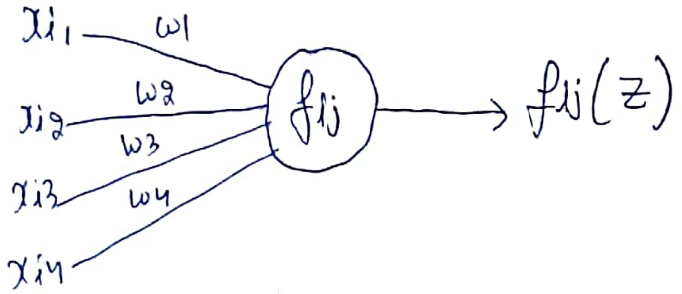
→ "Activation functions" :→

from 1980's till early 2000's there are two activation function which are most popular & these are :→

- ↳ Sigmoid function (Sigmoid activation unit)
- ↳ Tanh function (Tanh activation unit)

let's assume we have an activation unit, let's call it "f_{ij}". let's assume we have some inputs & some weights associated with these inputs.



Now when we multiply w_i & x_i & summate them, that, is equivalent to $W^T x$

$$\sum_i w_i x_i = W^T x$$

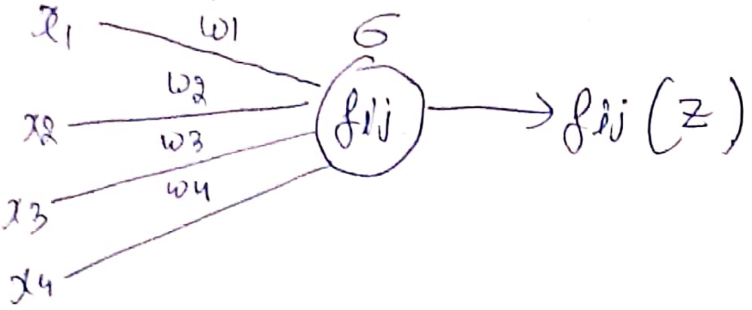
where $W = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}$ & $X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$

So, the input that we get at "f_{ij}" is the weighted sum of w_i & x_i.

let's call that input as, $Z = \sum_i w_i x_i = W^T x$.

if "fij" is the "sigmoid unit"

let's represent it with "σ" sign



$$z = \sum_i w_i x_i = w^T x$$

$$\& \ \sigma(z) = \frac{e^z}{1+e^z} \quad \text{or} \quad \frac{1}{1+e^{-z}}$$

This is what we used in "logistic regression" to squash the inputs.

"Requirements of the activation function":

1) It needs to be differentiable.

2) It should be easy & fast to differentiate.

we have already seen the usage of sigmoid function in "logistic regression".

let's now look at the advantages of sigmoid from "differentiation perspective"

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad \& \quad \boxed{z = w^T x}$$

Now if we compute derivative of ~~this~~ "σ" w.r.t. z.

we get.

P.T.O

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

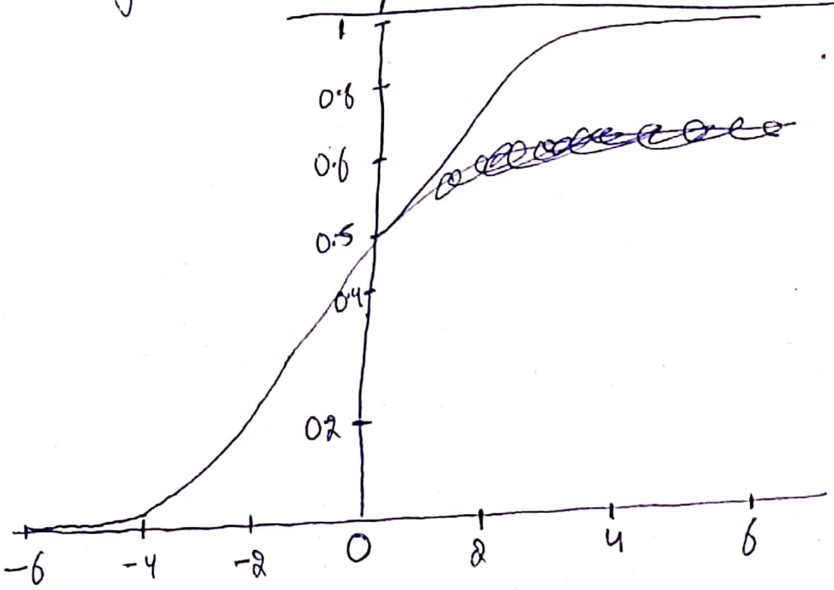
The important part here is, the derivative of the sigmoid function is represented in terms of sigmoid function itself.

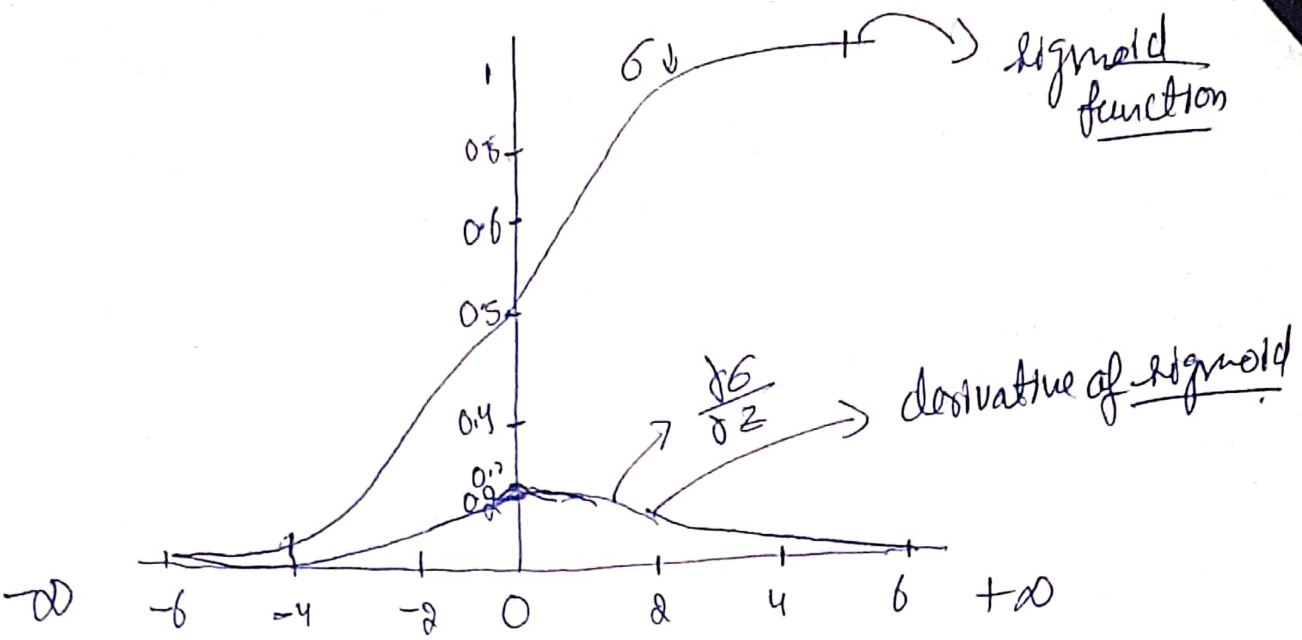
Let's say you write some code to implement this sigmoid function(z).

sigmoid (z) → [This function can be used both during forward propagation and backward propagation.]

↳ We can use the same function during "forward propagation". as well as computing derivatives during ["backward propagation."] for updating the weights.

The sigmoid function looks like an "S" shaped curve:





Sigmoid function is an S-shaped curve, It goes from $-\infty$ to $+\infty$, minimum value is "0" & maximum value is '1'

if we notice the derivative of sigmoid the max. value is ~~both~~ 0.2 & 0.3 & the derivative is only significant b/w {"−4" & "+4"} bcoz after −4 & +4 the derivative is extremely small.

Derivative basically says how fast is the sigmoid function changing.

$$\boxed{0 \le \frac{d\sigma}{\partial z} < 1}$$ (precisely less than 0.3)

There is another popular activation unit called tanh function

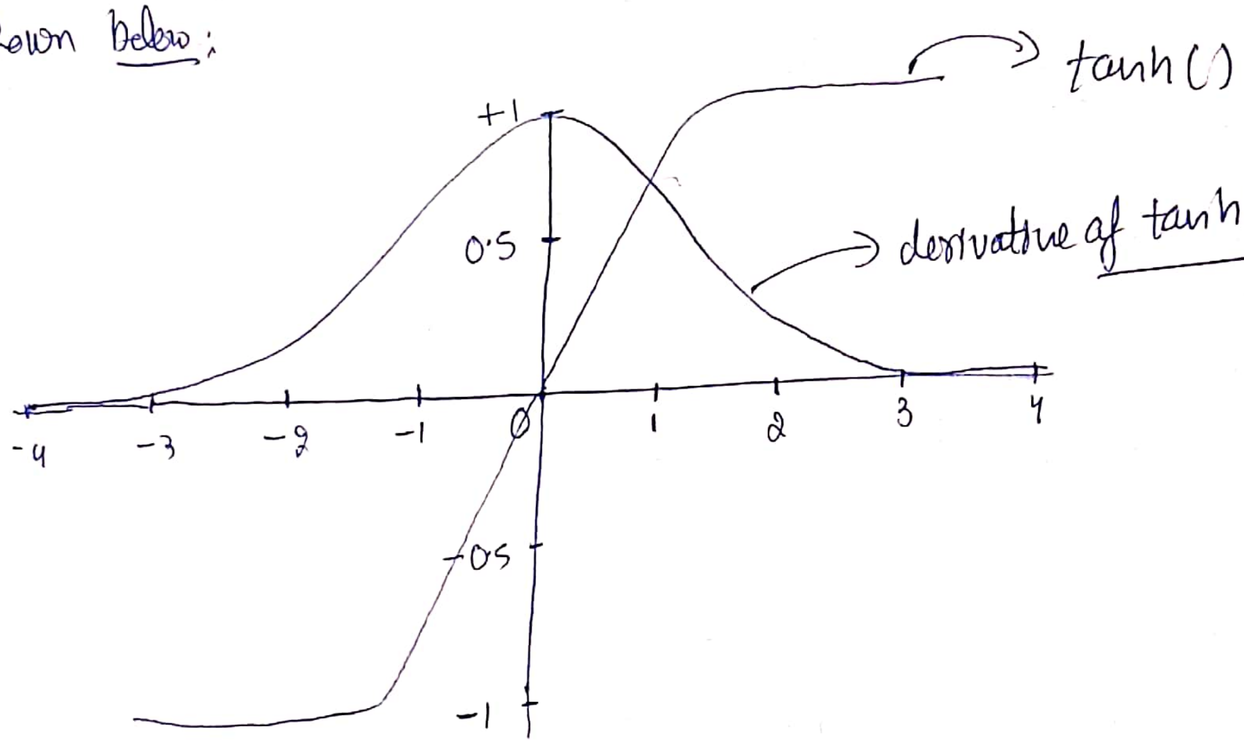
$$a = \tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

& $$\frac{\partial a}{\partial z} = 1 - a^2 = 1 - (\tanh^2(z))^2$$

tanh(z) function is very very similar to sigmoid function.

We know that derivative of sigmoid function can be represented in terms of sigmoid function itself, similarly derivative of tanh() function can also be represented in terms of tanh() function itself.

The plot of tanh() function & its derivative looks like as shown below:



tanh() function is also an S-shaped curve, but the "minimum value" is "-1" & "maximum value" is "+1".

Derivative of tanh(z) is more peaked.

It lies between "0" & "1". The maximum value of derivative of tanh() is at $\boxed{"1"}$ & minimum value of the derivative is at $\boxed{"0"}$.

& Between "-3" & "+3", it is having some reasonable value.

After '+3' & '-3', the derivative of "tanh( )" becomes very very small.

So traditionally "tanh( )" & "sigmoid( )" are the two mostly used activation functions, becoz they are differentiable & they are easily differentiable, becoz of that we can represent them using the same function. (the function as well as its derivative).

Note:-> Derivative of sigmoid function can be represented in terms of sigmoid function itself.

& Derivative of tanh( ) function can also be represented in terms of tanh( ) function itself.

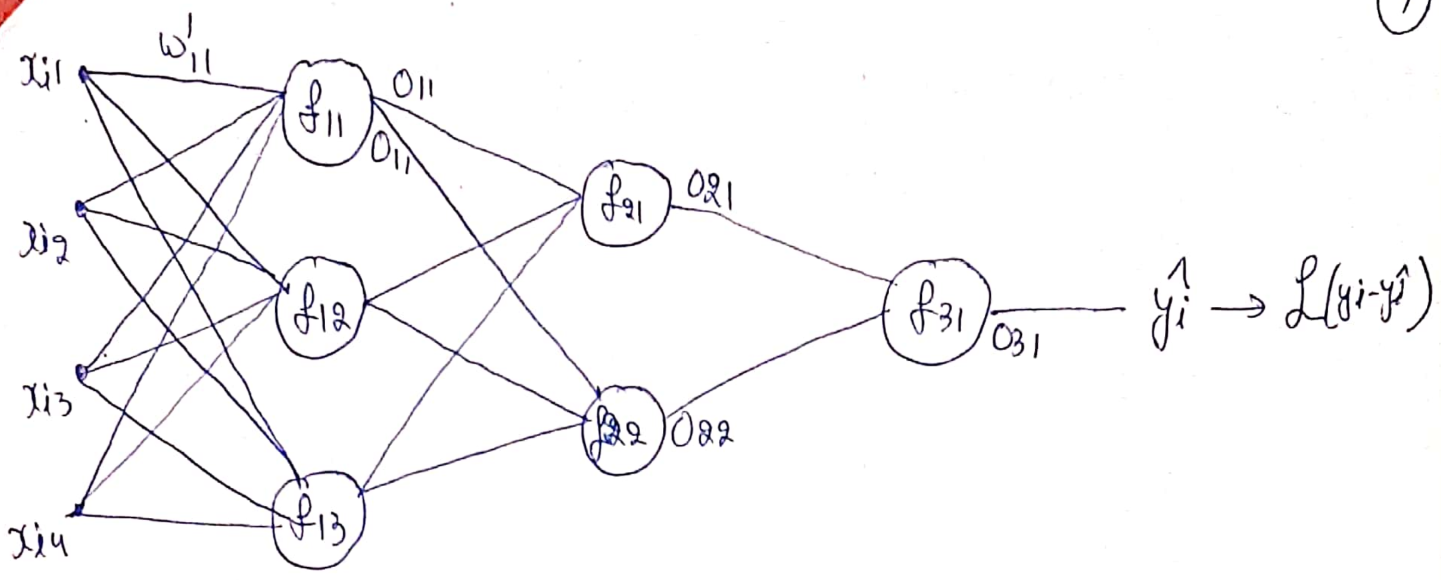
Even though these two are very popular. but the most popular activation function, we use now a days is the ReLU

↳ 'Rectified Linear Unit'.

→ Vanishing Gradients ('Problem') :->

One of the biggest problem with Neural Networks in 80's till early 2000s was this problem of vanishing Gradients.

This is literally one of the most important reason, why people lost interest in Neural Networks in late 90's & early 2000's. (This is not the only reason but one of the important reasons)



All of f_{ij}'s in earlier days, people used sigmoid or tanh() there typically as activation units.

There is one major problem with ["sigmoid()"] which is the same problem with ["tanh()"] also.

Let's understand the problem.

Suppose we want to update the weight "w^1_{11}".

We know that, to update this weight, we need to compute the derivative of "L" w.r.t. w^1_{11}:

As our update equation is:-

$$\left\{ (w^1_{11})_{new} = (w^1_{11})_{old} - \eta \left(\frac{\partial L}{\partial w^1_{11}}\right)_{old} \right\}$$

P.T.O

$$\frac{\partial L}{\partial w'_{11}} = \frac{\partial L}{\partial O_{31}}\left[\left(\frac{\partial O_{31}}{\partial O_{21}} \times \frac{\partial O_{21}}{\partial O_{11}} \cdot \frac{\partial O_{11}}{\partial w'_{11}}\right) + \left(\frac{\partial O_{31}}{\partial O_{22}} \times \frac{\partial O_{22}}{\partial O_{11}} \times \frac{\partial O_{11}}{\partial w'_{11}}\right)\right]$$

Now, we know that at all the activation units, we have used 'sigmoid' or 'tanh function' & in order to update the weights we need to compute the derivatives

& we know that derivative of sigmoid is $0 \leq \frac{\partial \sigma}{\partial z} \leq 1$

strictly speaking $\frac{\partial \sigma}{\partial z}$ is never reaching even 0.3 value

& in case of tanh() the derivative is strictly

b/w $\left[0 \leq \frac{\partial \tanh()}{\partial z} \leq 1\right]$

Since each of these activation units are sigmoid or tanh, the derivative of these will be b/w 0 & 1. Typically speaking about sigmoid, its derivative will always be less than 0.3

Here since we are multiplying these derivatives & since these derivatives are small so the multiplication term will be smaller even more.

Lets only focus on the first part here.

In the first multiplication term we have three numbers all less than 0.3 (as it is sigmoid)

$$\left[\frac{\partial O_{31}}{\partial O_{21}} \times \frac{\partial O_{21}}{\partial O_{11}} \times \frac{\partial O_{11}}{\partial w'_{11}}\right]$$

let it be

$$0.2 \times 0.1 \times 0.05 = 0.0010 = 10^{-3}$$

$<< 1$

very very less value

It is a minuscule value

& Remember, we are using this minuscule value in the update function.

$$\left[(w'_{11})_{new} = (w'_{11})_{old} - \eta\left(\frac{dL}{\partial w'_{11}}\right)\right]$$

let's say $(w'_{11})_{old}$ is 2.5

& $\eta = 1$ & let $\frac{\partial L}{\partial w'_{11}} = 0.001$

$$\text{Now} (w'_{11})_{new} = 2.5 - 1 * 0.001$$

$$= \underline{2.499}$$

which is very similar to the old value.

∴ New weight value is very very close to the old value (old weight value).

This is happening becoz derivative is small, & derivatives is small becoz individual derivatives are small & individual derivatives are small because we are using sigmoid() or tanh() as our activation units.

So, updating the weights become very very hard & training becomes very very slow.

Here we have used just a three layered network.
Imagine if we have a 20 layered or 10 layered network then updating weights for such a deep network would be even more tough.

Note :-> # of multiplication of derivatives =

# of hidden layers + 1

if we have 2 hidden layers then no. of multiplication of derivates is = 3

Similarly if we have 10 hidden layers, then no. of multiplication of derivatives is = 11

P.T.O

If just with 3 multiplication, we are getting such a small value of derivative

then imagine how minuscule the value would be if 11 such multiplication operations are performed.

In this example with 3 multiplication, we got derivate values as 10^{-3}

If we have eleven multiplication, then the value become even more smaller $= 10^{-12}$

$\downarrow$ vvv small

So, if the derivative is very small, then newer weights & older weights becomes almost the same.

Imagine the derivate was 10^{-10}

then our new weight would have been

$2.5 - 0.0000\ldots1$

old

$= 2.4999999\ldots9$

new.

which is exactly same.

~~when~~ This problem is called "Vanishing Gradient Problem", becoz the final gradient $\left(\frac{\partial L}{\partial w_{11}}\right)$ we are getting here is vanishing or it is becoming very very small. This is becoz of chain rule of multiplication of the sigmoid function we are using.

—o—

→ ReLU (Rectified Linear Units)

One of the big problems in classical neural networks era (pre-2006) have been the 'Vanishing Gradient Problem" which is very often seen when we are using a sigmoid() activation or a tanh() activation function.

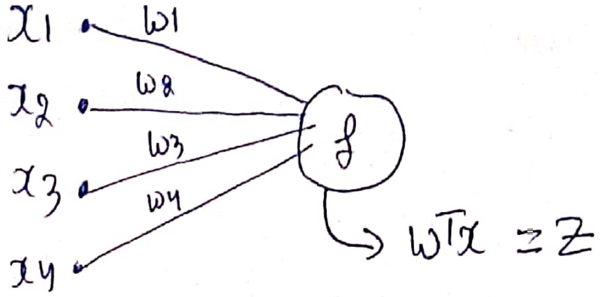
When we have this problem of 'vanishing gradient" our convergence also slows down & training ~~takes~~ becomes longer & longer.

The reason is, if we have vanishing gradient problem, the update is very very small.

"ReLU's were introduced in 2011 & ~~they~~ are one of the most important ideas in 'Deep Learning"

Note :- Today, all the activation problems are by default 'ReLU"

Let's now see how ReLU looks like :-



Now function $f(z)$ for ReLU is often written as

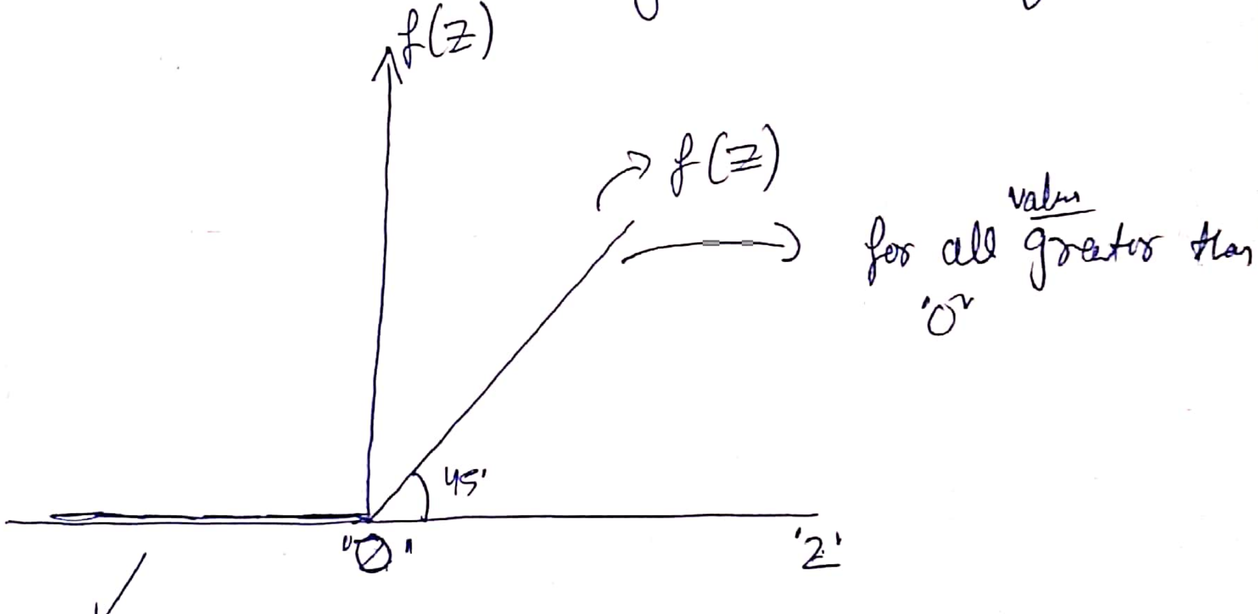
$$f(z) = z^{+} = \max(0, z)$$

↙

Relu definition (max value of 0, z)

Let's see how does it work :→

$$f(z) = \max(0, z) = \begin{cases} 0, & \text{if } z \leq 0 \\ z, & \text{otherwise} \end{cases}$$



↙

for all the values which are negative (less than 0) the value of the function is "0"

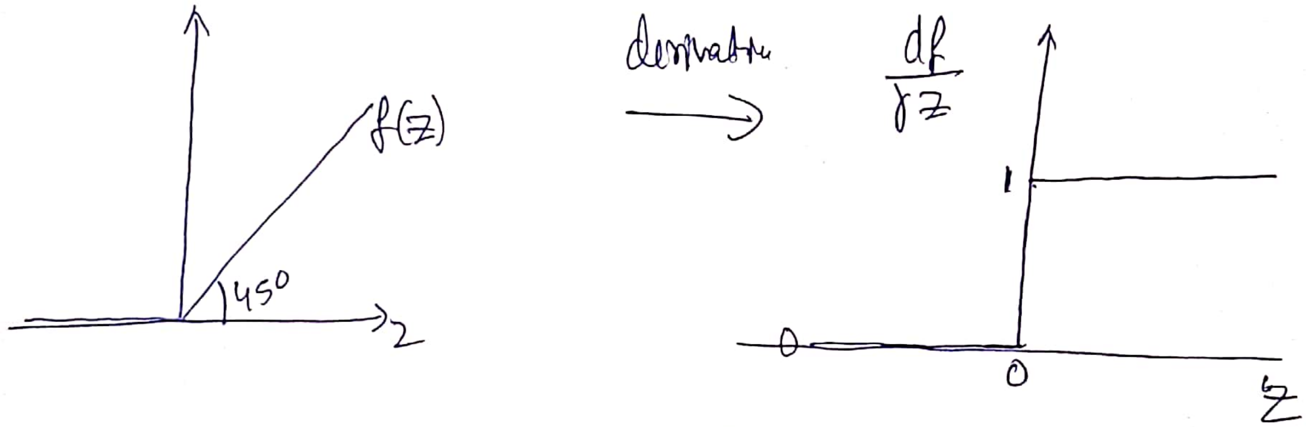
& whenever 'z' is greater than '0', its value simply will be "z"

This is basically a "45°" line

Computing derivative of this "ReLU function" is also very very simple.

The condition for a function to be used as an activation unit is that, it should be differentiable & it should be differentiated easily.

So if this "ReLU" function is $f(z)$ then its derivative looks like



for a 45° line, the slope is exactly equal to "1"

Derivative is not defined at "0", and everywhere else it is defined.

the derivative of this activation (ReLU) function is either "1" or "0"

$$\frac{d f_{ReLU}}{dz} \in \{0, 1\}$$

so here we will not be getting values like 0.2, 0.3, 0.1 & we are not multiplying them so that problem of vanishing gradient will not occurs. 15

—— o ——