm

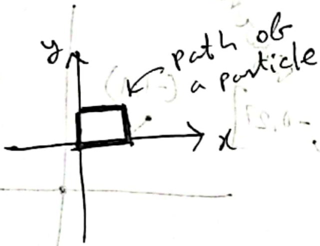


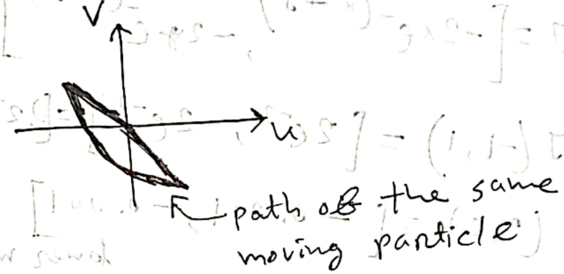


$f(x,y)$

$u(x,y) = x - 2y$

$v(x,y) = 3y - 2x$





$$J_u = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \end{bmatrix}$$

$$J_v = \begin{bmatrix} \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix}$$

$$J = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix}$$

so for $u = x - 2y$ and $v = 3y - 2x$,

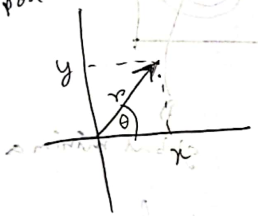
$$J = \begin{pmatrix} 1 & -2 \\ -2 & 3 \end{pmatrix}$$

the J contains no x or y because u, v are linear functions and the gradient is expected to be constant.

$$\begin{pmatrix} 1 & -2 \\ -2 & 3 \end{pmatrix}\begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} -4 \\ 5 \end{bmatrix}$$

this example is fairly simple since u, v are linear. but when u, v are non-linear functions, the jacobian gets complicated. But we can zoom in on the system and consider the small bits as linear functions and zoom out to add all the contributions to get an approximate result.

polar coordinates



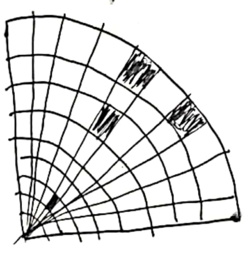
$$J = \begin{pmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{pmatrix}$$

$$|J| = r\left(\cos^2\theta + \sin^2\theta\right) = r$$

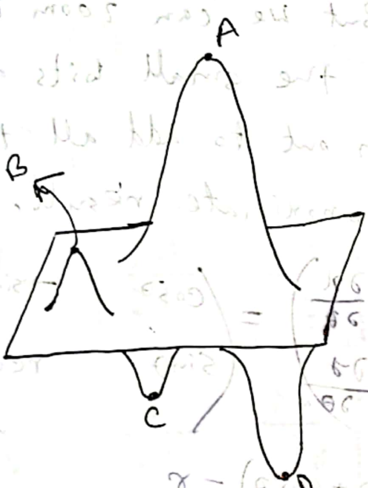
$x(r, \theta) = r\cos\theta$

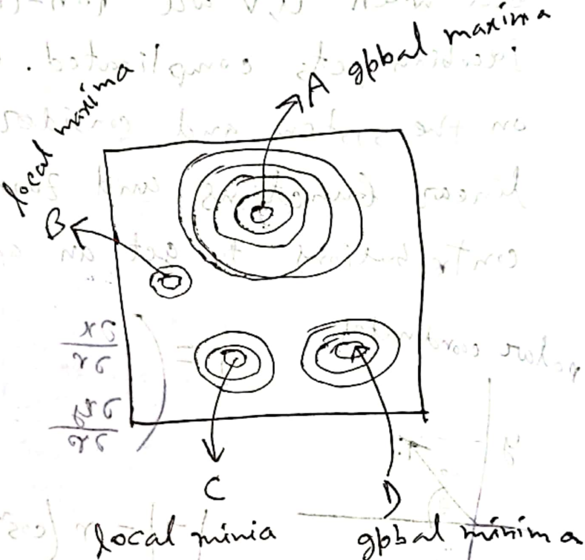
$y(r, \theta) = r\sin\theta$

since $|J|$ is r, as we move along the polar co-ordinates, small regions will behave like wrt function of r.



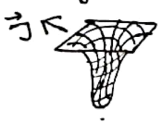
Jacobian of a scalar valued multivariate function gives a row vector pointing up the direction of greatest slope.





~~gradient~~ jacobian's length is proportional to the local steepness.

jacobian points to the greater slope and if slope is negative, jacobian points outwards



$\boxed{\text{Hessian}}$

n = number of variables in f

$$H = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 x_n} \\ \frac{\partial^2 f}{\partial x_2 x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 x_n} \\ \vdots & & & \vdots \\ \frac{\partial^2 f}{\partial x_n x_1} & \frac{\partial^2 f}{\partial x_n x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

$$\Rightarrow H = \begin{pmatrix} \partial_{x_1 x_2} f & \partial_{x_1 x_2} f & \cdots & \partial_{x_1 x_n} f \\ \partial_{x_2 x_1} f & \partial_{x_2 x_2} f & \cdots & \partial_{x_2 x_n} f \\ \vdots & & & \vdots \\ \partial_{x_n x_1} f & \partial_{x_n x_2} f & \cdots & \partial_{x_n x_n} f \end{pmatrix}$$

$\boxed{\text{Example}}$

$$J = [2xyz,\ x^2 z,\ x^2 y]$$

$$H = \begin{pmatrix} \frac{\partial\, 2xyz}{\partial x} & \frac{\partial\, 2xyz}{\partial y} & \frac{\partial\, 2xyz}{\partial z} \\ \frac{\partial x^2 z}{\partial x} & \frac{\partial x^2 z}{\partial y} & \frac{\partial x^2 y}{\partial z} \\ \frac{\partial x^2 y}{\partial x} & \frac{\partial x^2 y}{\partial y} & \frac{\partial x^2 y}{\partial z} \end{pmatrix} = \begin{pmatrix} 2yz & 2xz & 2xy \\ 2xz & 0 & x^2 \\ 2xy & x^2 & 0 \end{pmatrix}$$

notice that the Hessian is symmetric through the main diagonal. (for continuous functions).

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

(see plot on geogebra)

$J = [2x, 2y]$

$$H = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

$|H| = 4$

since $|H|$ is (+)ve, we know it's either a maximum or a minimum.

since H_{11} is (+)ve, it's a minimum. (see 3D plot)

Point with zero gradient

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

$J = [2x, -2y]$

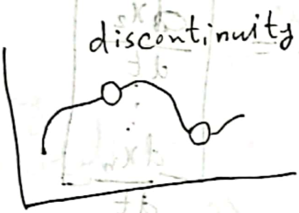
$$H = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix} \quad |H| = -4$$

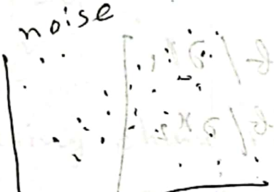
since $|H|$ is (-)ve, it is a saddle point. (see 3D plot)

But real world functions are not continuous. they have noise, discontinuity. many real world functions don't have an easy to express expression.

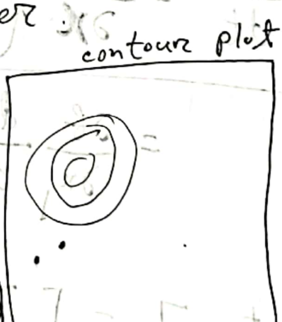
$$f(x) = \frac{1}{e} x^{-2^{x}} - 5xe^{2\ln x} + \operatorname{arc\,cot}(\log_9 x) + \ldots ?!$$

We can't differentiate a function and calculate J if we don't have a complete expression.





to overcome these real world problems' issues, we have developed numerical methods which give us an approximate answer :



although we can't get a function for this plot, but we can approximate jacobians for given points.

$$J = \left[\frac{f(x+\Delta x, y) - f(x,y)}{\Delta x}, \frac{f(x, y+\Delta x) - f(x,y)}{\Delta y} \right]$$

↳ but we can't take too small steps or too large steps

~~$f(x_1, x_2, \ldots$~~

Multivariate chain rule

$$f(x_1, x_2, \ldots x_n) = f(\mathbf{x})$$

$x_1(t) = \ldots \quad x_2(t) = \ldots \quad x_n(t) = \ldots$

$$\frac{df}{dt} = ?$$

vector

vector

$$\frac{\partial f}{\partial \mathbf{x}} = \begin{bmatrix} \partial f / \partial x_1 \\ \partial f / \partial x_2 \\ \vdots \\ \partial f / \partial x_n \end{bmatrix} \qquad \frac{d\mathbf{x}}{dt} = \begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \vdots \\ \frac{dx_n}{dt} \end{bmatrix}$$

$$\boxed{\frac{df}{dt} = \frac{\partial f}{\partial \mathbf{x}} \cdot \frac{d\mathbf{x}}{dt}}$$

dot product

$$= \frac{\partial f}{\partial x_1}\frac{dx_1}{dt} + \frac{\partial f}{\partial x_2}\frac{dx_2}{dt} + \ldots + \frac{\partial f}{\partial x_n}\frac{dx_n}{dt}$$

but $J_f = \left[\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \ldots \quad \frac{\partial f}{\partial x_n} \right]$

so $\dfrac{\partial f}{\partial \mathbf{x}} = (J_f)^T$

$\therefore \boxed{\dfrac{df}{dt} = J_f \dfrac{d\mathbf{x}}{dt}}$

for $f(x(t))$

since

$\vec{a} \cdot \vec{b}$

$= (a_1\ a_2 \cdots a_n) \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$

---

given, $f(x) = 5x$

$x(u) = 1-u$

$u(t) = t^2$

using substitution,

$f(t) = 5(1-t^2)$

$= 5 - 5t^2$

$\dfrac{df}{dt} = -10t$

using chain rule,

$\dfrac{df}{dt} = \dfrac{df}{dx}\dfrac{dx}{du}\dfrac{du}{dt}$

$= 5\,(-1)\,(2t)$

$= -10t$

given, $f\left(x\left(u(t)\right)\right)$

$$f(x) = f(x_1, x_2)$$

$$x(u) = \begin{bmatrix} x_1(u_1, u_2) \\ x_2(u_1, u_2) \end{bmatrix}$$ [vector valued function]

$$u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$ (vector valued function)
(t scalar input)

$$\frac{df}{dt} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial u} \frac{du}{dt}$$

$$\frac{df}{dt} = \frac{\partial f}{\quad} \frac{\quad}{dt}$$

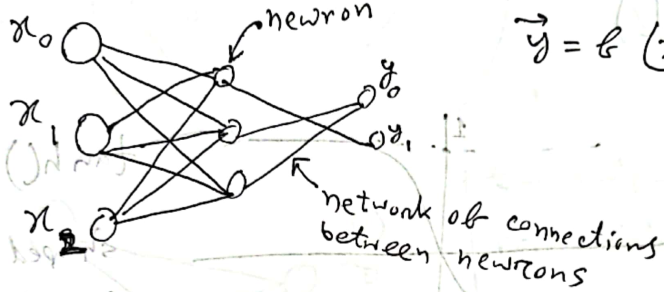
$$= \frac{\partial f}{\partial x} \frac{\partial x}{\quad} \frac{\quad}{dt}$$

$$= \frac{\partial f}{\partial x} \frac{\partial x}{\partial u} \frac{du}{dt}$$

$$= \begin{bmatrix} \frac{\partial f}{\partial x_1}, & \frac{\partial f}{\partial x_2} \end{bmatrix} \begin{bmatrix} \frac{\partial x_1}{\partial u_1} & \frac{\partial x_1}{\partial u_2} \\ \frac{\partial x_2}{\partial u_1} & \frac{\partial x_2}{\partial u_2} \end{bmatrix} \begin{bmatrix} \frac{du_1}{dt} \\ \frac{du_2}{dt} \end{bmatrix}$$

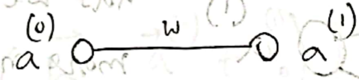
$$\underset{\text{scalar}}{(1\times1)} = (1\times2)(2\times2)(2\times1)$$

Simple neural network



$$\vec{y} = f(\vec{x})$$

a neural network is a mathematical function which takes a variable in and gives a variable back where both of these variables could be vectors.



a network that takes a scalar $a^{(0)}$ and gives back a scalar $a^{(1)}$

$$a^{(1)} = \sigma\left(w a^{(0)} + b\right)$$

b, w are numbers
σ is a function.

$a \Rightarrow$ activity

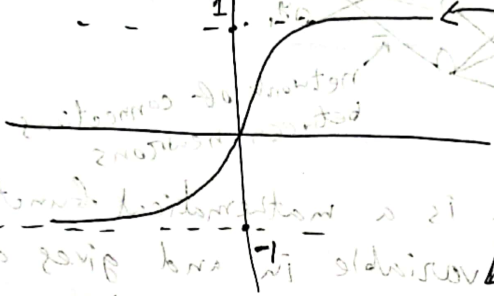
$w \Rightarrow$ weight

$b \Rightarrow$ bias

$\sigma \Rightarrow$ activation function

the term activation function comes from the fact that in our brain, when the chemical stimulation reaches a threshold amount, the neuron is activated and the neuron stimulates its neighbor neurons.

a function that has the threshold property is tanh()



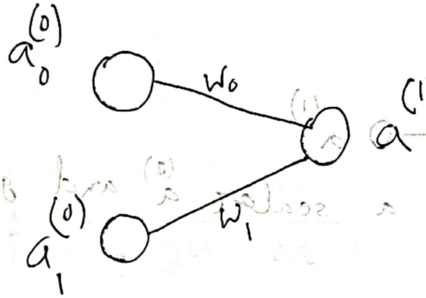
$$\tanh() = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

shaped like S

Sigmoid function

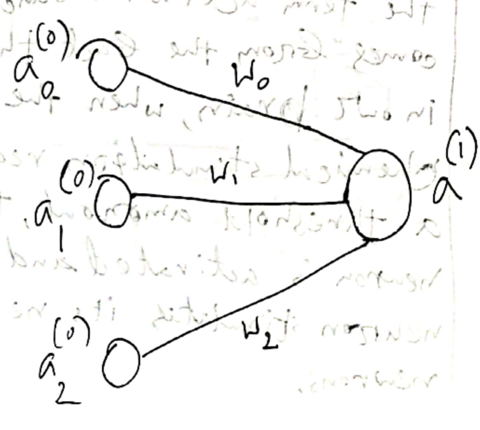
σ

each link has its own bias w_i



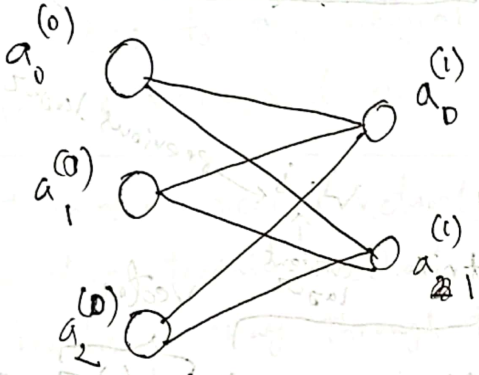
now we have added more neurons to increase the functionality of our neural network.

$$a^{(1)} = \sigma\left(w_0 a_0^{(0)} + w_1 a_1^{(0)} + b\right)$$



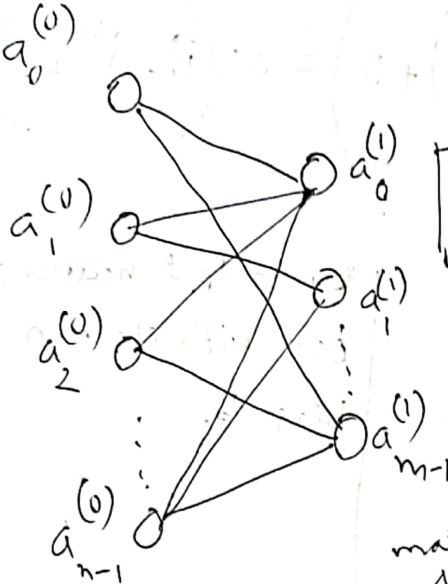
$$a^{(1)} = \sigma\left(w_0 a_0^{(0)} + w_1 a_1^{(0)} + w_2 a_2^{(0)} + b\right)$$

2) $$a^{(1)} = \sigma\left(\left(\sum_{j=0}^{n} w_j a_j^{(0)}\right) + b\right) = \sigma\left(\vec{w} \cdot \vec{a}^{(0)} + b\right)$$



two output neurons, each with its own bias b_i

$$\left.\begin{aligned} a_0^{(1)} &= \sigma\left(\vec{w}_0 \cdot \vec{a}^{(0)} + b_0^{(1)}\right) \\ a_1^{(1)} &= \sigma\left(\vec{w}_1 \cdot \vec{a}^{(0)} + b_1^{(1)}\right) \end{aligned}\right\} \underbrace{a^{(1)}}_{\begin{bmatrix} a_0^{(1)} \\ a_1^{(1)} \end{bmatrix}} = \sigma\Big(\underbrace{W^{(1)}}_{\begin{pmatrix} \vec{w}_0 & \vec{w}_1 \end{pmatrix}} \cdot \vec{a}^{(0)} + \underbrace{\vec{b}^{(1)}}_{\begin{bmatrix} b_0^{(1)} \\ b_1^{(1)} \end{bmatrix}}\Big)$$



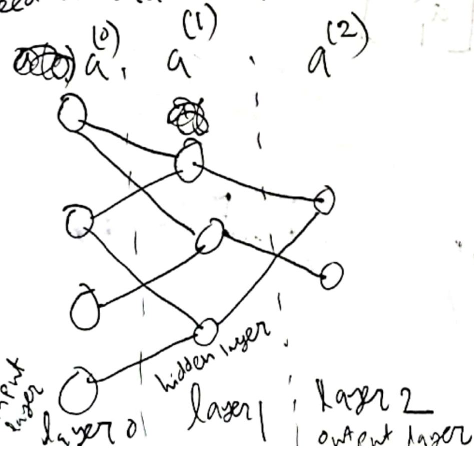
$$\boxed{\vec{a}^{(1)} = \sigma\left(W^{(1)} \cdot \vec{a}^{(0)} + \vec{b}^{(1)}\right)}$$

W_{ij} ← previous layer (j)

↑ current layer (i)

$$\begin{bmatrix} a_0^{(1)} \\ a_1^{(1)} \\ \vdots \\ a_{m-1}^{(1)} \end{bmatrix} = \sigma\left(\overbrace{\begin{bmatrix} W_{0,0}^{(1)} & W_{0,1}^{(1)} & \cdots & W_{0,n-1} \\ W_{1,0}^{(1)} & W_{1,1}^{(1)} & \cdots & W_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ W_{m-1,0}^{(1)} & W_{m-1,1} & \cdots & W_{m-1,n-1}^{(1)} \end{bmatrix}}^{\text{matrix}} \overbrace{\begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_{n-1}^{(0)} \end{bmatrix}}^{\text{vector}} + \begin{bmatrix} b_0^{(1)} \\ b_1^{(1)} \\ \vdots \\ b_{m-1}^{(1)} \end{bmatrix} \right)$$

feed forward network



$$a^{(1)} = \sigma\left(W^{(1)} \cdot a^{(0)} + b^{(1)}\right)$$

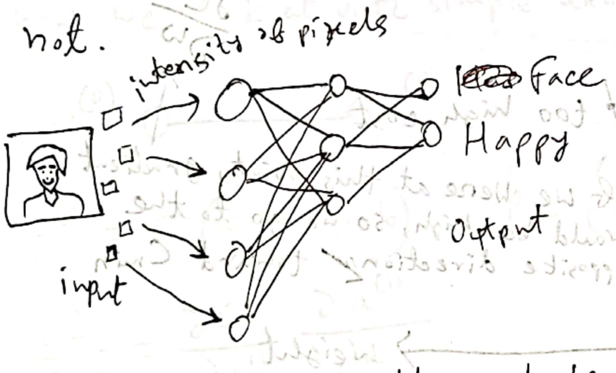
$$a^{(2)} = \sigma\left(W^{(2)} \cdot a^{(1)} + b^{(2)}\right)$$

$$\boxed{a^{(L)} = \sigma\left(W^{(L)} \cdot a^{(L-1)} + b^{(L)}\right)}$$

We'll have ~~the~~ to assign the right weights and biases to our neural network for it to do image recognition.

For training our neural network, we will use back propagation training method which starts at outputs and works its way through the hidden layer into the input layer.

We will determine whether a face is happy or not.



Which values of weights and biases best fits our data?

first we feed the neural network with some data to get some initial values which wont be quite useful but it'll set up the initial values.

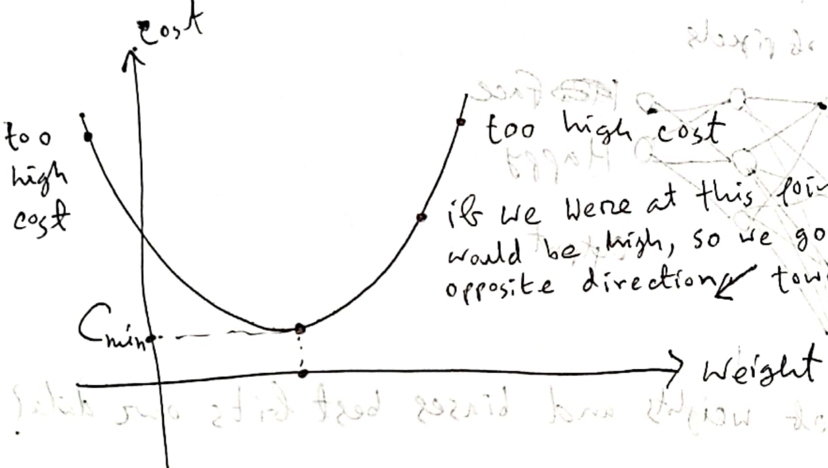
$$a^{(L)} = \sigma\left(W^{(L)} \cdot a^{(L-1)} + b^{(L)}\right)$$

now we define a cost function

$$C = \sum_i \left(a_i^{(L)} - y_i\right)^2$$

current value   desired value

we have to minimize the cost function.

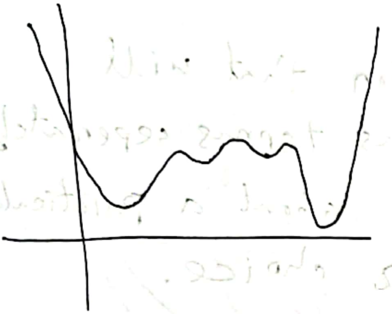


$\frac{\partial C}{\partial W}$

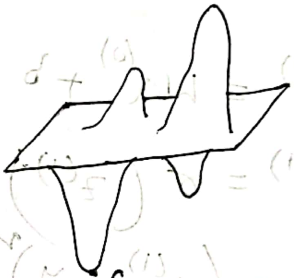
if we were at this point, gradient would be high, so we go to the opposite direction toward C_{min}

(considering only one weight)

but the graph may have several local minima



and we want to determine minimum of multidimensional hypersurface. then we have to build a jacobian.



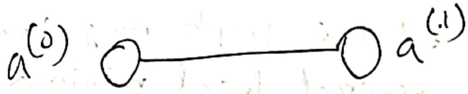
now we look at our simple example

$$a^{(0)} \circ\!\!-\!\!-\!\!-\!\!-\!\!\circ\ a^{(1)}$$

$$a^{(1)} = \sigma\left(w\, a^{(0)} + b\right) \qquad C = \left(a^{(1)} - y\right)^2$$

$$\frac{\partial C}{\partial w} = \frac{\partial C}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial w}$$

$$\frac{\partial C}{\partial b} = \frac{\partial C}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial b}$$



now we include a function that will hold our weight plus bias terms seperately. This will help us to think about a particular sigmoid function of our choice.

$$z^{(1)} = wa^{(0)} + b$$

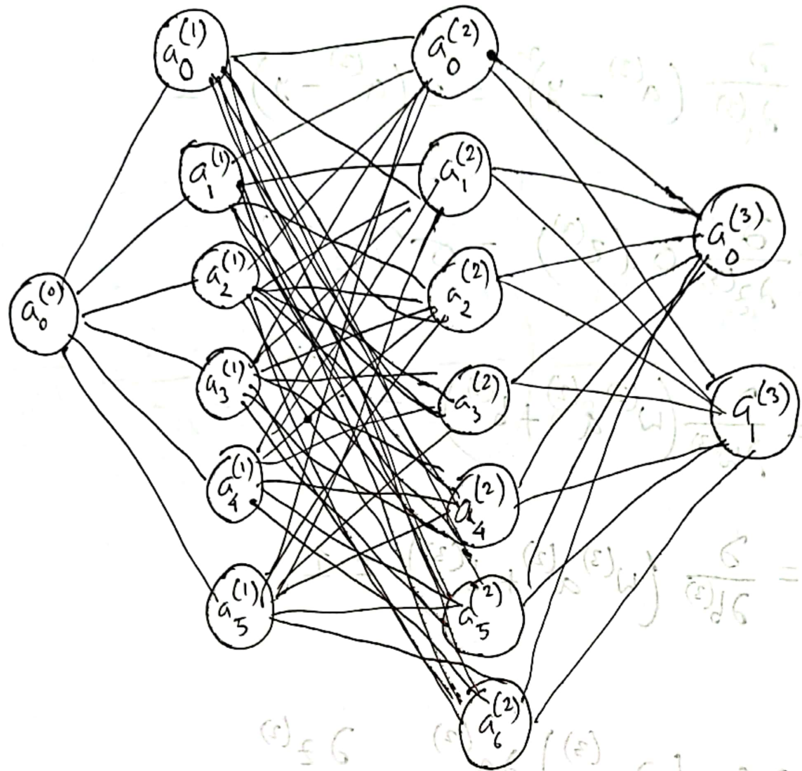
$$a^{(1)} = \sigma\left(z^{(1)}\right)$$

$$C = \left(a^{(1)} - y\right)^2$$

$$\frac{\partial C}{\partial w} = \frac{\partial C}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial w}$$

$$\frac{\partial C}{\partial b} = \frac{\partial C}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial b}$$

now We have the tools to navigate around the 2-D W-B space to minimize the cost function.



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

$$z^{(n)} = W^{(n)} a^{(n-1)} + b^{(n)}$$

$$J_{W^{(3)}} = \frac{\partial C}{\partial W^{(3)}} \qquad J_{b^{(3)}} = \frac{\partial C}{\partial b^{(3)}} \qquad C = \frac{1}{N}\sum_{k} C_k$$

$$\frac{\partial C}{\partial W^{(3)}} = \frac{\partial C}{\partial a^{(3)}} \frac{\partial a^{(3)}}{\partial z^{(3)}} \frac{\partial z^{(3)}}{\partial W^{(3)}}$$

$$\frac{\partial C}{\partial b^{(3)}} = \frac{\partial C}{\partial a^{(3)}} \frac{\partial a^{(3)}}{\partial z^{(3)}} \frac{\partial z^{(3)}}{\partial b^{(3)}}$$

layer 3

$$\frac{\partial C}{\partial a^{(3)}} = \frac{\partial}{\partial a^{(3)}}\left(a^{(3)}-y\right)^2 = 2\left(a^{(3)}-y\right) \cdot 1$$

$$\frac{\partial a^{(3)}}{\partial z^{(3)}} = \frac{\partial}{\partial z^{(3)}}\sigma\left(z^{(3)}\right) = \sigma'\left(z^{(3)}\right)$$

$$\frac{\partial z^{(3)}}{\partial w^{(3)}} = \frac{\partial}{\partial w^{(3)}}\left(w^{(3)}a^{(2)}+b^{(3)}\right) = a^{(2)}$$

$$\frac{\partial z^{(3)}}{\partial b^{(3)}} = \frac{\partial}{\partial b^{(3)}}\left(w^{(3)}a^{(2)}+b^{(3)}\right) = 1$$

layer 2

$$\frac{\partial C}{\partial w^{(2)}} = \frac{\partial C}{\partial a^{(3)}}\left(\frac{\partial a^{(3)}}{\partial a^{(2)}}\right)\frac{\partial a^{(2)}}{\partial z^{(2)}}\,\frac{\partial z^{(2)}}{\partial w^{(2)}}$$

$$\frac{\partial C}{\partial b^{(2)}} = \frac{\partial C}{\partial a^{(3)}}\left(\frac{\partial a^{(3)}}{\partial a^{(2)}}\right)\frac{\partial a^{(2)}}{\partial z^{(2)}}\,\frac{\partial z^{(2)}}{\partial b^{(2)}}$$

$$\frac{\partial a^{(3)}}{\partial a^{(2)}} = \frac{\partial a^{(3)}}{\partial z^{(3)}}\,\frac{\partial z^{(3)}}{\partial a^{(2)}} = \sigma'\left(z^{(3)}\right)w^{(3)}$$

layer-1

$$\frac{\partial C}{\partial W^{(1)}} = \frac{\partial C}{\partial a^{(3)}} \left(\frac{\partial a^{(3)}}{\partial a^{(2)}} \frac{\partial a^{(2)}}{\partial a^{(1)}} \right) \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial W^{(1)}}$$

$$\frac{\partial C}{\partial b^{(1)}} = \frac{\partial C}{\partial a^{(3)}} \left(\frac{\partial a^{(3)}}{\partial a^{(2)}} \frac{\partial a^{(2)}}{\partial a^{(1)}} \right) \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial b^{(1)}}$$

$\boxed{\text{Taylor series}}$ → useful for approximations of complicated functions

time to cook a chicken,

$t(m, T, \text{OvenFactor}, \text{ChickenShapeFactor})$

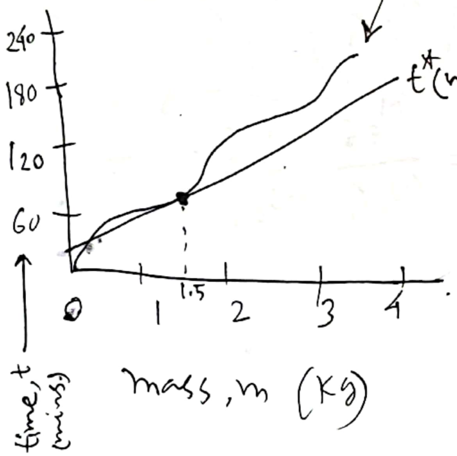
$= 7.33m^5 - 72.3m^4 + 253m^3 - 368m^2 + 250m + 0.02 + \text{OvenFactor} + \text{ChickenShapeFactor}$

how do we simplify this?

<u>2 assumptions:</u>

- Everyone has same oven
- chicken size 0-4 Kg

$\therefore t(m) = 7.33m^5 - 72.3m^4 + 253m^3 - 368m^2 + 250m + 0.02$



$t^*(m) \approx 50m + 15$

taylor approximation at $m = 1.5$ Kg

$g(x) = a + bx + cx^2 + dx^3 + \ldots$ (power series)

0th order approximation:

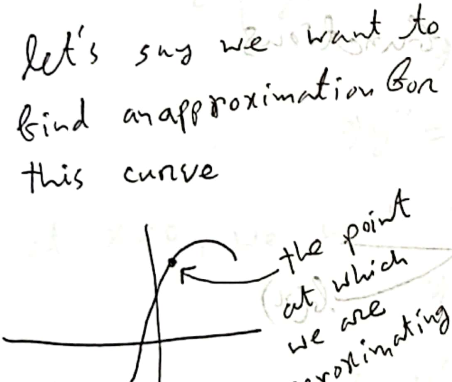
$g_0(x) = a$

$g_1(x) = a + bx$ (1st order approximation)

$g_2(x) = a + bx + cx^2$

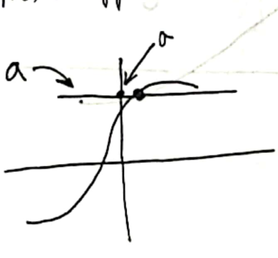
g_0, g_2, g_1 are called truncated series.

let's say we want to find an approximation for this curve



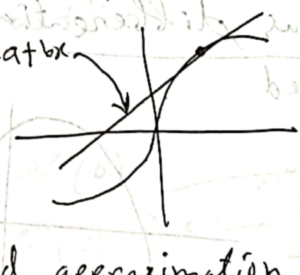
our first approximation

$g_0(x) = a$



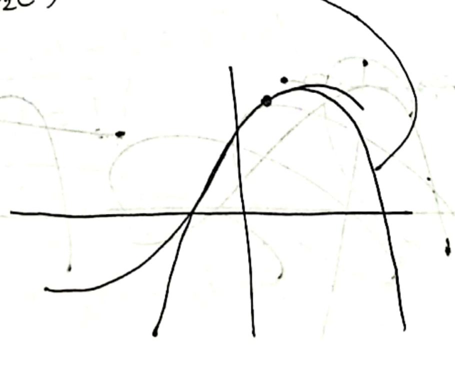
second approximation:

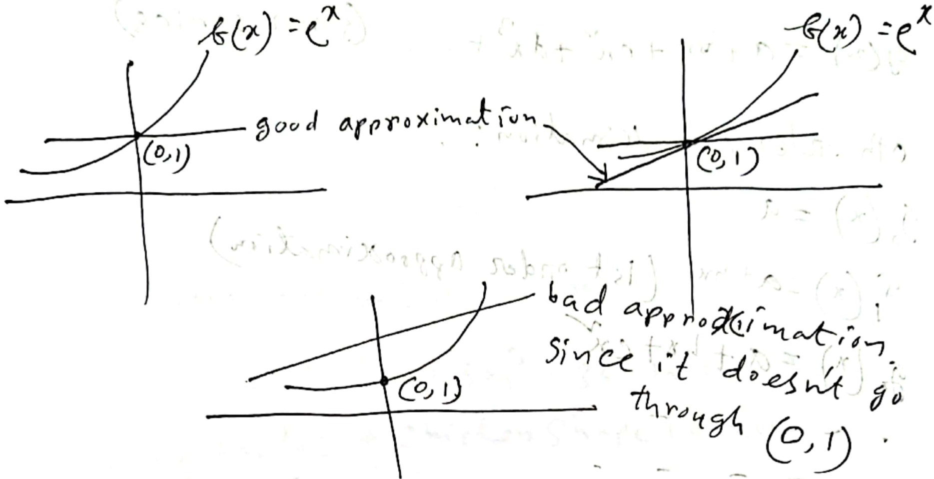
$g_1(x) = a + bx$



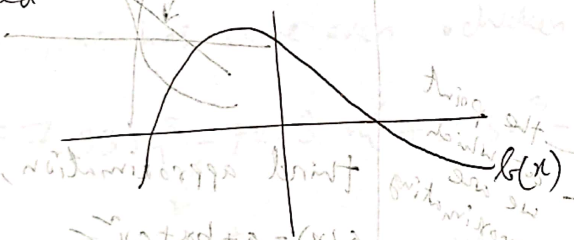
third approximation, (2nd order)

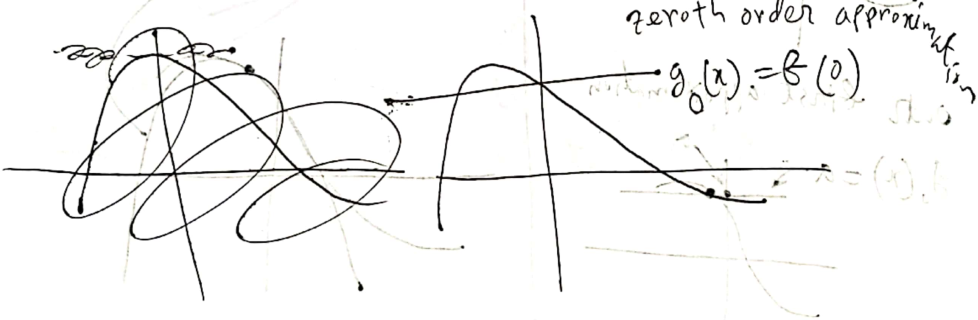
$g_2(x) = a + bx + cx^2$



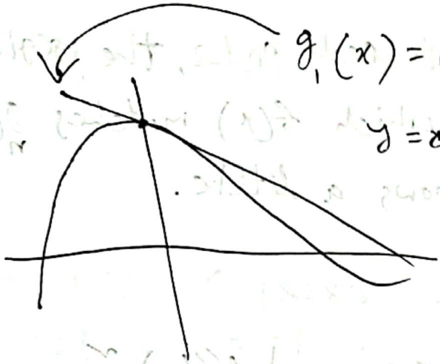


taylor series approximation works for continuous, differentiable functions.
well behaved





$g_1(x) = f(0) + f(0)\,x$, (first order approximation)



$y = c + m\,x$

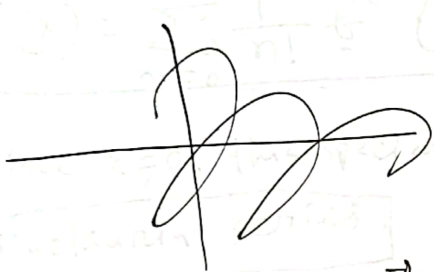
straight line

For our next approximation, we need a quadratic equation

$$y = ax^2 + bx + c$$

$$\Rightarrow y' = 2ax + b$$

$$\Rightarrow y'' = 2a$$



at $x=0$, we want $y'' = f''(0)$

$$\Rightarrow 2a = f''(0) \Rightarrow \boxed{a = \frac{f''(0)}{2}}$$

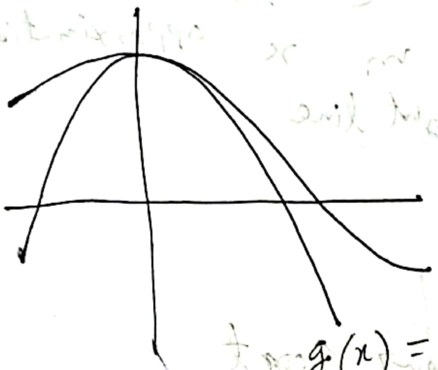
at $x=0$, we want $y' = f'(0)$

$$\Rightarrow 2a \cdot 0 + b = f'(0)$$

$$\Rightarrow \boxed{f'(0) = b}$$

at $x=0$, we want $y = f(0)$

$$\Rightarrow a \cdot 0^2 + b \cdot 0 + c = f(0) \Rightarrow \boxed{f(0) = c}$$



with each order, the region in which $f(x)$ matches $g_n(x)$ grows a little.

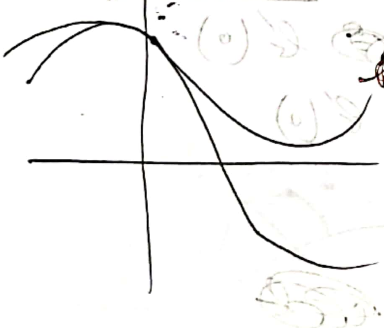
$$g_2(x) = f(0) + f'(0)x + \frac{1}{2} f''(0) x^2$$

$$y = ax^3 + bx^2 + cx + d$$

$$y' = 3ax^2 + 2bx + c$$

$$y'' = 6ax + 2b$$

$$y''' = 6a \Rightarrow f'''(0) = 6a \Rightarrow \boxed{a = \frac{f'''(0)}{6}}$$



$$g_3(x) = f(0) + f'(0)x + \frac{1}{2} f''(0) x^2 + \frac{1}{6} f'''(0) x^3$$

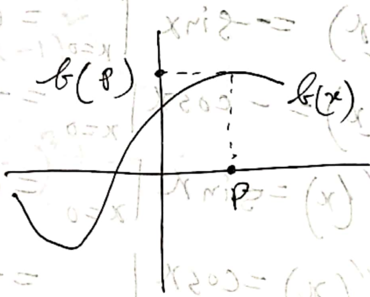
$$g_4(x) = f(0) + f'(0)x + \frac{1}{2}f''(0)x^2 + \frac{1}{6}f^{(3)}(0)x^3 + \frac{1}{24}f^{(4)}(0)x^4$$

$2 = 1\times2 = 2!$ $\quad 6 = 1\times2\times3 = 3!$ $\quad 24 = 1\times2\times3\times4 = 4!$

$$\boxed{g(x) = \sum_{n=0}^{\infty} \frac{1}{n!} f^{(n)}(0)\, x^n} \leftarrow \text{taylor series}$$

since $x=0$, this special case is called $\boxed{\text{Maclaurin series.}}$

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$



mclauren series $\rightarrow x=0$

taylor series $\longrightarrow x=p$

for linear approximation, $y = mx + c = f'(p)x + c$

$\Rightarrow f(p) = f'(p)p + c \Rightarrow c = f(p) - f'(p)p$

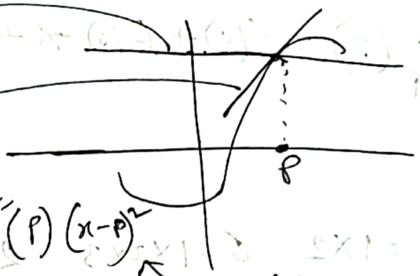
$\therefore y = f'(p)x + f(p) - f'(p)p$

$\quad = f'(p)(x-p) + f(p)$

$$g_0(x) = f(p)$$

$$g_1(x) = f(p) + f'(p)(x-p)$$

$$g_2(x) = f(p) + f'(p)(x-p) + \frac{1}{2} f''(p)(x-p)^2$$



same as mclauren for $p=0$

$$\boxed{g(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(p)}{n!}(x-p)^n}$$

one dimensional taylor series

Example $f(x) = \cos x$ $\quad g(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(0)}{n!} x^n$

$f'(x) = -\sin x \Big|_{x=0} = 0$ (for odd n)

$f''(x) = -\cos x \Big|_{x=0} = -1$ (for even n)

$f'''(x) = \sin x \Big|_{x=0} = 0$

$f''''(x) = \cos x \Big|_{x=0} = -1$

$$\cos x = 1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720} + \dots = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}$$

when we keep plotting for higher order terms, the plots looks more and more accurate for higher n close to the region $x=0$.

We have to be aware of which region the approximation is accurate. (domain)

$\boxed{\text{Example}}$ $f(x) = \frac{1}{x}$ (undefined at $x=0$, so we must use taylor series)

$$g(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(p)}{n!}(x-p)^n$$

at $p=1$, $g_0(x) = \frac{f^{(0)}(1)}{0!}(x-1)^0 = 1$

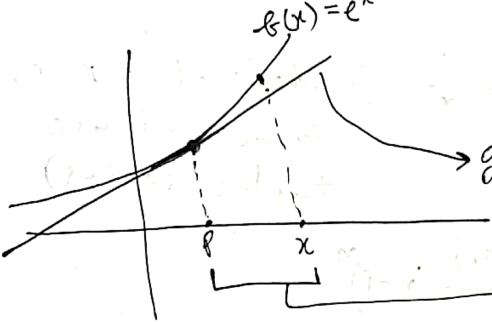
$$g_1(x) = \frac{f^{(1)}(1)}{1!}(x-1)^1 = -1(x-1) = -(x-1)$$

$$g_2(x) = \frac{f^{(2)}(1)}{2!}(x-1)^2 = (x-1)^2$$

$$\frac{1}{x} = 1-(x-1)+(x-1)^2-(x-1)^3+\cdots$$

$$\boxed{\frac{1}{x} = \sum_{n=0}^{\infty}(-1)^n(x-1)^n}$$

How to express error in taylor series?

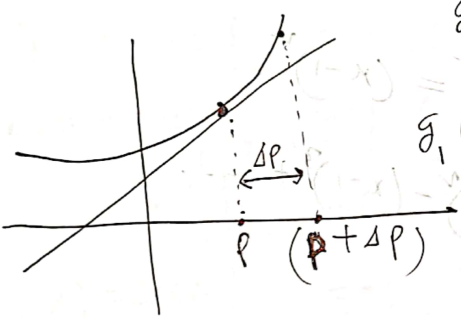


$$g(x)=\sum_{n=0}^{\infty}\frac{f^{(n)}(p)}{n!}(x-p)^n$$

$$g_1(x)=f(p)+\underbrace{f'(p)}_{\text{slope}}\underbrace{(x-p)}_{\text{run}}$$

$$\text{gradient}=\frac{\text{rise}}{\text{run}} \Rightarrow \text{rise}=\text{run}\times\text{gradient}$$

now we replace some terms



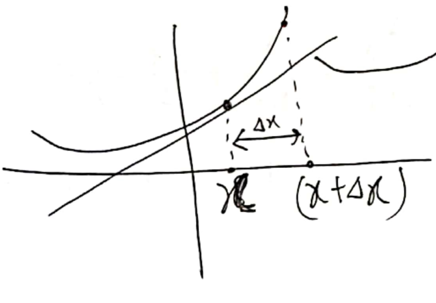
$$g_1(x)=f(p)+f'(p)(x-p)$$

$$\downarrow$$

$$g_1(p+\Delta p)=f(p)+f'(p)\Delta p$$

$$\downarrow$$

$$g_1(x+\Delta x)=f(x)+f'(x)(\Delta x)$$



our new taylor series notation,

$$\boxed{f(x+\Delta x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x)}{n!}\Delta x^n} \qquad \boxed{\Delta x = x-p}$$

$$f(x+\Delta x) = f(x) + f'(x)\underbrace{\Delta x}_{\text{small}} + \frac{f''(x)}{2}\underbrace{\Delta x^2}_{\text{very small}} + \frac{f^{(3)}(x)}{6}\underbrace{\Delta x^3}_{\text{very very small}} + \dots$$

$$f(x+\Delta x) = f(x) + f'(x)(\Delta x) + O(\Delta x^2)$$

error occurs at order of Δx^2

error $\propto \Delta x^2$

second order accurate

this way of expressing a function avoiding higher order terms is called $\boxed{\text{linearization}}$.

$$f(x+\Delta x) = f(x) + f'(x)\Delta x + \frac{f''(x)}{2}\Delta x^2 + \frac{f^{(3)}(x)}{6}\Delta x^3 + \dots$$

$$\Rightarrow f'(x)\Delta x = f(x+\Delta x) - f(x) - \frac{f''(x)}{2}\Delta x^2 - \frac{f^{(3)}(x)}{6}\Delta x^3 + \dots$$

$$\Rightarrow f'(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x} - \frac{f''(x)}{2}\Delta x - \frac{f^{(3)}(x)}{6}\Delta x^2 + \dots$$

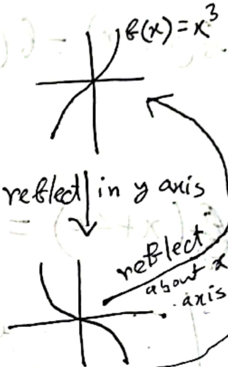
$$\Rightarrow f'(x) = \underbrace{\frac{f(x+\Delta x) - f(x)}{\Delta x}}_{\text{forward difference method}} - O(\Delta x)$$

first order accurate

odd function: $-f(x) = f(-x)$ → rotational symmetry with respect to origin

even function: $f(-x) = f(x)$

→ symmetry about y-axis →  $f(x)=x^2$

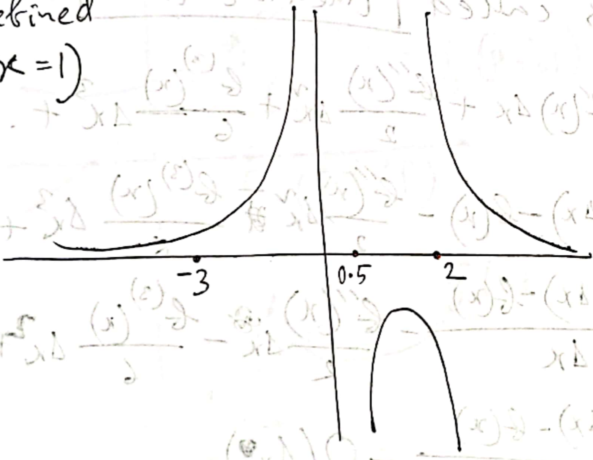


taylor polynomial for an odd function contains only odd powers.

taylor polynomial for an even function contains only even powers.

$$f(x) = \frac{2}{x^2 - x}$$

(undefined at $x = 1$)



$x = -3$ will approximate $f(x)$ for $x < 0$ only

$x = 2$ will appr. $f(x)$ in $x > 1$ only

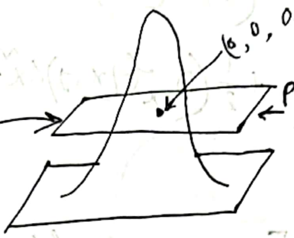
$x = 0.5$ will approximate $f(x)$ in $0 < x < 1$ only

Multivariate Taylor series

$$f(x, y) = e^{-(x^2 + y^2)}$$

zeroth order approximation at $(0,0,1)$ and $(0,0,\frac{1}{2})$

$g_0(x+\Delta x, y+\Delta y)$

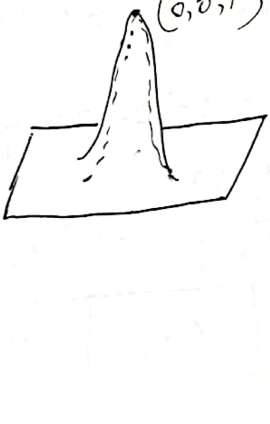


Plane parallel to xy plane

1st order approximation

$g_1(x+\Delta x, y+\Delta y)$

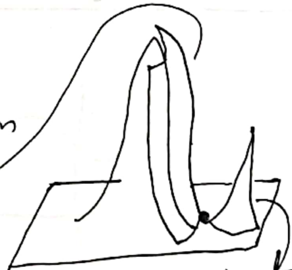
same height and gradient as the point



2nd order approximation

$g_2(x+\Delta x, y+\Delta y)$

(parabolic surface)



saddle function

gradient and curvature matches with $f(x, y)$ close to the point

$$f(x+\Delta x,\ y+\Delta y) \ \ldots$$

$$= f(x,y) + \left(\partial_x f(x,y)\,\Delta x + \partial_y f(x,y)\,\Delta y\right)$$

$$+ \frac{1}{2}\left(\partial_{xx} f(x,y)\,\Delta x^2 + 2\,\partial_{xy} f(x,y)\,\Delta x\,\Delta y + \partial_{yy} f(x,y)\,\Delta y^2\right)$$

$$+ \ldots$$

$$= f(x,y) + \left[\partial_x f(x,y),\ \partial_y f(x,y)\right]\begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix}$$

$$+ \frac{1}{2}\left[\Delta x,\ \Delta y\right]\begin{bmatrix}\partial_{xx} f(x,y) & \partial_{xx} f(x,y)\\ \partial_{yx} f(x,y) & \partial_{yy} f(x,y)\end{bmatrix}\begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix}$$

$$\boxed{= f(x,y) + J_f\,\Delta x + \frac{1}{2}\,\Delta x^t H_f\,\Delta x + \ldots}$$

transpose of Δx

Newton Raphson method

→ how to find ~~minima/maxima~~ solution of a function when $f(x)$ is complicated and in higher dimensions.
(difficult to plot)

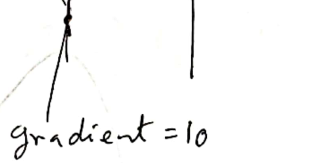
$$y = x^3 - 2x + 2$$

$$\frac{\partial y}{\partial x} = 3x^2 - 2$$

we want to find the solution where $y = 0$. let's take an arbitrary point in $\frac{\partial y}{\partial x}$ and hunt for solution. we can guess that $x = -2$ because $\frac{\partial y}{\partial x}$ has a negative constant.

$$y(-2) = (-2)^3 - 2(-2) + 2 = -8 + 4 + 2 = -2$$

$$\frac{\partial y}{\partial x} = 3(-2)^2 - 2 = 10 \text{ (+ve)}$$



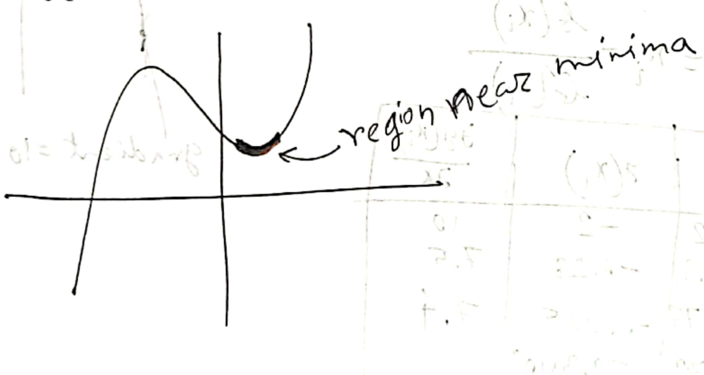
$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

i	x_i	$y(x_i)$	$\frac{\partial y(x_i)}{\partial x}$
0	-2	-2	10
1	-1.8	-0.28	7.5
2	-1.77	-0.005	7.4
3	-1.769	-2.3×10^{-6}	

but the newton-raphson method fails when $x=0$ for this $f(x)$.

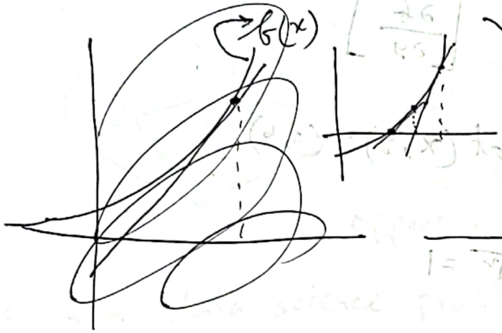
i	x_i	$g(x_i)$	$\frac{\partial y(x_i)}{\partial x}$
0	0	2	-2
1	1	1	1
2	0	2	

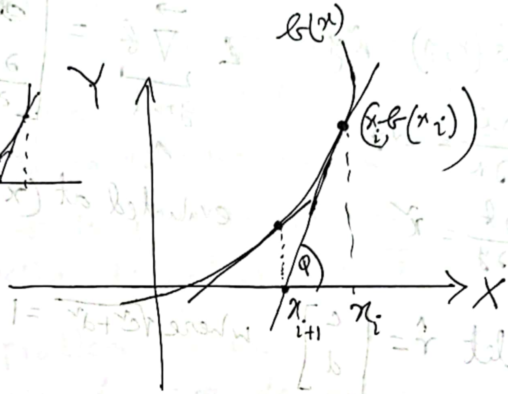
another issue is that if we pick x close to a maxima / minima, the gradient would be 0 or close to zero. when we divide by the gradient $\left(\frac{f(x_i)}{f'(x_i)}\right)$, x_{i+1} will be far off from what we want it to be. so the newton-raphson (will not converge) method will fail again.



Derivation

The newton raphson method works on the following principle:





first we take a guess x_i and draw tangent to $f(x)$ and see that the tangent crosses X axis at some point where $y=0$. we take that point x_{i+1} as our next guess where we draw tangent to $f(x)$ again. this way we'll get closer and closer to the point in $f(x)$ where $y=0$.

$$f'(x) = \tan\varphi = \frac{f(x_i) - 0}{x_i - x_{i+1}} \Rightarrow f'(x)\,x_i - f'(x)\,x_{i+1} = f(x_i)$$

$$\Rightarrow f'(x)\,x_{i+1} = f'(x)\,x_i - f(x_i)$$

$$\Rightarrow \boxed{x_{i+1} = x_i - \frac{f(x_i)}{f'(x)}}$$

$\boxed{\text{Gradient descent}}$

$f(x,y) = x^2 y$ $\qquad$ $\underbrace{\vec{\nabla} f}_{\text{grad}} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$

$\frac{\partial f}{\partial x} = 2xy$ $\qquad$ $\uparrow$ evaluated at $(x,y) = (a,b)$

$\frac{\partial f}{\partial y} = x^2$

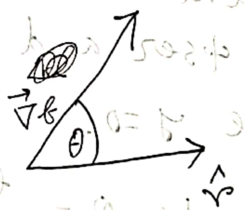
let $\hat{r} = \begin{bmatrix} c \\ d \end{bmatrix}$ where $\sqrt{c^2+d^2} = 1$

so directional gradient $= \begin{bmatrix} \frac{\partial f}{\partial x}\Big|_{(a,b)} \\ \frac{\partial f}{\partial y}\Big|_{(a,b)} \end{bmatrix} \cdot \begin{bmatrix} c \\ d \end{bmatrix}$

$= \vec{\nabla} f \cdot \hat{r}$

what is the maximum value of directional derivative?

$\vec{\nabla} f \cdot \hat{r} = \|\vec{\nabla} f\| \, \|\hat{r}\| \cos\theta$



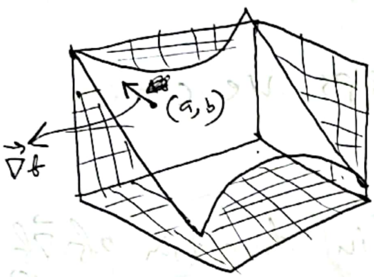
$\cos\theta_{max} = 1$ (when $\theta = 0^\circ$)

so $\hat{r}$ must be parallel to $\vec{\nabla} f$

since $\hat{r}$ is a unit vector, $\hat{r} = \frac{\vec{\nabla} f}{\|\vec{\nabla} f\|}$

$\therefore \vec{\nabla} f \cdot \frac{\vec{\nabla} f}{\|\vec{\nabla} f\|} = \frac{\|\vec{\nabla} f\|^2}{\|\vec{\nabla} f\|} = \|\vec{\nabla} f\|$ $\leftarrow$ maximum gradient of f in (a,b)

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



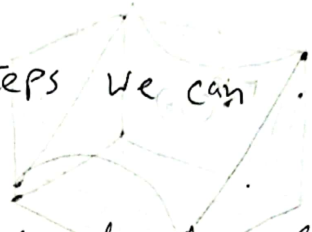
$\vec{\nabla} f$ points to the steepest descent in the point (a,b) so all we have to do to go to the minimum of f is to go to the opposite direction of $\vec{\nabla} f$

so in a data science problem if we want to go from a bad result to our desired result, we want to minimize (find minima) the error function. gradient descent will help us find at which point f is minimum.

$\vec{\nabla} f$ is perpendicular to the contour lines

$$S_{n+1} = S_n - \gamma \vec{\nabla} f(S_n)$$

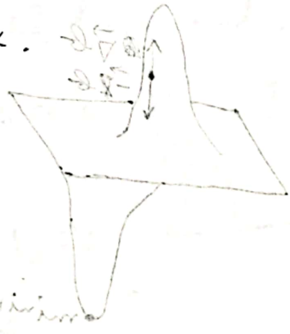
by taking a series of small steps we can reach the minima.



γ=amount of small step we take in direction of $\vec{\nabla} f(S_n)$

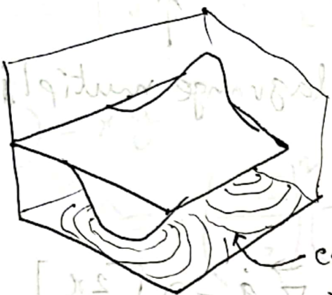
we can set gamma to a small number (say 0.5) or we can set $-H_f^{-1} \cdot J$ as our step size.

if (a,b) is close to the minima, this works quite well. if (a,b) is far away from minima, it may not work.

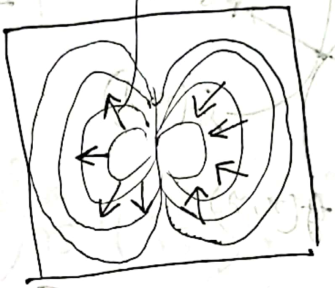


Lagrange multiplier

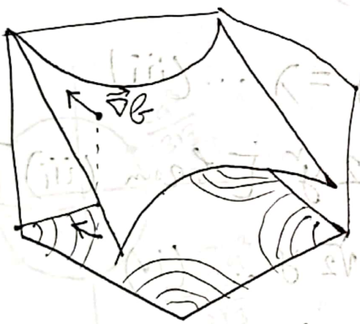
find maxima / minima subject to some constraint curve (circle, straight line etc).



$f(x,y) = x e^{-(x^2+y^2)}$



contour map



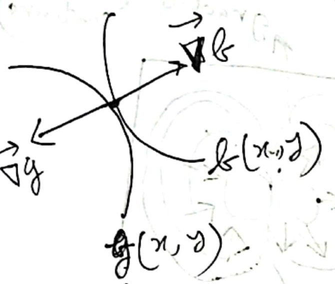
$f(x,y) = x^2 y$

what's the max value of $x^2 y$ along $g(x,y) = x^2 + y^2 = a^2$ circle?

lagrange noticed that f is max / min at the points $\vec{\nabla} f$ touches contours and $\vec{\nabla} f$ is $\perp$ to g at those points.

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

constraint $g(x,y) = x^2 + y^2 = a^2$ .... (i)



<u>solve:</u> $\vec{\nabla} f = \lambda \vec{\nabla} g$

lagrange mutiplier

$$\vec{\nabla} f = \vec{\nabla}(x^2 y) = \begin{bmatrix} 2xy \\ x^2 \end{bmatrix} = \lambda \vec{\nabla} g = \lambda \begin{bmatrix} 2x \\ 2y \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 2xy \\ x^2 \end{bmatrix} = \begin{bmatrix} 2\lambda x \\ 2\lambda y \end{bmatrix} \quad \ldots\ldots \text{(ii)}$$

$\therefore$ from (ii), $2xy = 2\lambda x \Rightarrow y = \lambda$ ... (iii)

from (ii) $x^2 = 2\lambda y \Rightarrow x^2 = 2y^2$ [from (iii)]

$\Rightarrow x = \pm\sqrt{2}\, y$ ... (iv)

from (i), $x^2 + y^2 = a^2 \Rightarrow (\pm\sqrt{2}\,y)^2 + y^2 = a^2$ [from (iv)]

$\Rightarrow 2y^2 + y^2 = a^2 \Rightarrow 3y^2 = a^2$

$\Rightarrow y^2 = \frac{a^2}{3} \Rightarrow \boxed{y = \pm\frac{a}{\sqrt{3}}}$ ... (v)

~~$x = \pm\sqrt{2}\, y = \pm\sqrt{2}\left(\pm\frac{a}{\sqrt{3}}\right)$~~

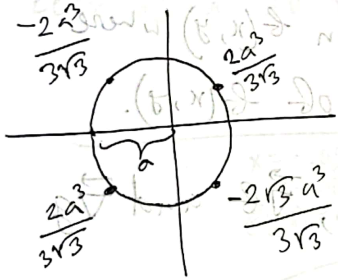
Solution: $\begin{bmatrix} x \\ y \end{bmatrix} = \frac{a}{\sqrt{3}}\begin{bmatrix} \sqrt{2} \\ 1 \end{bmatrix}, \frac{a}{\sqrt{3}}\begin{bmatrix} \sqrt{2} \\ -1 \end{bmatrix}, \frac{a}{\sqrt{3}}\begin{bmatrix} -\sqrt{2} \\ 1 \end{bmatrix}, \frac{a}{\sqrt{3}}\begin{bmatrix} -\sqrt{2} \\ -1 \end{bmatrix}$

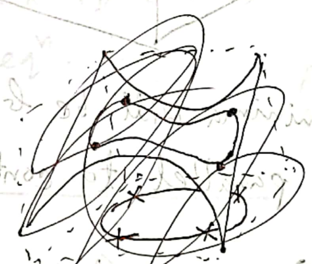
$$f(x,y) = x^2 y$$

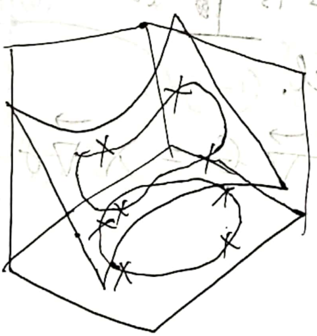
$$= \left(\frac{a\sqrt{2}}{\sqrt{3}}\right)^2 \cdot \frac{a}{\sqrt{3}}, \left(\frac{a\sqrt{2}}{\sqrt{3}}\right)^2\left(-\frac{a}{\sqrt{3}}\right), \left(\frac{-a\sqrt{2}}{\sqrt{3}}\right)^2 \frac{a}{\sqrt{3}}, \left(\frac{-\sqrt{2}a}{\sqrt{3}}\right)^2\left(-\frac{a}{\sqrt{3}}\right)$$

$$= \frac{2a^3}{3\sqrt{3}}, -\frac{2\sqrt{2}a^3}{3\sqrt{3}}, \frac{2a^3}{3\sqrt{3}}, \frac{-2a^3}{3\sqrt{3}}$$

max min max min





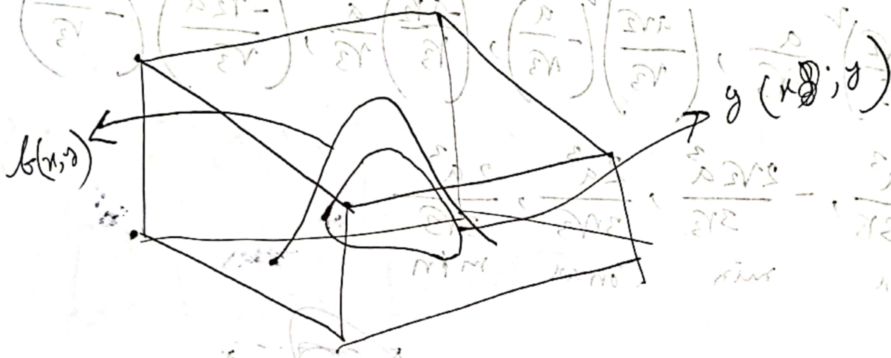


plot in matlab for better graph.

$\boxed{\text{example}}$ find max, min of $f(x,y) = e^{\left(-\frac{2x^2+y^2-xy}{2}\right)}$

along the curve $g(x,y) = x^2 + 3(y+1)^2 - 1 = 0$

$f(x)$ does not have any minima itself, but along $g(x)$ there are two minima and two maxima.



maxima / minima will be found on $f(x,y)$ where $g(x,y)$ is parallel to contours of $f(x,y)$.

since $\vec{\nabla} f$ is $\perp$ to contours, $\vec{\nabla} f$ and $\vec{\nabla} g$ are parallel.

$$\vec{\nabla} f = \lambda \vec{\nabla} g$$

$$\Rightarrow \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \lambda \begin{bmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial g}{\partial y} \end{bmatrix}$$

$$\vec{\nabla} \mathcal{L}(x,y,\lambda) = f(\vec{x}) - \lambda g(\vec{x})$$

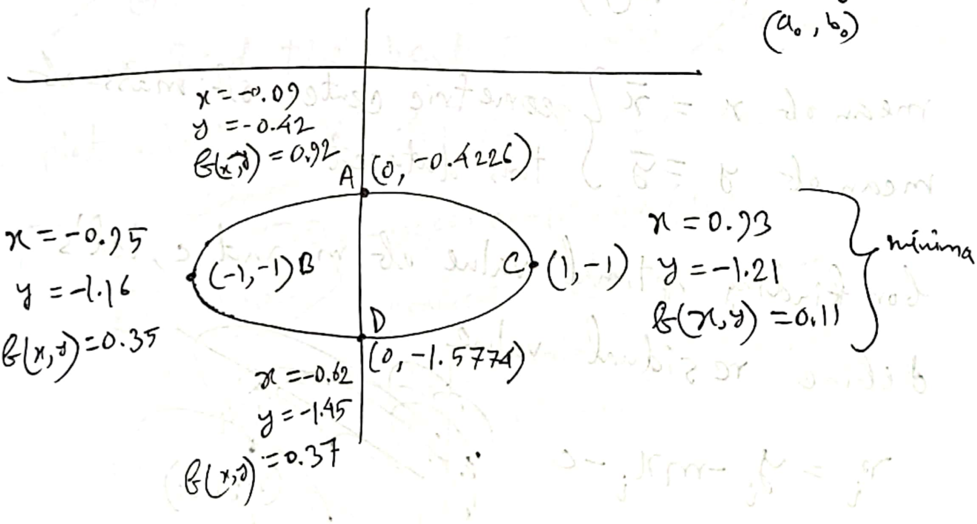
$$\vec{\nabla} \mathcal{L}(x,y,\lambda) = \begin{bmatrix} \frac{\partial f}{\partial x} - \lambda \frac{\partial g}{\partial x} \\ \frac{\partial f}{\partial y} - \lambda \frac{\partial g}{\partial y} \\ -g(\vec{x}) \end{bmatrix} = 0$$

$$g(\vec{x}) = 0 \rightarrow g(x,y) = 0$$

~~We have co~~ now finding the zeros of this system will solve our problem.

We have gone from finding a minimum of 2D function constrained to a 1D curve, to finding zeros of a 3D vector eq^n.

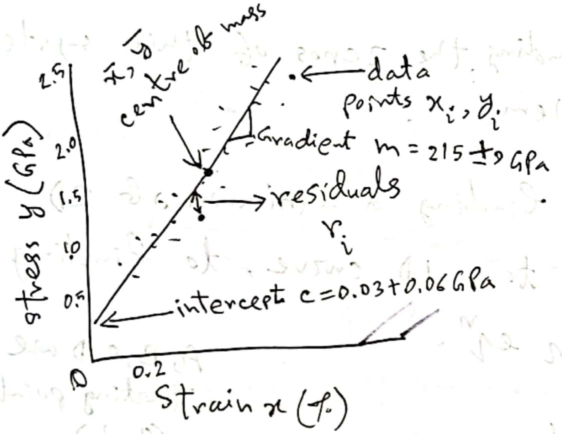
A, B, C, D are starting points (a_0, b_0)



Simple linear regression

Say we have some data from our experiments.

For fitting the data, we may use some theoretical knowledge about the data to find a formula or figure out a formula based on how the distribution looks.



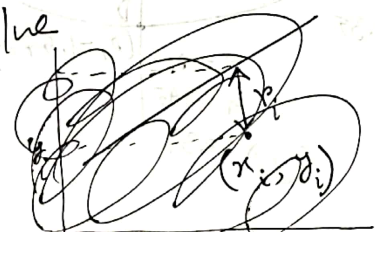
$$y = y(x; a_i) = mx_i + c$$

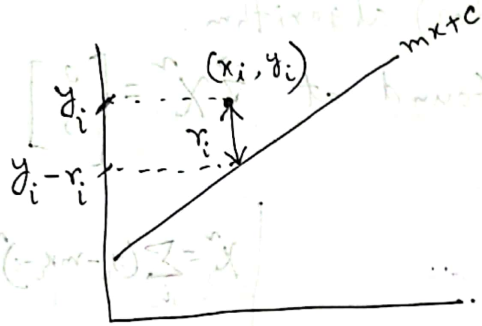
$$a = \begin{bmatrix} m \\ c \end{bmatrix}$$

mean of $x = \bar{x}$ } geometric center of mass of this dataset
mean of $y = \bar{y}$

for finding optimal value of m and c, let's define residual value

$$r_i = y_i - mx_i - c$$





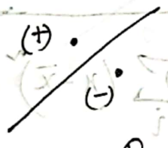
$$y_i - r_i = mx_i + c$$

$$\Rightarrow r_i = y_i - mx_i - c$$

now we define a term to measure the overall quality of the fit : χ^2 (chi squared)

$$\chi^2 = \sum_i r_i^2 = \sum_i (y_i - mx_i - c)^2$$

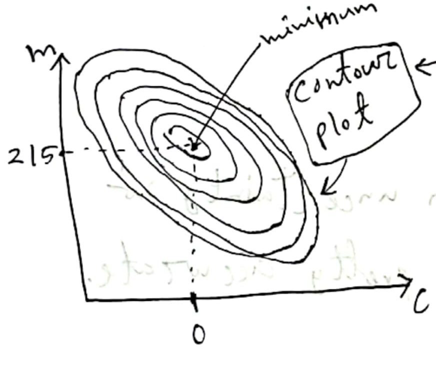
we are taking χ^2 because we want all the data above and below the fit.



we don't want any confusions related to pluses and minuses.

now we find the best possible χ^2 (minimization)

we plot values of χ^2 for different values of m and c



← 200,000 computations (slow)

gradient descent may be tricky, since minimum isn't reachable from all points we need a better (Faster) algorithm.

The minimum will be found at $\vec{\nabla}\chi^2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$$\Rightarrow \begin{bmatrix} \frac{\partial \chi^2}{\partial m} \\ \frac{\partial \chi^2}{\partial c} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\chi^2 = \sum_i (y_i - mx_i - c)^2$$

$$\Rightarrow \begin{bmatrix} -2\sum_i x_i (y_i - mx_i - c) \\ -2\sum_i (y_i - mx_i - c) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$c = \frac{\sum y_i}{n} - m\frac{\sum x_i}{n} = \bar{y} - m\bar{x}$$

n = number of data items

uncertainty in c, $\sigma_c \simeq \sigma_m \sqrt{\bar{x}^2 + \frac{1}{n}\sum_i (x - \bar{x})^2}$

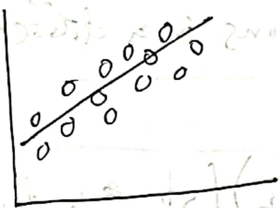
uncertainty in m, $\sigma_m^2 \simeq \dfrac{\chi^2}{\sum (x-\bar{x})^2 (n-2)}$

$$m = \frac{\sum (x - \bar{x}) y}{\sum (x - \bar{x})^2}$$

using this method, we get an uncertainty of only 5% at m, which is pretty accurate.

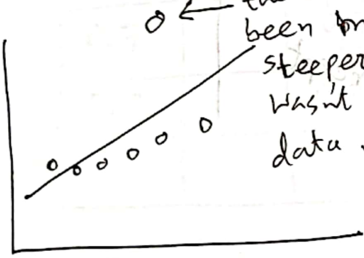
Anscombe's quartet

4 graphs with same χ^2

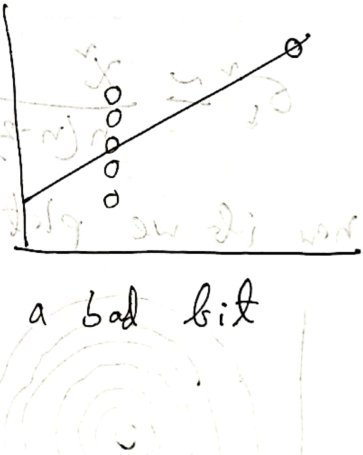


linear regression does what it is expected to do here

using linear regression isn't a good idea here



the fit would've been much steeper if it wasn't for this data point



a bad fit

another optimization:

$$c = \bar{y} - m\bar{x}$$

intercept c depends on gradient m.

if we redefine our expressions in a different manner,

$$y = (m \pm \sigma_m)(x - \bar{x}) + (b \pm \sigma_b)$$

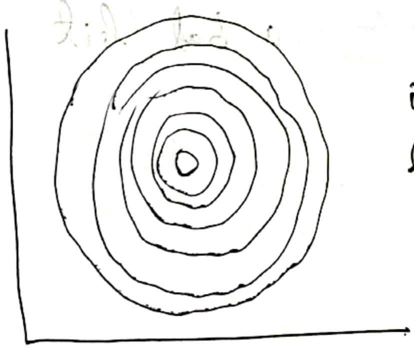
$$\sigma_m^2 \simeq \frac{\chi^2}{\sum (x-\bar{x})(n-2)}$$

$$\sigma_b^2 \simeq \frac{\chi^2}{n(n-2)}$$

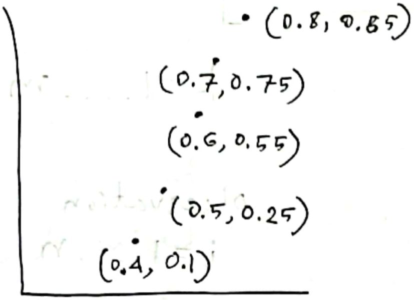
$b = \bar{y}$ intercept

so b doesn't depend on m anymore

now if we plot χ^2,



it is much easier to find minima from this contour plot. Almost all points lead to minima.



$$c = \bar{y} - m\bar{x} = \frac{\sum y_i}{N} - m\frac{\sum x_i}{N}$$

$$= \frac{2.5}{5} - m\frac{3}{5}$$

$$= 0.5 - 2 \times 0.6$$

$$= -0.7$$

x_i	$\bar{x}$	y_i	$(x_i - \bar{x})y_i$	$(x_i - \bar{x})^2$
0.4		0.1	-0.02	0.04
0.5		0.25	-0.025	0.01
0.6	0.6	0.55	0	0
0.7		0.75	0.075	0.01
0.8		0.85	0.17	0.04
$\Sigma \rightarrow$			0.2	0.1

$$m = \frac{\sum (x_i - \bar{x})y_i}{\sum (x_i - \bar{x})^2} = \frac{0.2}{0.1} = 2$$

General non linear least squares

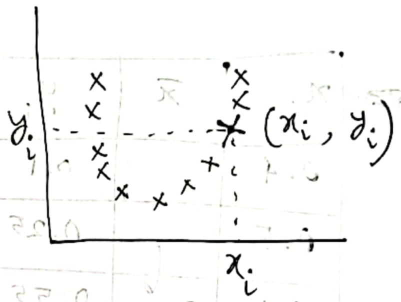
$y(x; a_k) = (x - a_1)^2 + a_2$ $\qquad k = 1 \dots m$

non-linear function

observation
$i = 1, \dots n$

(y_i, x_i, σ_i)

the more I am uncertain about the data point y_i, the greater σ_i will be



goodness of the fit will be defined by

$$\chi^2 = \sum_{i=1}^{n} \frac{[y_i - y(x_i; a_k)]^2}{\sigma^2}$$

we are penalizing $[y_i - y(x_i; a_k)]^2$ by σ^2 at every iteration, so that highly ~~bound~~ uncertain points have less weight.

if we don't ~~have~~ know σ, we can set σ to 1

$$\boxed{\nabla \chi^2 = 0}$$

We can find the solution from here by finding out algebraic expressions, but that might not always be possible or efficient.

so we will use steepest descent method where we updated our guesses gradually (similar to Newton-raphson)

$$a_{next} = a_{current} - \overset{\text{aggression}}{\text{constant}}\; \nabla \chi^2$$

$$\frac{d\chi^2}{da_k} = \sum_{i=1}^{n} -2 \frac{[y_i - y(x_i; a_k)]}{\sigma^2} \frac{dy}{da_k}$$

$$\therefore \boxed{a_{next} = a_{current} + \sum_{i=1}^{n} \frac{[y_i - y(x_i; a_k)]}{\sigma^2} \frac{dy}{da_k}}$$

$$\frac{dy}{da_1} = -2(x - a_1) \qquad \frac{dy}{da_2} = 1$$

$$\vec{J} = \left(\frac{\partial(\chi^2)}{\partial a_k} \right) = \left(\frac{\partial(\chi^2)}{\partial a_1} \quad \frac{\partial(\chi^2)}{\partial a_2} \right)$$

$$\underbrace{x}_{\text{variable}}\ ;\ \overbrace{\sigma, x_p, I, b}^{\text{parameters}}$$

$$f(x; \sigma, x_p, I, b) = b + \frac{I}{\sigma\sqrt{2\pi}} e^{\left(\frac{-(x-x_p)^2}{2\sigma^2}\right)}$$

the Jacobian matrix wich ~~represents~~ is required to correctly perform the nonlinear least square fit using the gauss function defined above

$$J = \left(\frac{\partial(\chi^2)}{\partial\sigma}, \frac{\partial(\chi^2)}{\partial x_p}, \frac{\partial(\chi^2)}{\partial I}, \frac{\partial(\chi^2)}{\partial b} \right)$$ [row matrix].

$$f(\sigma, x_p, I, b)$$

$$\frac{\partial \chi^2}{\partial a_j} = -2 \sum_{i=1}^{n} \frac{y_i - y(x_i; a)}{\sigma_i^2} \frac{\partial y(x_i; a)}{\partial a_j} \quad \text{for } j = 1 \ldots n$$

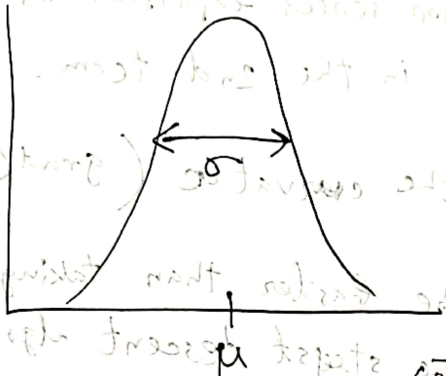
If we take a taylor series expansion of χ^2,
we get the hessian in the 2nd term.

↑

gives us an idea of the curvature (grad(J))

taking Hessian will be faster than taking
small steps like the steepst descent algorithm.
we'll be using H to give us an idea of the
size of steps we should take in gradient descent.
but H isn't very stable far from
the minimum.

$\boxed{\text{Levenberg - Marquardt method}}$ uses gradient
descent (small steps) when too far away from
minimum and H when close to minima.
when χ^2 is getting better



$$f(\vec{x};\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\vec{x}-\mu)^2}{2\sigma^2}}$$

$$\chi^2 = \left|\vec{y} - f(\vec{x};\mu,\sigma)\right|^2$$

$$\frac{\partial \chi^2}{\partial \mu} = 2\,\cancel{f}\left(\vec{y} - f(\vec{x};\mu,\sigma)\right)\cdot\frac{\partial}{\partial\mu}\left(-\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\vec{x}-\mu)^2}{2\sigma^2}}\right)$$

$$= -2\left(\vec{y} - f(\vec{x};\mu,\sigma)\right)\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\vec{x}-\mu)^2}{2\sigma^2}} \cdot \frac{\partial}{\partial\mu} - \frac{(\vec{x}-\mu)^2}{2\sigma^2}$$

$$= -2\left(\vec{y} - f(\vec{x};\mu,\sigma)\right)\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\vec{x}-\mu)^2}{2\sigma^2}} \cdot \frac{1}{\cancel{2}\sigma^2}\cancel{2}(\vec{x}-\mu)\cdot(-1)$$

$$= \frac{2(\vec{x}-\mu)}{\sigma^2}\,\cancel{f}\left(\vec{y} - f(\vec{x};\mu,\sigma)\right)\cdot f(\vec{x};\mu,\sigma)$$

$$= -2\frac{\partial f}{\partial\mu}\left(\vec{y} - f(\vec{x};\mu,\sigma)\right)$$

$$\frac{\partial f}{\partial \mu} = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(\vec{x}-\mu)^2}{2\sigma^2}} \cdot \frac{\partial}{\partial \mu} \frac{-(\vec{x}-\mu)^2}{2\sigma^2}$$

$$= \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(\vec{x}-\mu)^2}{2\sigma^2}} \cdot \frac{-1\cdot 2}{2\sigma^2}(\vec{x}-\mu)\cdot(-1)$$

$$= f(\vec{x};\, \mu, \sigma)\cdot(\sigma^{-2})\,(\vec{x}-\mu)$$

$$\cancel{\frac{\partial f}{\partial \sigma} = -\frac{1}{\sigma^2\sqrt{2\pi}}\, e^{-\frac{(\vec{x}-\mu)^2}{2\sigma^2}} \cdot \frac{\partial}{\partial \sigma} \frac{-(\vec{x}-\mu)^2}{2\sigma^2}}$$

$$\cancel{= -\frac{1}{\sigma}\, \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(\vec{x}-\mu)^2}{2\sigma^2}} \cdot \frac{-(\vec{x}-\mu)^2}{2}\left(\frac{-2}{\sigma^3}\right)}$$

$$\cancel{= -\frac{(\vec{x}-\mu)^2}{\sigma^4}\, f(\vec{x};\mu,\sigma)}$$

$$\frac{\partial f}{\partial \sigma} = \left(\frac{\partial}{\partial \sigma}\, \frac{1}{\sigma\sqrt{2\pi}}\right) e^{-\frac{(\vec{x}-\mu)^2}{2\sigma^2}} + \frac{1}{\sigma\sqrt{2\pi}}\, \frac{\partial}{\partial \sigma}\, e^{-\frac{(\vec{x}-\mu)^2}{2\sigma^2}}$$

$$= -\frac{1}{\sigma^2\sqrt{2\pi}}\, e^{\frac{-(\vec{x}-\mu)^2}{2\sigma^2}} + \frac{1}{\sigma\sqrt{2\pi}}\, e^{\frac{-(\vec{x}-\mu)^2}{2\sigma^2}} \cdot \frac{\partial}{\partial \sigma}\, \frac{-(\vec{x}-\mu)^2}{2\sigma^2}$$

$$= -\frac{1}{\sigma} f(\vec{x};\mu,\sigma) + f(\vec{x};\mu,\sigma)\, \frac{-(\vec{x}-\mu)^2}{\cancel{2}}\left(-\cancel{2}\sigma^{-3}\right)$$