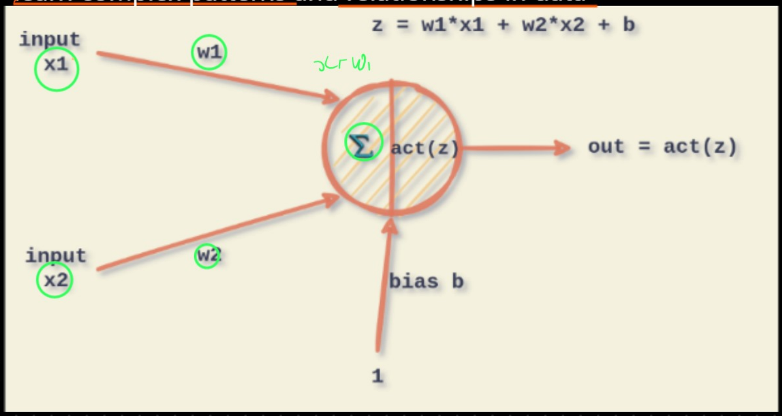
Activation Function

→ The activation function of a neuron defines the output of that neuron given its input.

→ It introduces non-linearities into the network, enabling it to learn complex patterns

→ Activation functions play a critical role in neural networks by introducing non-linearity into the model, allowing it to learn complex patterns and relationships in data



$$\boxed{y = \sigma(\Sigma w_i x_i + B)}$$

↓

Activation Function

Common Activation Functions

1. Sigmoid Functions
2. Hyperbolic Tangent Function (Tanh)
3. Rectified Linear Unit
4. Leaky ReLU
5. Exponential linear Unit (ELU)

① Sigmoid Function (Logistic Function)

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

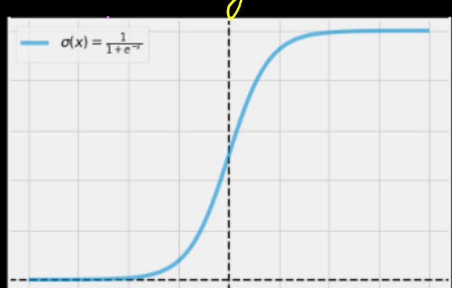
→ Range (0, 1)

Explaination

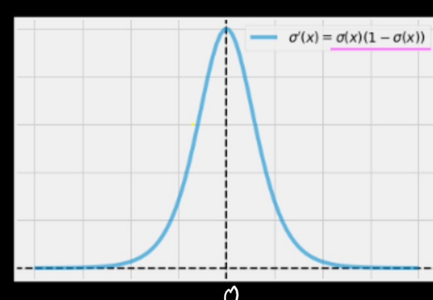
① The sigmoid function squashes the input values b/w 0 and 1.

② It is used in Binary Classification
↓
Output → probability

Sigmoid Function



Derivative →



Advantage

① Output values bound 0 and 1

② Clear prediction i.e very close 1 or 0

Disadvantage

① Function output is not zero centered

② Prone to gradient vanishing

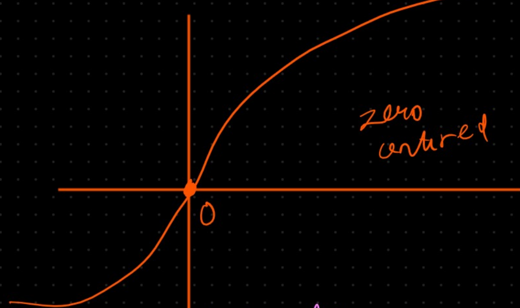
② Hyperbolic Tangent Function (Tanh)

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

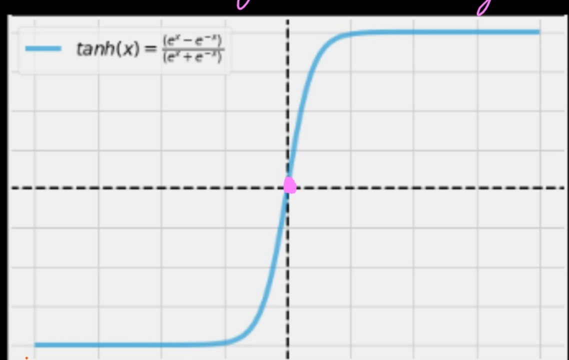
Range → $(-1, 1)$

Explanation →

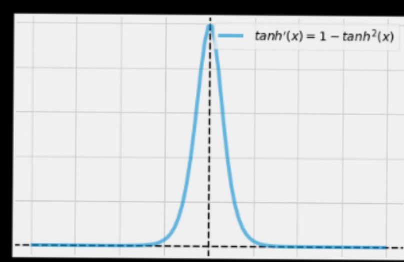
→ Tanh function is similar to the sigmoid function but squashes input values between -1 and 1, which makes it zero-centered. It's often used in hidden layers of neural networks.



Hyperbolic Tangent (tan h)



derivative



Advantage:

① zero centred → weight updation Efficient

Disadvantage

① Prone to Vanishing Gradient Problem

② Time complexity

③ Rectified Linear Unit (Relu)

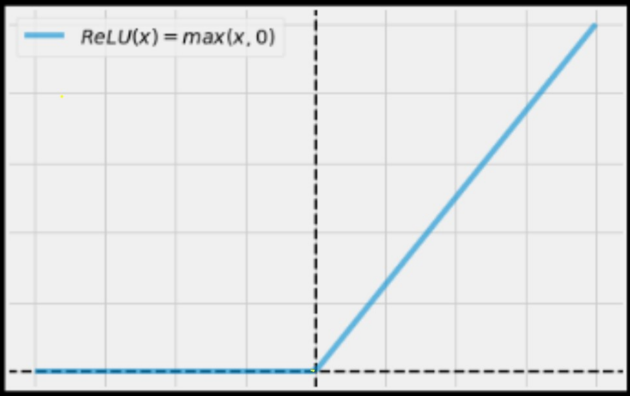
$$ReLU(x) = max(0, x)$$

Range : $[0, \infty]$

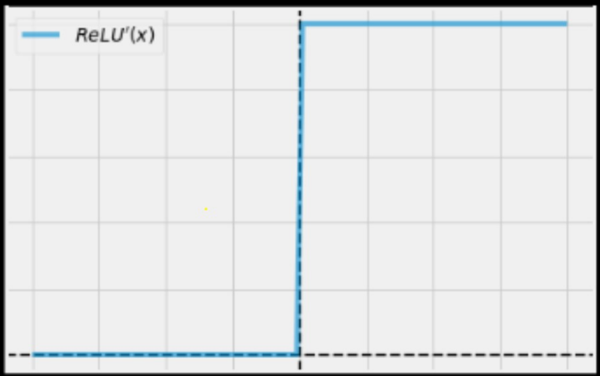
Explanation:

ReLU activation function replaces all negative values with zero and leaves positive values unchanged. It's the most commonly used activation function in deep learning due to its simplicity and effectiveness.

→ It help in overcoming the vanishing gradient problem.



derivative



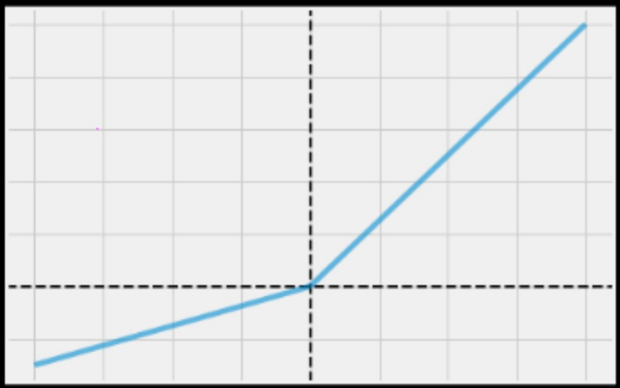
(4) Leaky Relu

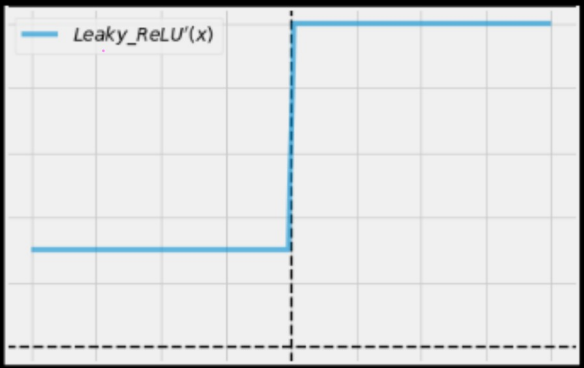
$$\text{Relu}(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha x, & \text{otherwise} \end{cases}$$

$\alpha \rightarrow$ small constant (0.01)

Explanation

Leaky ReLU is similar to ReLU but allows a small, non-zero gradient when the input is negative. It helps to mitigate the dying ReLU problem where neurons can become inactive during training.





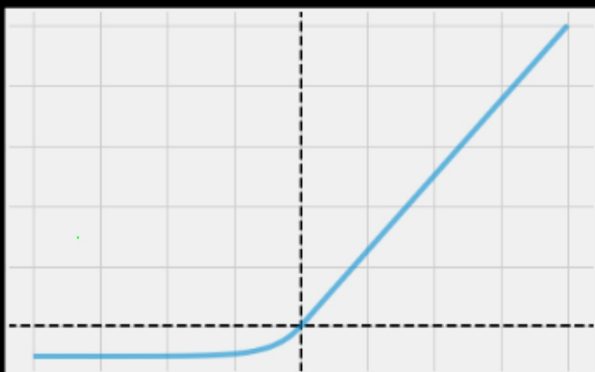
(5) Exponentetial linear unit (ELU)

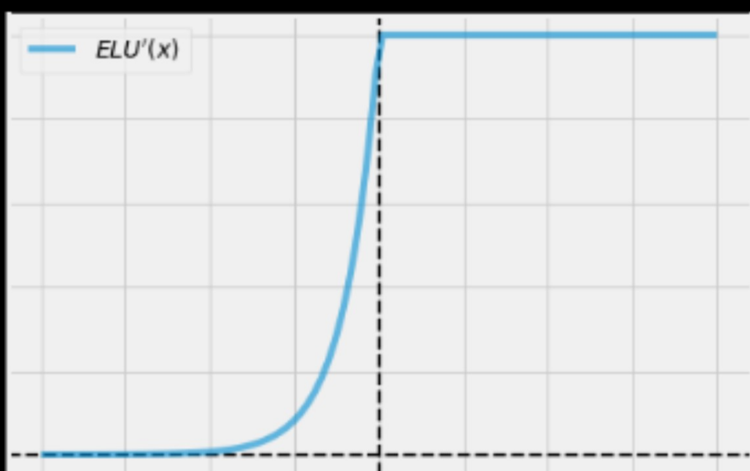
$$ELU = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^{x} - 1), & \text{otherwise} \end{cases}$$

Range $(-\infty, \infty)$

Explanation:

ELU is similar to Leaky ReLU but uses the exponential function for negative values, which allows it to have negative values with a smooth gradient.





ELU is also proposed to solve the problems of ReLU. Obviously, ELU has all the advantages of ReLU, and:

* No Dead ReLU issues
* The mean of the output is close to 0, zero-centered

One small problem is that it is slightly more computationally intensive. Similar to Leaky ReLU, although theoretically better than ReLU, there is currently no good evidence in practice that ELU is always better than ReLU.

Generally speaking, these activation functions have their own advantages and disadvantages. There is no statement that indicates which ones are not working, and which activation functions are good. All the good and bad must be obtained by experiments