Made with Goodnotes

1) ReLU $\Rightarrow$ Rectified Linear Unit

$$\boxed{f(x) = ReLU = \max(0, x)}$$

$\rightarrow$ Formula

$$f(x) = \begin{cases} x, & x > 0 \\ 0, & x < 0 \end{cases} \approx \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Both defn of ReLU are same i.e.

$$\max(0, x) \approx \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

why?

Because if $x > 0$, $\max(0, x) = x$

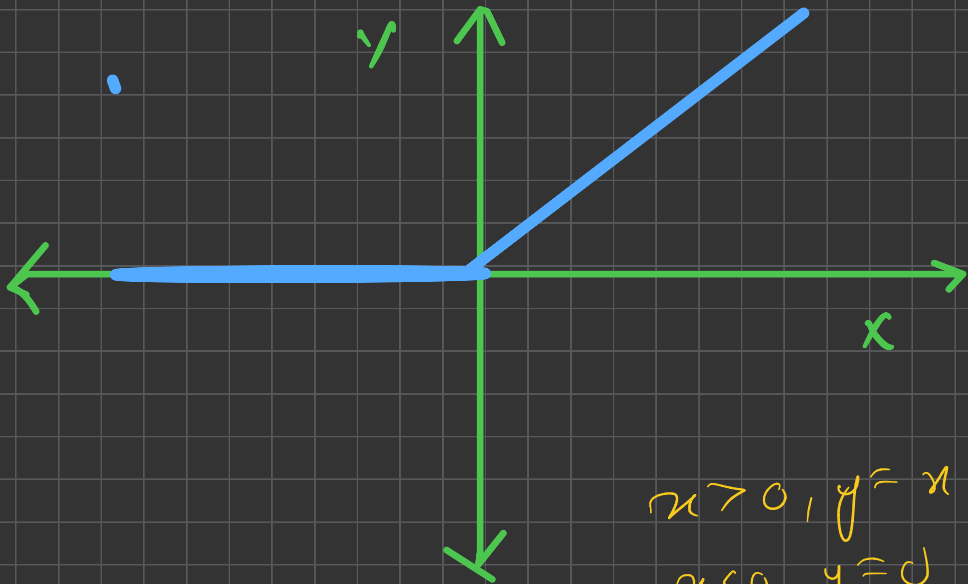
if $x < 0$, $\max(0, x) = 0$

for +ve input linear graph, $y = x$

-ve input, constant of. $y = 0$

$\Rightarrow$

Made with Goodnotes



$x > 0, y = x$

$x < 0, y = 0$

Q Have we introduced Non-linearity in this fun^n?

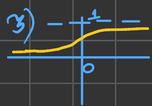
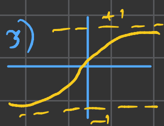
A- Non-linearity is there, as when $x > 0$, we maintain linearity
$x < 0$, we do 0,
so Linear + Non-lin. = Non-linear

Q Does squashing occur?

A- -ve signal → Block (some squash occur)
+ve signal → passed as it is.

Made with Goodnotes

ReLu is not much abt derivⁿ but more about obs. & defⁿ.

Actⁿ Funⁿ	Formula	Key char.	Pros	Cons	Use case
Sigmoid	$\sigma(z) = \frac{1}{1+e^{-z}}$	1) output b/w ⇒ (0,1) 2) S-shape curve 3) 	1) Smooth curve (easily interpretable) 2) Outputs are represented as probabilities	1) vanishing gradients (slow learning) 2) Not zero-centred.	1) classification problems 2) older models
tanh	$\tanh(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$	1) output -1 & 1. 2) S-shape curve 3) 	1) Smooth curve. (easily interpretable) 2) funⁿ is zero centric	1) Prone to vanishing gradients	1) Hidden layers in RNN it is used
ReLu (Rectified linear unit)	$f(z) = \max(0, z)$	1) if $z>0$, $y=z$ (linear) 2) if $z<0$ (neuron off)	1) Fast computation 2) No-vanishing gradient for +ve value	1) Dying Neuron problem. 2) Not-smooth	1) most modern networks hidden layers have this Actⁿ funⁿ (like transformers)

Made with Goodnotes