Activation function 2

25 September 2023 07:07

Input	Hidden	output
✗	✓	✓

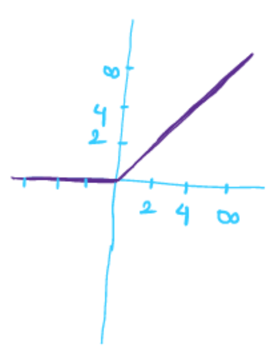
Input	$-\infty$ to ∞	$-\infty$ to ∞
output	0 to 1	-1 to 1
Deriva	0 to 0.25	0 to 1

proba

1

RELU

→ Rectified linear unit



$\boxed{max(y,0)}$

$max(2,0) = 2$

$max(4,0) = 4$

$max(-2,0) = 0$

$max(-4,0) = 0$

Input	$-\infty$ to ∞
output	$max(y,0)$
Derivati	are out

Relu is the perfect fit for the hidden layers.

Dying Relu problem.

Once a neuron is dead it is forever dead.

$x \rightarrow \bigcirc \xrightarrow{w_1} \bigcirc \xrightarrow{w_2} \bigcirc \xrightarrow{w_3} \bigcirc \rightarrow \hat{y}$

$$w_{new} = w_{old} - \alpha \frac{\partial L}{\partial w_{old}}$$

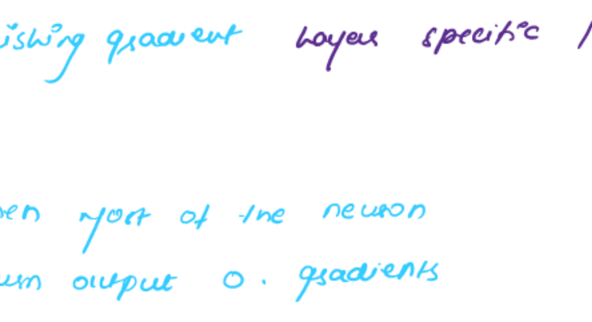
$$\frac{\partial L}{\partial w_{old}} = \frac{\partial L}{\partial O_{31}} \times \underbrace{\frac{\partial O_{31}}{\partial O_{21}}}_{\rightarrow 0} \times \frac{\partial O_{21}}{\partial O_{11}} \times \frac{\partial O_{11}}{\partial w_{11}}$$

$= 0.8 \times 0 \times 0.4 \times 0.3$

$= 0.$

$$w_{new} = w_{old} - \alpha \underbrace{\frac{\partial L}{\partial w_{old}}}_{\rightarrow 0}$$

$\boxed{w_{new} = w_{old}}$



$[0] + [w]t[w]$

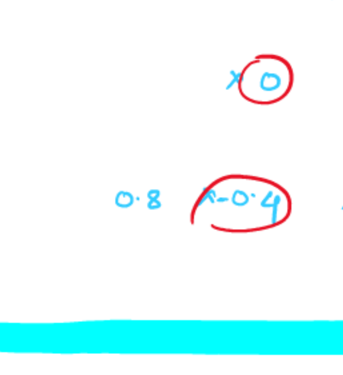
Dying relu. Neuron specific problem.

vanishing gradient layer specific problem.

When most of the neuron return output 0. gradients fail to flow back during back propagation ultimately large part of Neural network become inactive.

1. High learning rate.
2. High negative bias.

Leaky Relu.



$max(2, 0.1x)$

$max(4, 0.1 \cdot 2) = 4$

$max(-4, \underline{-4 \times 0.1})$

$=$

$= -0.4$

$$\frac{\partial L}{\partial O_{31}} \times \frac{\partial O_{31}}{\partial O_{21}} \times \frac{\partial O_{21}}{\partial O_{11}} \times \frac{\partial O_{11}}{\partial w_{11}}$$

$\times 0 \times \quad = $ Result 0.

$0.8 \times -0.4 \times 0.04 \times 0.01 = $ Result generated.

PRELU

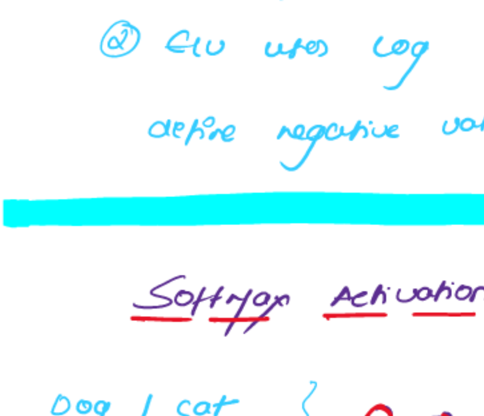
→ Parametric Relu.

α = trainable parameters.

this is an another variant of Relu aims to solve dying Relu problem as gradient becoming zero.

ELU

→ Exponential linear unit



$$f(x) = \begin{cases} x & \text{value} > 0 \\ a(e^{x}-1) & < 0 \end{cases}$$

1. No dying Relu problem.
2. Elu uses log curve to define negative values.

Softmax Activation fn.

Dog / cat

sigmoid

> 0.5

O → Dog	10	0.4 ✓
O → cat	20	0.1
O → lion	30	0.2
O → tiger	40	0.3

P=1

$$\frac{e^{z}}{\sum_{i=1}^{k} e^{z}} = \frac{e^{dog}}{e^{dog} + e^{cat} + e^{lion} + e^{tiger}}$$

classes

Softmax is used in case of multiclass classification problem. (output layer)

Swish.

→ google

Image classification.

$$f(x) = \frac{x}{1+e^{-x}}$$

$\boxed{H > 40}$

Hidden layer.

sigmoid

tanh.

Relu.

Leaky Relu.

PRelu.

Elu.

Output layer.

Regression.	Linear Activation fn.	$-\infty$ to ∞.	
Binary classi	sigmoid	0.5>	1, 0.
Multiclass	Softmax	no. of neurons = no. of classes.	

sig / tan	vgp	Layer specific
Relu.	DRP.	Neuron specific