## horizontal line

Activation Functions

27.01.2025

# Activation Functions

Each neuron forms a weighted sum of its inputs and passes the resulting scalar value through a function i.e. activation function / transfer function .

a = g(w1x1 + w2x2 + w3x3 - - - - - - - - -+ wnxn +b)

g is an activation function.

If activation is linear there’s no effect on output.

# Why are Activation Functions needed ?

To classify non linear data and complex patterns.



By forward propagation :

Z1 = w1a0 + b1

a1 = g(z1) = Z1 (linear activation)

a2 = g(Z2) = g(w2a1 + b2) = W2 (W1a0 + b1) + b2

= w2w1a0 + w2b1 + b2

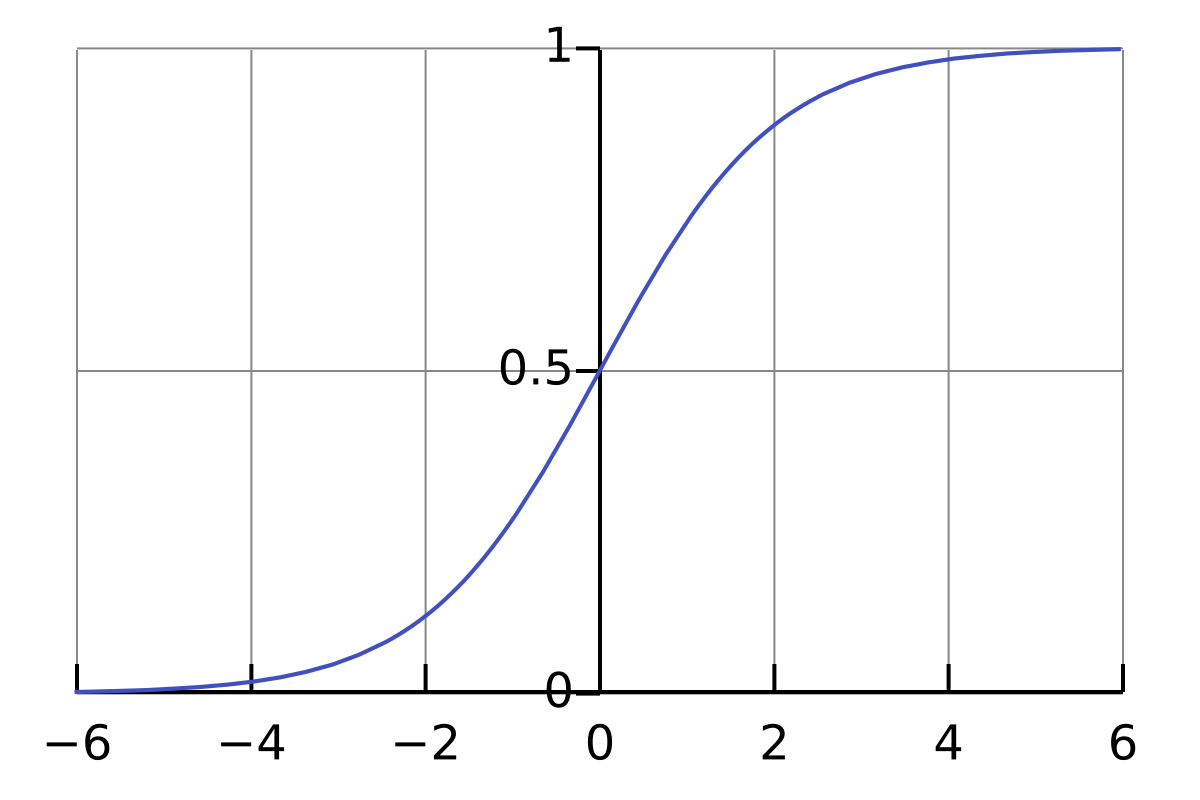
= w’a0 + b’

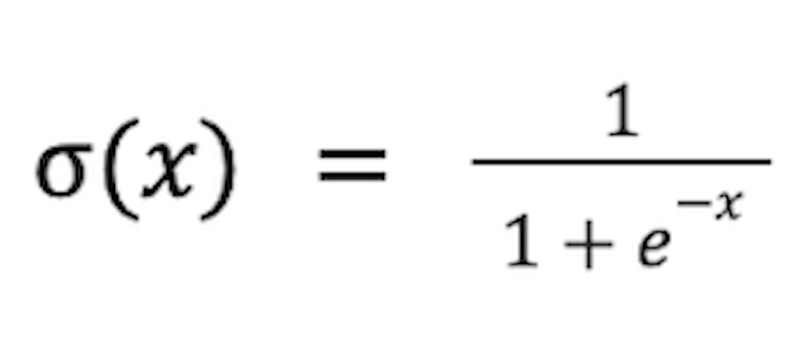
Degree of input and output is the same as 1 thus only linear data can be classified .

# Ideal Activation Function

1. Non linear
2. Differentiable (gd)
3. Computationally inexpensive
4. Zero centered (mean of activated outputs = 0)
5. Non saturating (should not bounded to a range)

# Sigmoid Activation Function

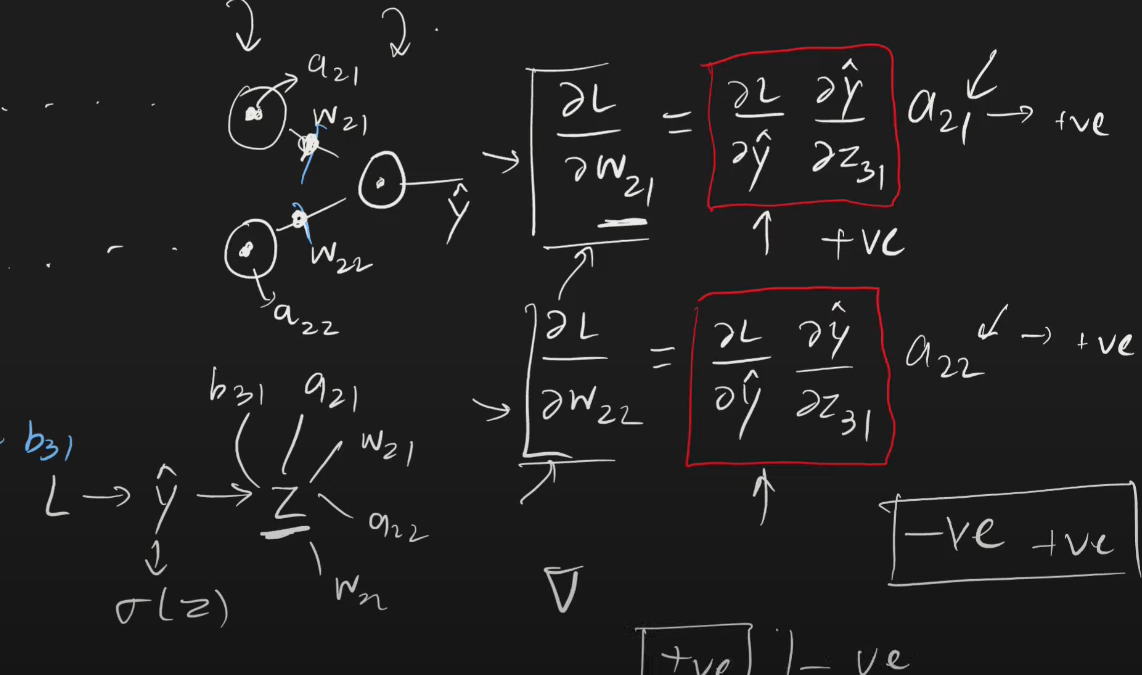


Advantages : 

1. [0,1] probability in output for binary classification.
2. Non - linear
3. Differentiable

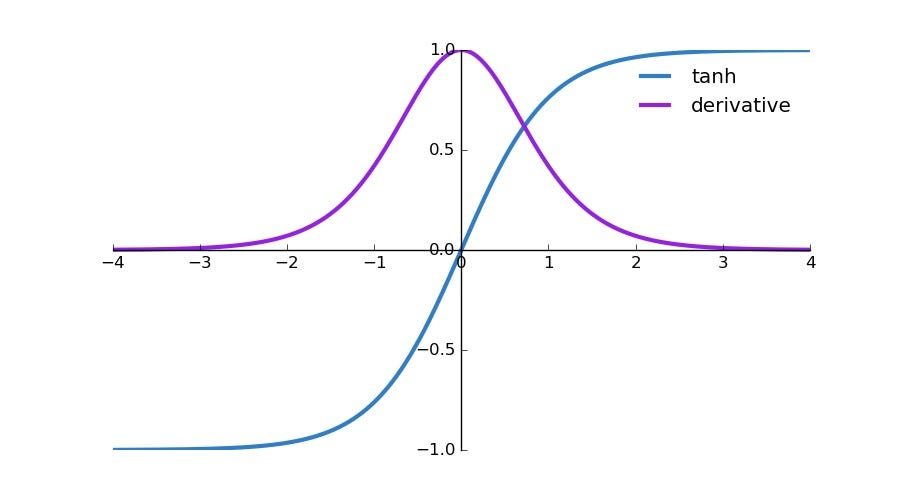
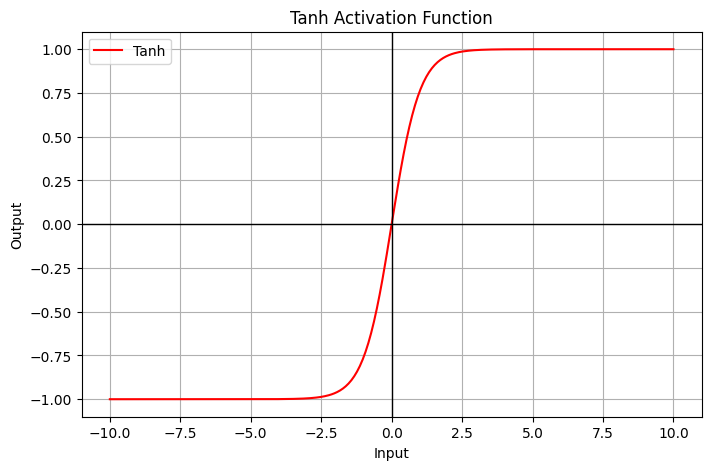
Disadvantages :

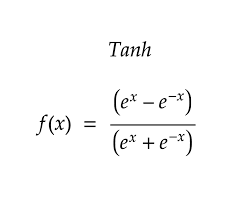
1. Saturating problem , input range [- , ] , output range [0,1] , causes Vanishing Gradient Problem in backpropagation .
2. Non Zero centered (mean not 0) , either weights increase or either all of them decrease , means there is restriction on training thus training gets slow.



1. Computationally expensive

# 2. Tanh Activation Function





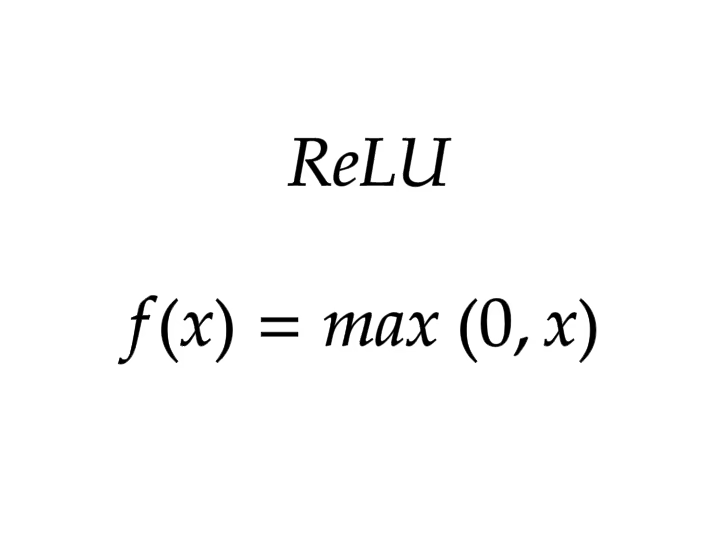
Advantages :

1. Non linear
2. Differentiable
3. Zero centered (training faster)

Disadvantages :

1. Saturating function (VGP)
2. Computationally expensive

# Relu Activation function



Advantages :

1. Non linear
2. Not saturated in +ve region
3. Computational inexpensive
4. Converge faster than sigmoid , tanh.

Disadvantages :

1. Differentiable
2. Non zero centered (Dying relu problem)

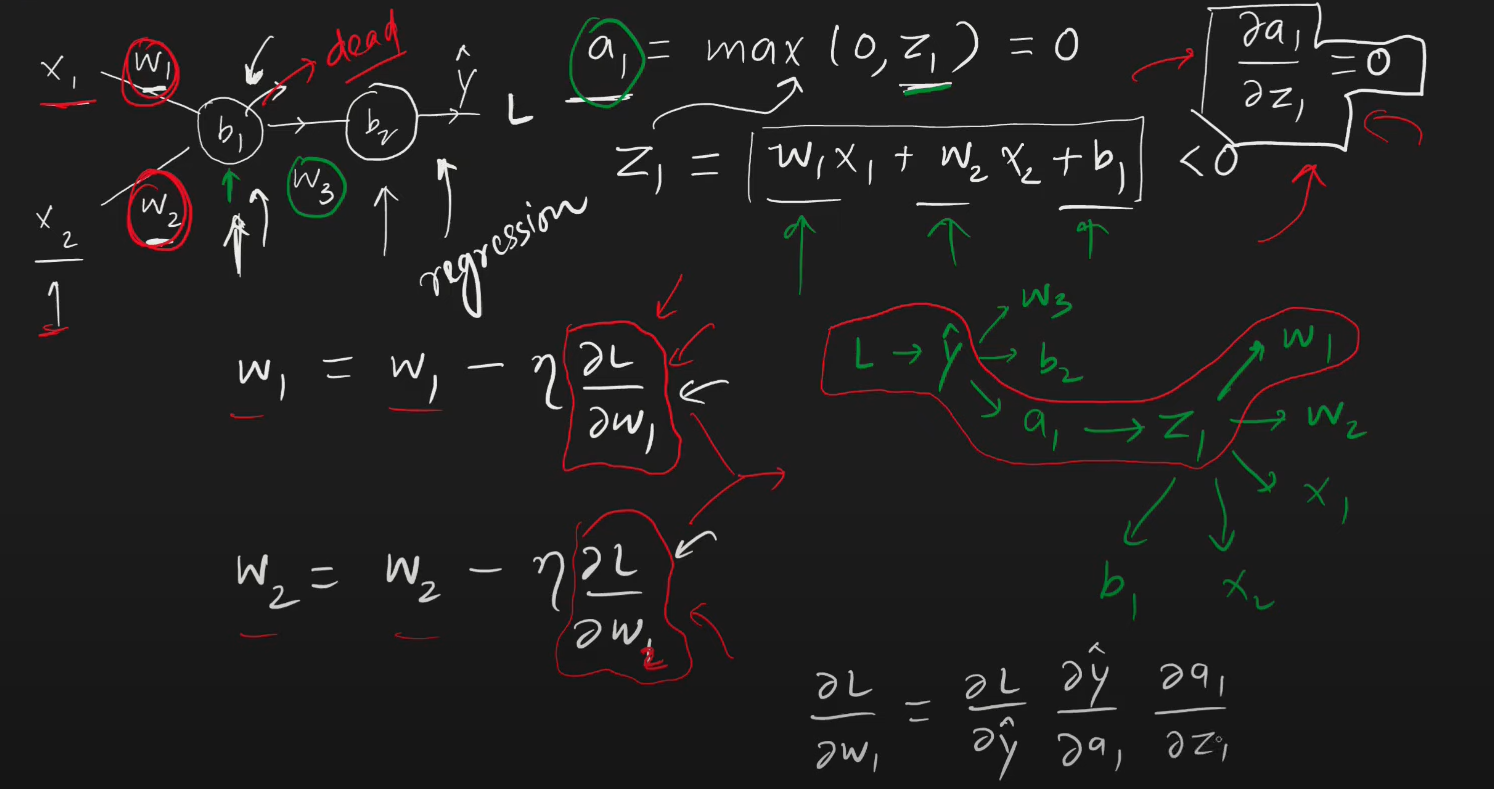
# 

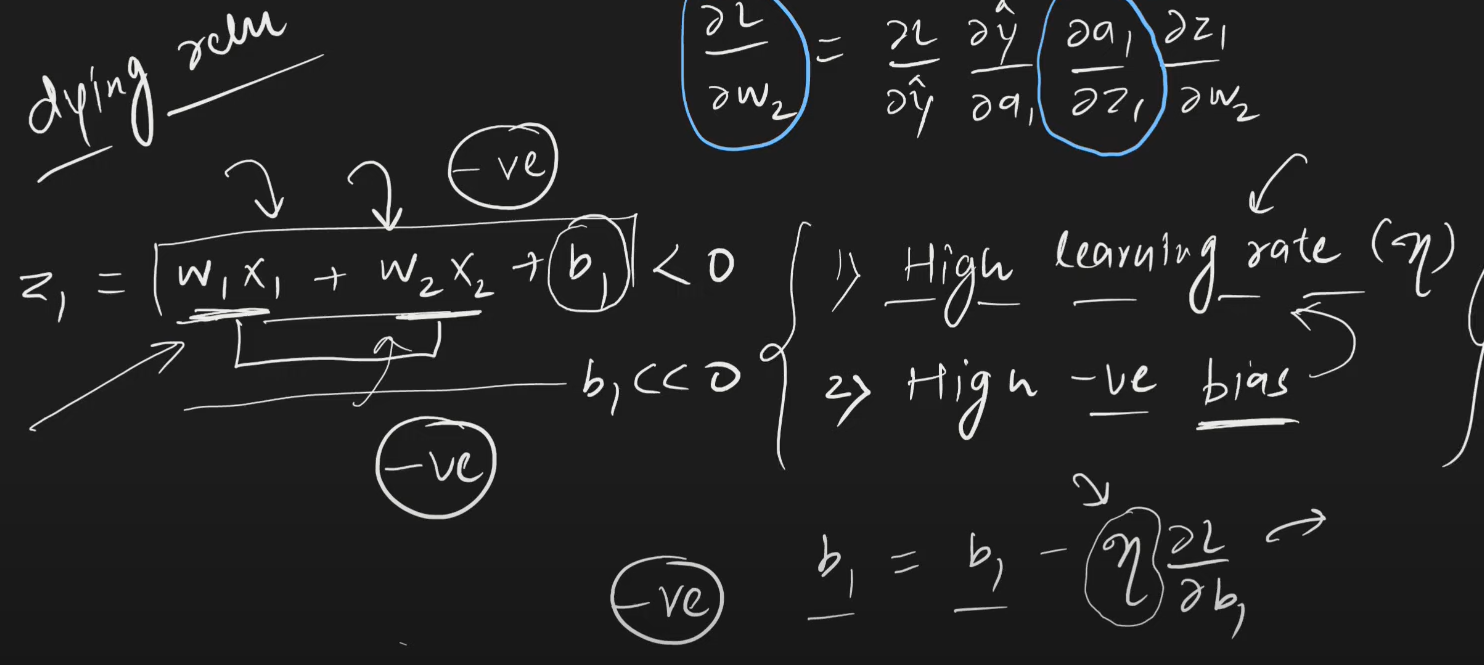
# Relu Variants :

## Dying Relu Problem :



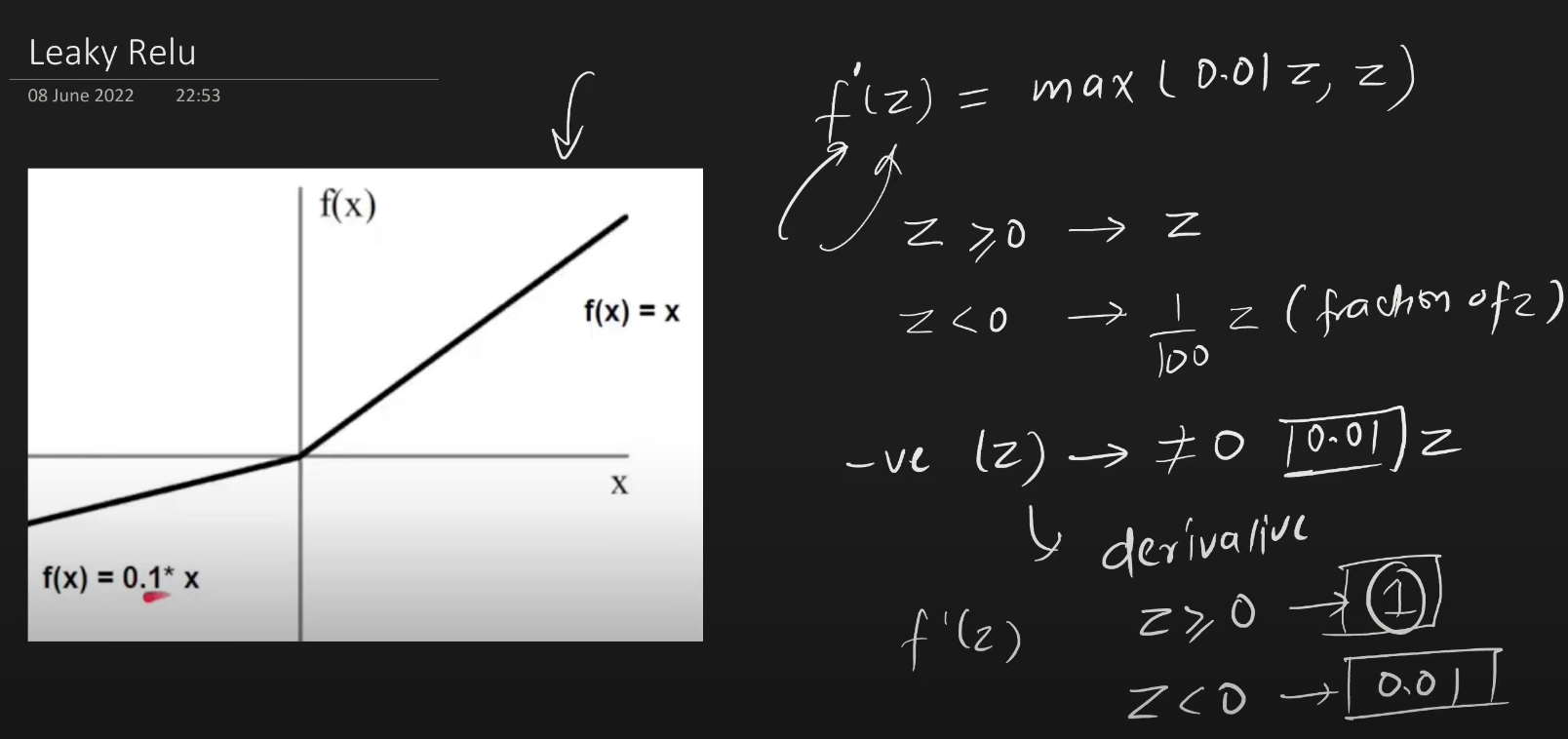
Assume b12 is a dead neuron , i.e. it gives 0 output for all inputs . Thus forever dead. Now the model cannot capture all the features.

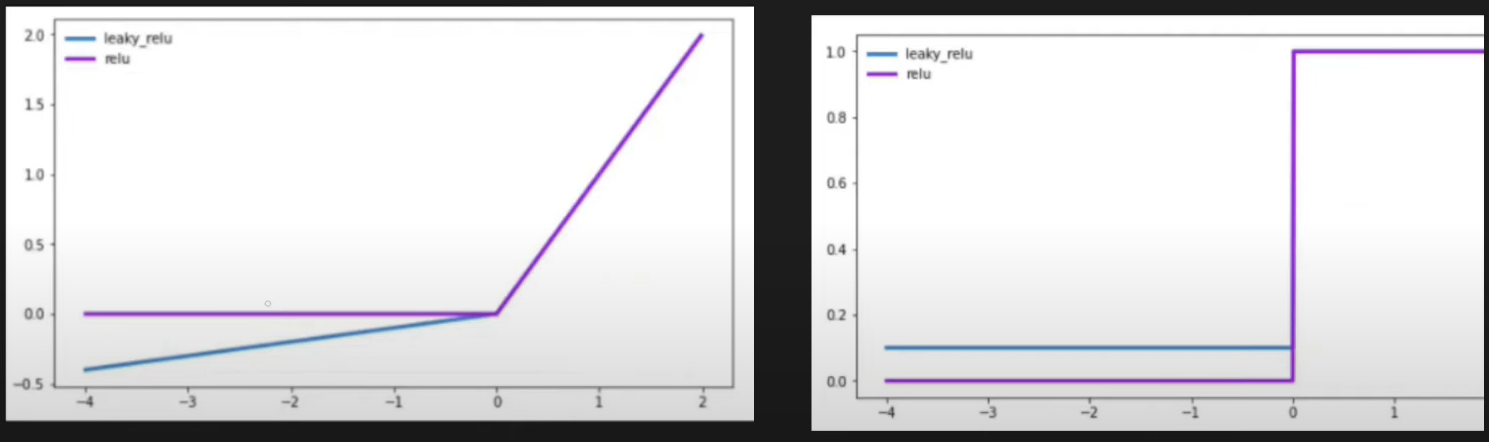


Solutions : 

1. Set low learning rate
2. Bias +ve value
3. Use relu variants instead of relu .

# Leaky Relu

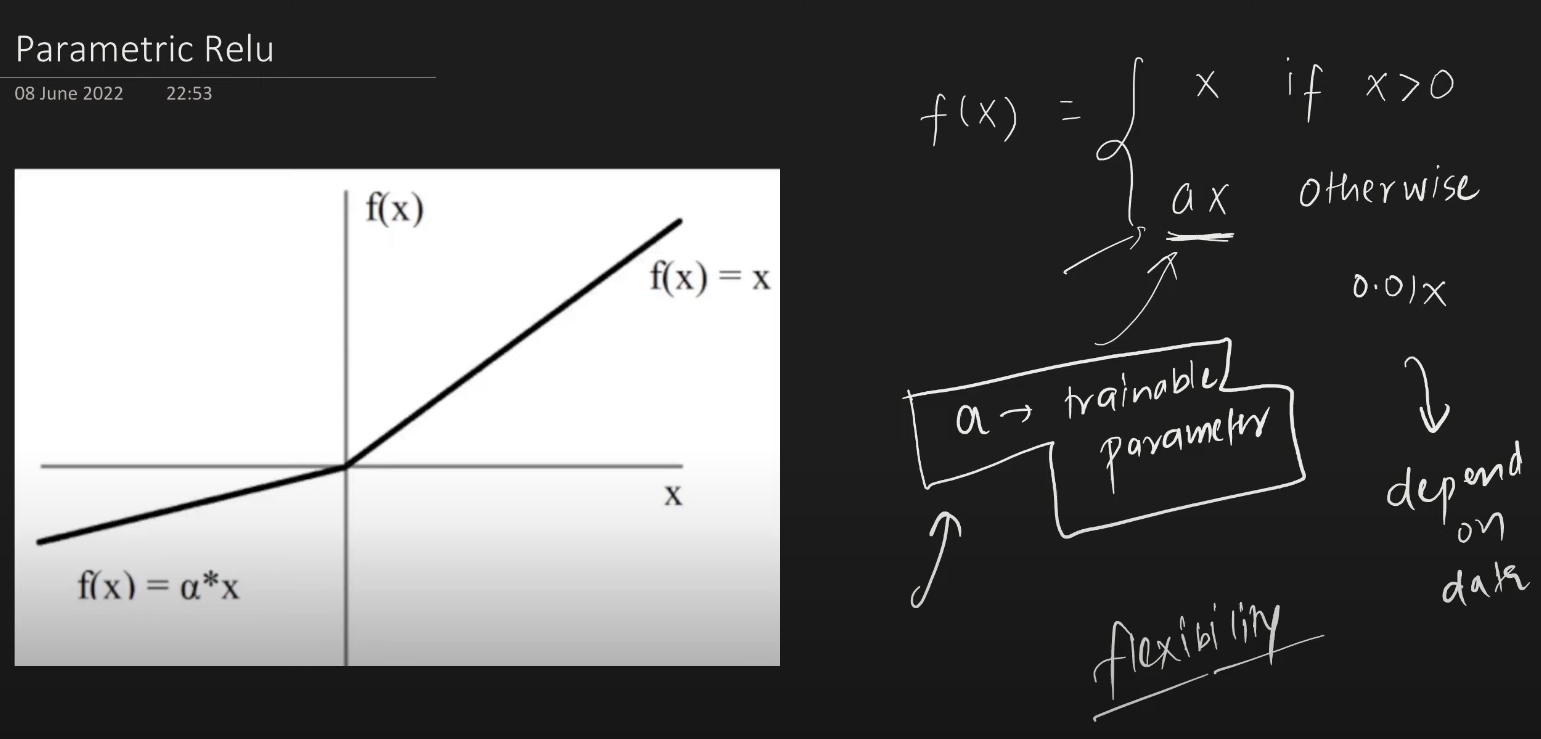




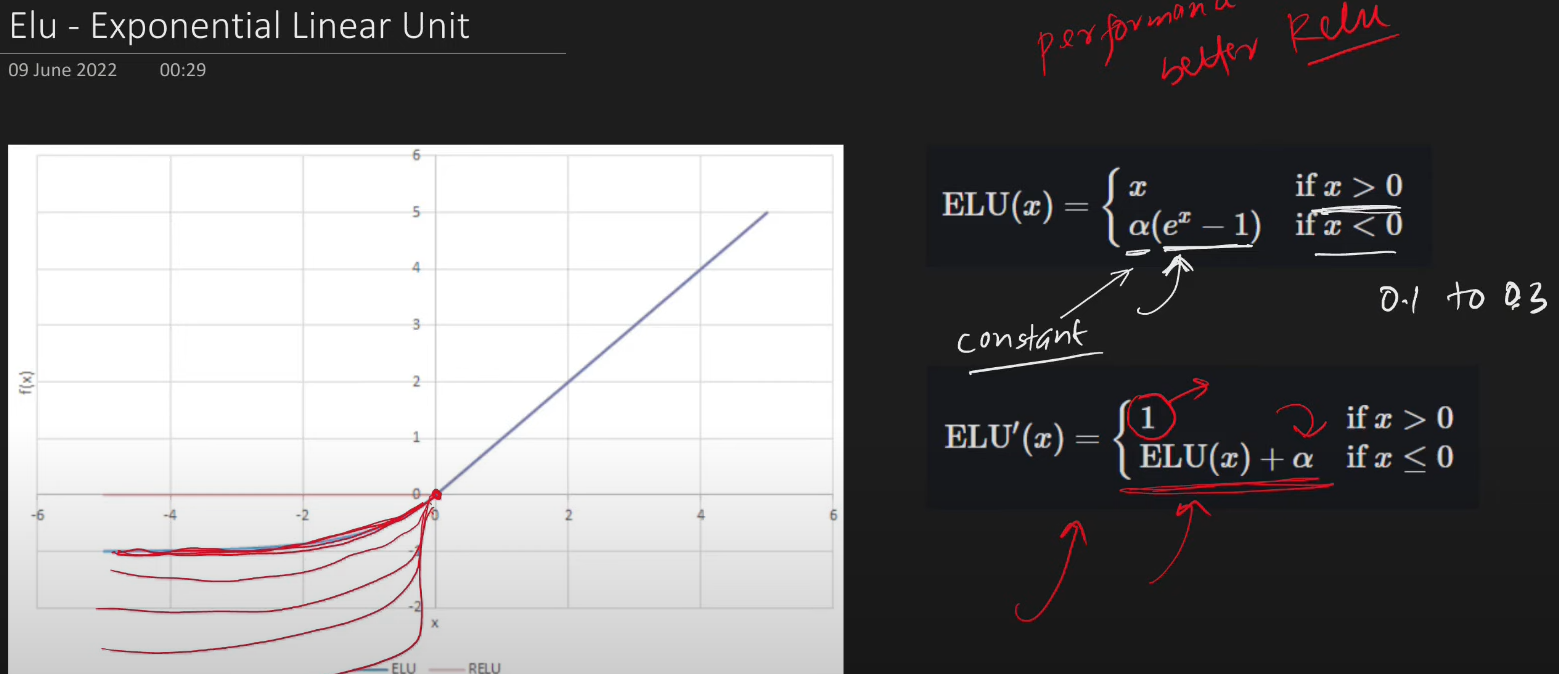
Advantages :

1. Non saturated (Unbounded)
2. Easily computed
3. No dying relu problem
4. Close to 0 centered

# 2. Parametric Relu



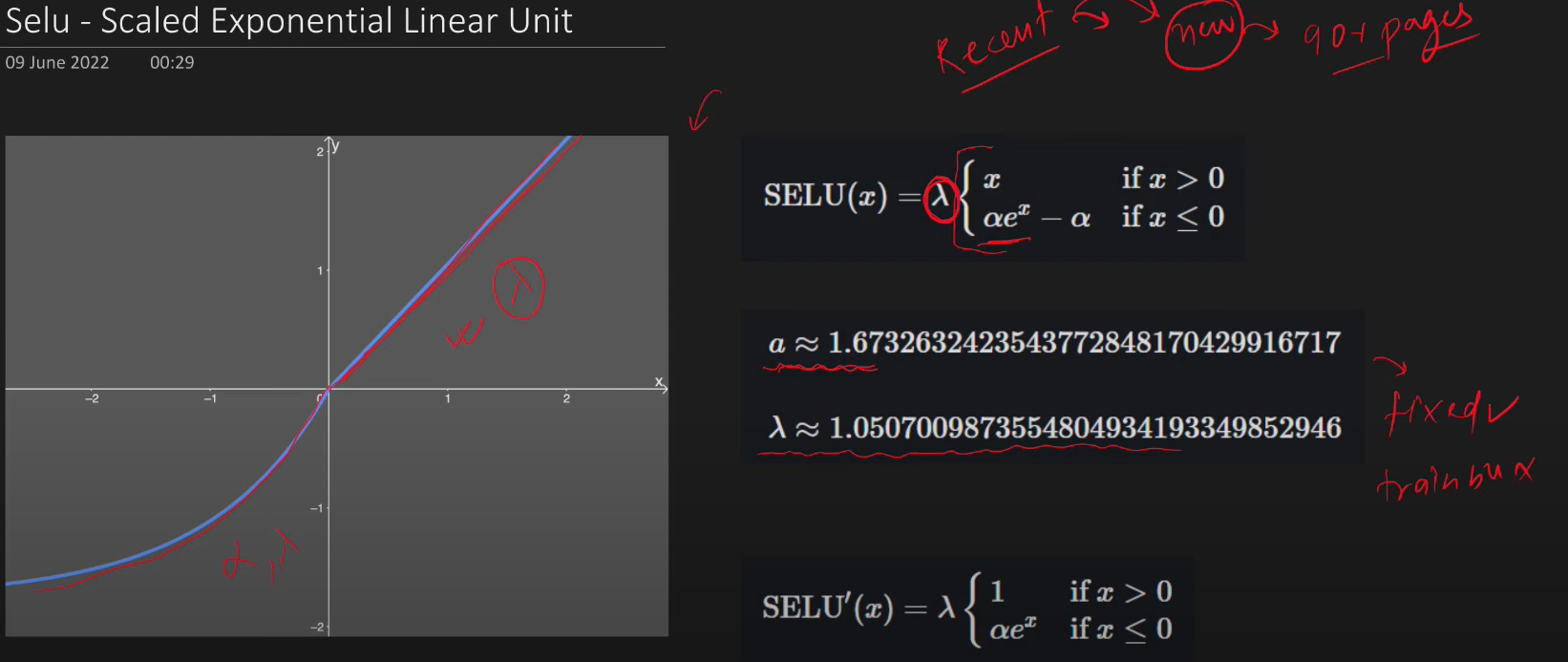
# 3. Elu (Exponential Linear Unit)



Advantages : Disadvantage :

1. Close zero centered (convergence faster) 1. Computationally
2. Better generalization expensive
3. Dying relu
4. Always continuous as well as differentiable

# 4. Selu (Scaled Exponential Linear Unit)



Advantages :

1. Self normalization
2. Converge faster