

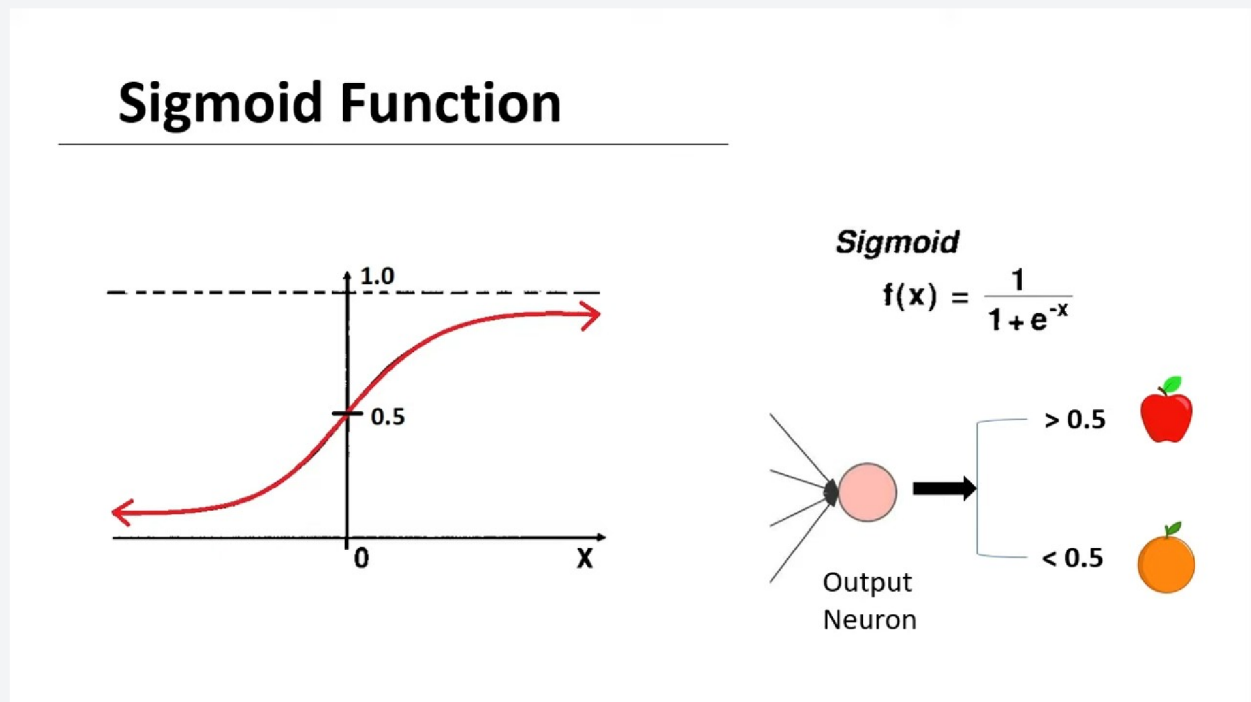


8 notes 7 screenshots

03:08



Sigmoid function



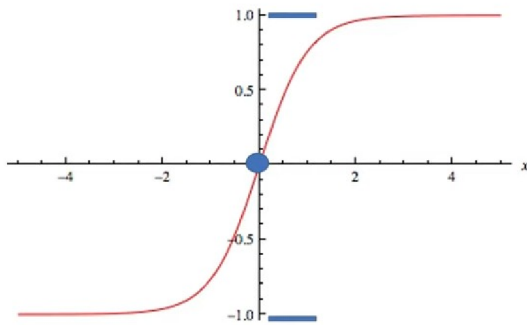
26 minutes ago

03:43



Tanh function

## Tanh Function



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

$$W = W - \alpha * \frac{\partial \text{cost}}{\partial W}$$

We need to take derivative of activation functions

$$\frac{\partial \tanh}{\partial x}$$

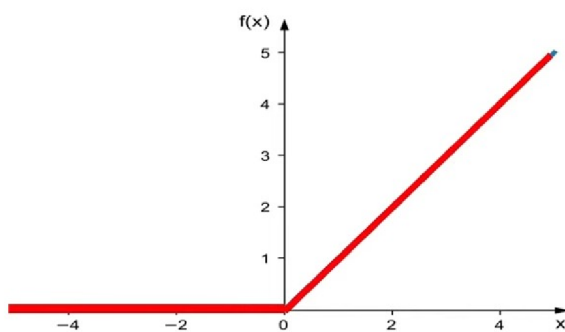
16 minutes ago

06:09



ReLU method if the derivative of activation function is low which will lead to slow learning

## ReLU (Rectified Linear Unit)



Piece-wise Linear

*Advantages of both linear and non-linear property*

$$f(x) = \max(0, x)$$

Overcome the Vanishing Gradient Problem

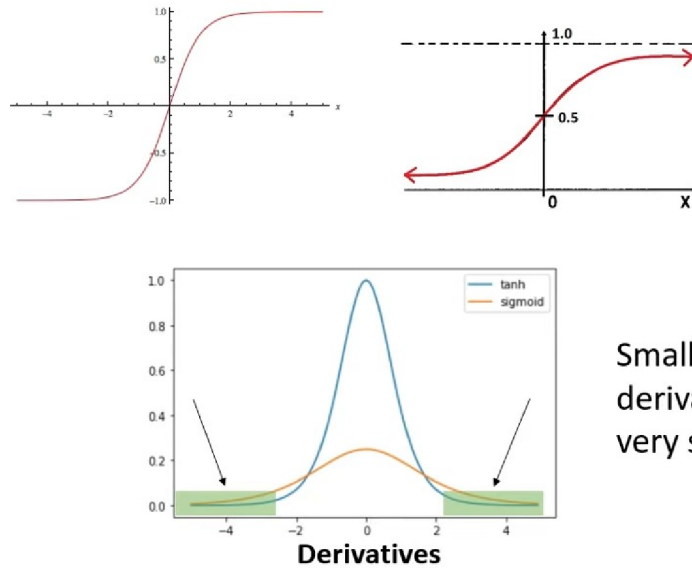
$$\frac{\partial f(x)}{\partial x} = \begin{cases} 1, & x > 0 \\ 0, & x < 0 \end{cases}$$

14 minutes ago

05:09



To resolve this problem we use RELU.



Vanishing Gradient Problem

11 minutes ago

07:45



We can use the sigmoid function for binary classification only. For multi-class classification, we can use the softmax function.

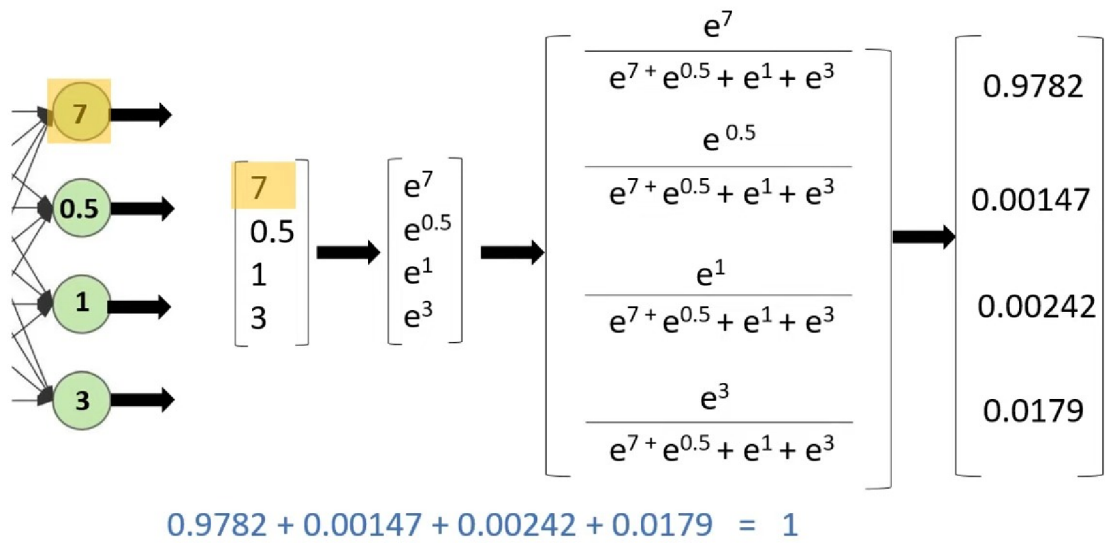
8 minutes ago

08:17



The output of softmax function and converting the value in the form of probability

## Softmax Function



5 minutes ago

09:07



Softmax function formalue

## Softmax Function

$$\text{Softmax } f_i(x) = \frac{e^{a_i}}{\sum_k e^{a_k}}$$

4 minutes ago

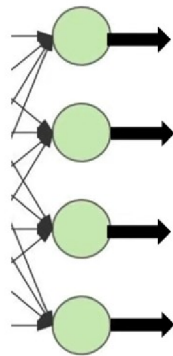
11:16



For linear regression problems, no activation function is required in the output neuron.

### Multi-Class Classification

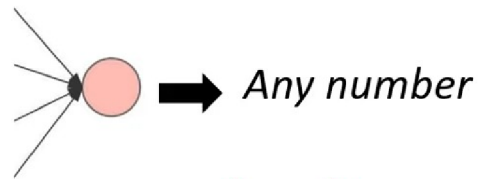
*tanh or ReLU in hidden layers*



$$\text{Softmax } f_i(x) = \frac{e^{a_i}}{\sum_k e^{a_k}}$$

*Softmax in Output layer*

### Linear Regression



*No Activation Function in Output Neuron*

2 minutes ago