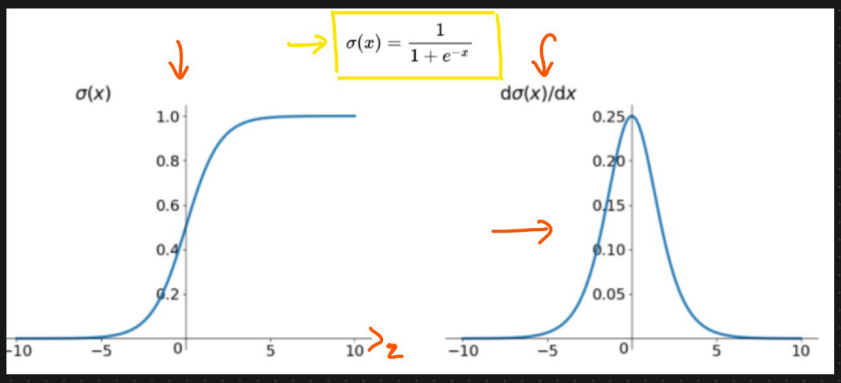
Activation Functions

Sigmoid, softmax are more prone to vanishing gradient as $\sigma_z \leq 0.25$

① Sigmoid Activation function [0 to 1]

$$z = \sum_{i=1}^{n} w^T x + b$$

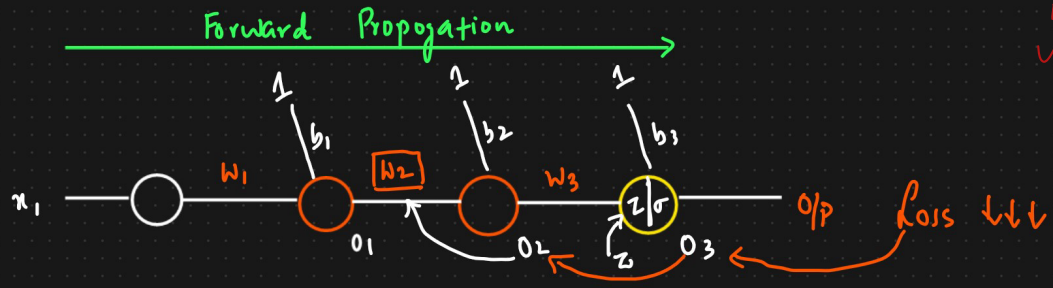


$\sigma(z) \Rightarrow$ 0 to 1

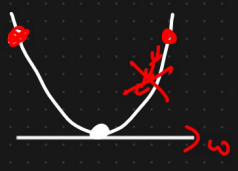
$\phi(z) \Rightarrow$

$\frac{\partial \sigma(z)}{\partial z}$ = 0 to 0.25

problem for vanishing Gradient.



Backward Propogation.



$$W_{2new} = W_{2old} - \eta \left[\frac{\partial h}{\partial W_{2old}} \right]$$

0.001

0.00001

$\Rightarrow W_{2new} \approx W_{2old} \Rightarrow$

$$\frac{\partial h}{\partial W_{2old}} = \frac{\partial h}{\partial O_3} * \frac{\partial O_3}{\partial O_2} * \frac{\partial O_2}{\partial W_2}$$

0.20 * 0.01 *

Let $z = (O_2 * W_3) + b_3$

$$\frac{\partial O_3}{\partial O_2} = \frac{\partial(\sigma(z))}{\partial z} * \frac{\partial z}{\partial O_2}$$

[0 to 1]

$$= [0 - 0.25] * \frac{\partial [(O_2 * W_3) + b_3]}{\partial O_2}$$

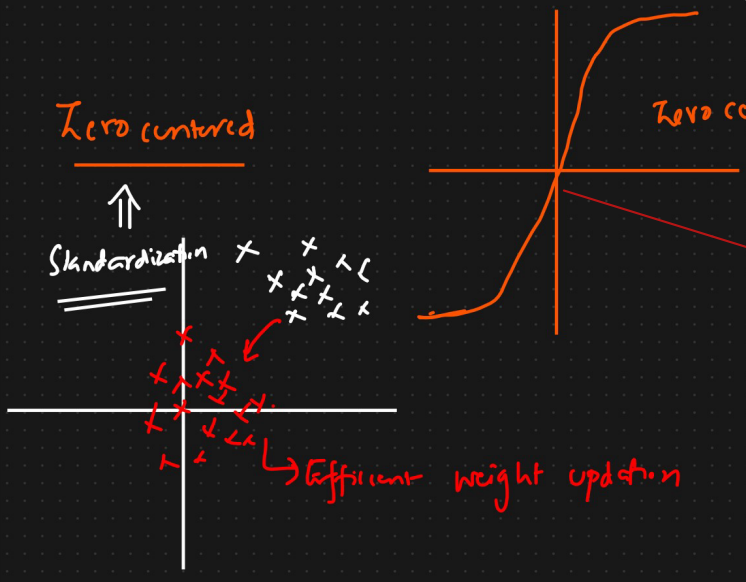
$$= [0 - 0.25] * W_3 \Rightarrow \text{Small value} \Rightarrow$$

Advantages

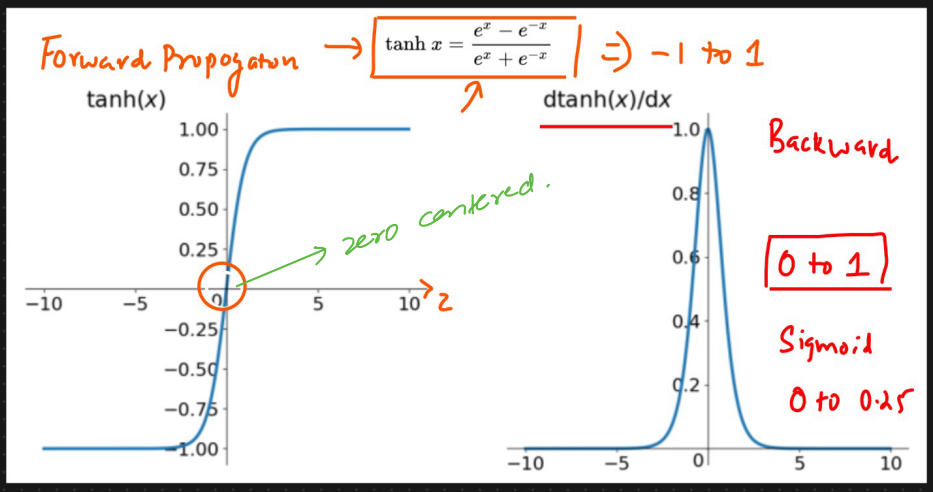
1. Binary classification Suitable.
2. Clear prediction i.e very close 1 or 0

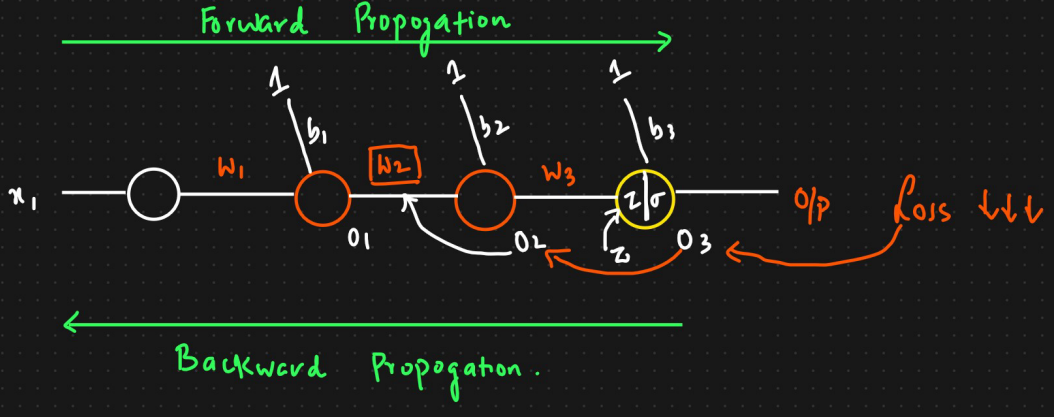
Disadvantages

1. Prone to vanishing gradient Problem.
2. Function output is not Zero centred ⇒ Efficient weight updation
3. Mathematical operation are relatively time Consuming



② Tanh Activation Function





$$\frac{\partial h}{\partial w_{2old}} = \frac{\partial h}{\partial o_3} * \frac{\partial o_3}{\partial o_2} * \frac{\partial o_2}{\partial w_2}$$

$0.20_{11} * 0.01 \times$

Let $z = (o_2 * w_3) + b_3$

$$\frac{\partial o_3}{\partial o_2} = \frac{\partial(\tanh(z))}{\partial z} * \frac{\partial z}{\partial o_2}$$

[0 to 1]

$$= [0-1] * \frac{\partial[(o_2 * w_3) + b_3]}{\partial o_2}$$

$$= [0-1] * w_3 \Rightarrow \text{Small value} \Rightarrow$$

Advantages

① Zero Centric ⇒ Weight updation is Efficient

Disadvantages

① Prone to Vanishing Gradient Problem

② Time complexity

③ Relu Activation Function

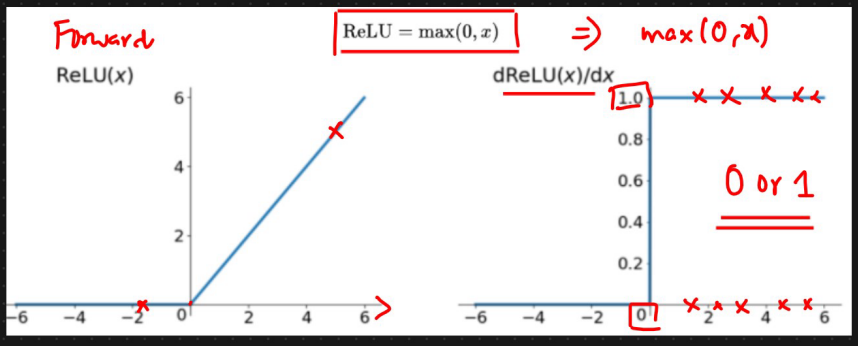
[Rectified Linear Unit].

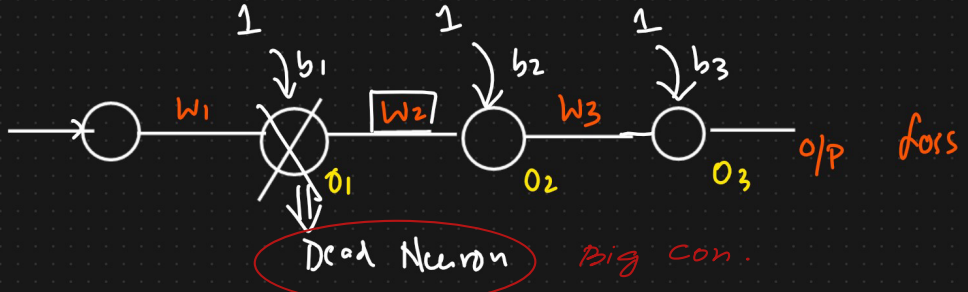
Tanh ⇒ 0 to 1

Sigmoid ⇒ 0 to 0.25

Forward

$\text{ReLU} = \max(0, x)$ ⇒ $\max(0, x)$





$$\frac{\partial h}{\partial w_{2old}} = \frac{\partial h}{\partial o_3} * \boxed{\frac{\partial o_3}{\partial o_2}} * \frac{\partial o_2}{\partial w_2}$$

0.20_{11} * 0.01 ×

Let $z = (o_2 * w_3) + b_3$

$$\frac{\partial o_3}{\partial o_2} = \boxed{\frac{\partial (\text{Relu}(z))}{\partial z}} * \frac{\partial z}{\partial o_2} \qquad [0 \text{ to } 1]$$

$$= [0 \text{ or } 1] * \frac{\partial [(o_2 * w_3) + b_3]}{\partial o_2}$$

$$= [\overset{-ve}{0} \text{ or } \overset{+ve}{1}] * w_3 \Rightarrow \text{Small value} \Rightarrow$$

Derivative of

If Relu output is $\boxed{1}$ $\Rightarrow$ Weight updation will happen

If Relu output is $\boxed{0}$ $\Rightarrow$ Dead Neuron

$$w_{2new} = w_{2old} - \eta \boxed{\frac{\partial h}{\partial w_{2old}}} \Rightarrow 0$$

If Derivative of Relu(z) is 0

$$\boxed{w_{2new} \approx w_{2old}} \Rightarrow \text{Dead Neuron}$$

If $z = +ve$ $\qquad \frac{\partial \text{Relu}(z)}{\partial z} = 1$

If $z = -ve$ $\qquad \frac{\partial \text{Relu}(z)}{\partial z} = 0$

Advantages

① Solving Vanishing Gradient Problem

② Max(0, x) → Calculation is superfast. The Relu function has a linear Relationship

③ It is much faster than Sigmoid or Tanh.

Disadvantages

① Dead Neuron

② Relu function O/P

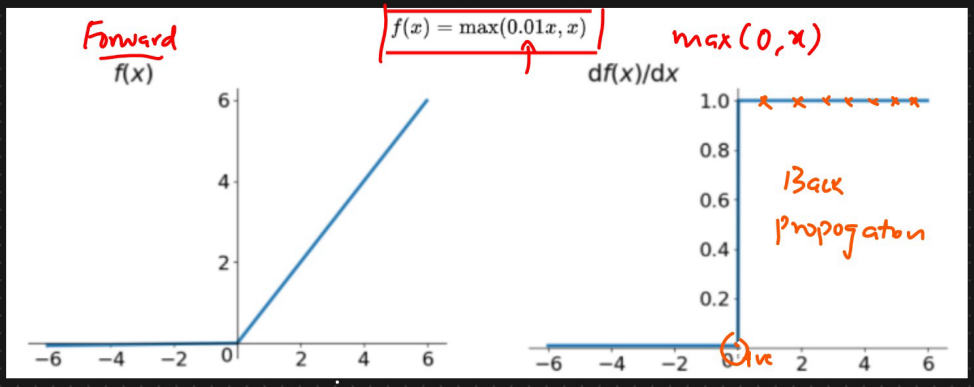
(0, x) ⇒ 0 or +ve number

⇓

It is not zero centric

④ Leaky Relu And Parametric Relu

max(αx, x) → hyperparameter α = alpha = 0.01, 0.02, 0.03



Relu → Dead Neuron → Dead Relu Problem

Advantages

① Leaky Relu has all the advantage of Relu

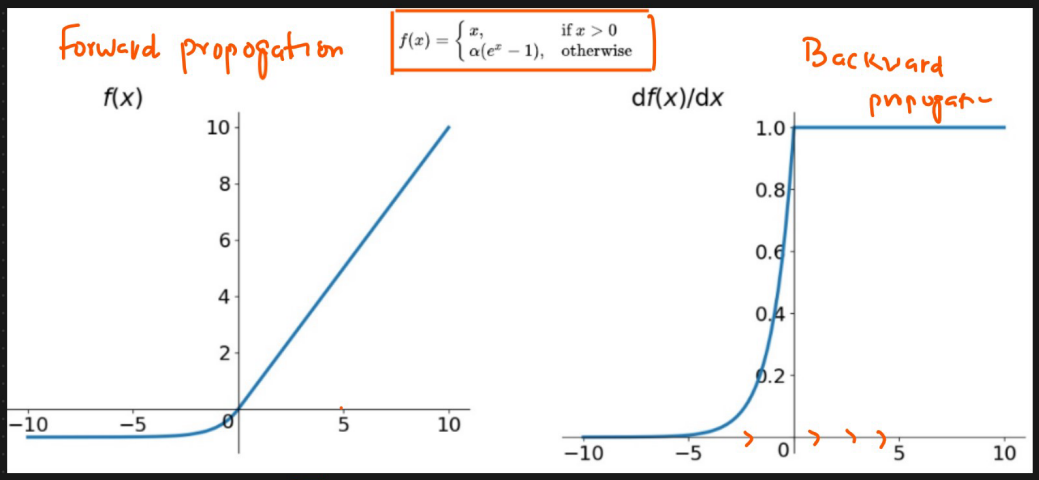
② It removes the Dead Relu Problem

Disadvantage

① It is not zero centric

⑤ ELU (Exponential Linear Units)

$\alpha(e^x - 1)$.



[It is used to solve Relu problems]

Advantages

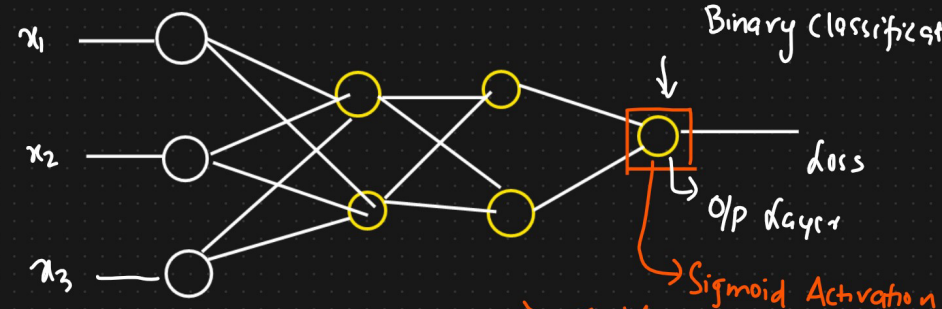
① No Dead Relu Issues

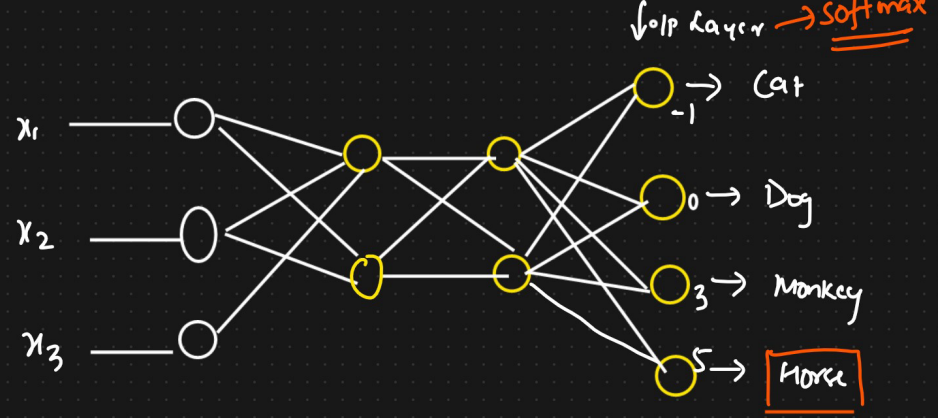
② Zero centered

Disadvantage

1) Slightly more computationally Intensive.

⑥ Softmax Activation function [Multiclass classification Problem]





Softmax Activation

$$\text{Softmax} = \frac{e^{y_i}}{\sum_{K=0}^{n} e^{y_K}}$$

$y_i = 0 * w + b$

Softmax ⇒ Cat $= \frac{e^{-1}}{e^{-1+0+3+5}} = 0.00033$

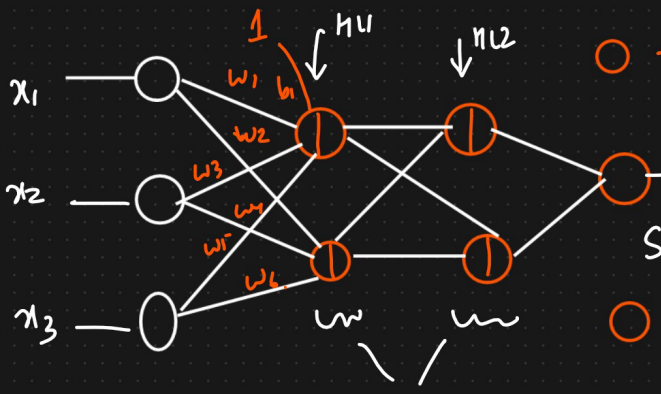
Dog $= \frac{e^{0}}{e^{-1+0+3+5}} = 0.0024$

Monkey $= \frac{e^{3}}{e^{-1+0+3+5}} = 0.0183$

Horse $= \frac{e^{5}}{e^{-1+0+3+5}} = 0.1353$

$Pr(\text{Horse}) = \frac{0.1353}{0.00033 + 0.0024 + 0.0183 + 0.1353} \approx 86\%$

⑦ Which Activation Function To Use When ?



In short :

Hidden Layer — ReLU

Output Layer (classification) — Sigmoid / Softmax

Output Layer (Regression) — Regression Activator

✅ Activation Function Decision Chart (with Linear)

Situation / Layer Type	Best Activation Function
✅ Hidden Layers (default)	**ReLU**
Hidden Layers (risk of dead neurons)	**Leaky ReLU**
Hidden Layers (RNNs / time series)	**tanh** Imp.
Output – Binary Classification	**Sigmoid**
Output – Multi-class Classification	**Softmax**
Output – Regression (real number prediction)	**Linear**
Advanced models (deep / experimental)	**Swish / ELU**