♦

Activation Function

◉ Difficulty	Easy 🍋
◉ Select	main

📌Title : Activation Function

🐇 **Summary**

- Activation function and weighted sum of the input is in every neuron
- Activation function helps us to formulate the output to a specific range
- Activation function helps us to make the model non-linear

▼ Google Collab Practice 💻

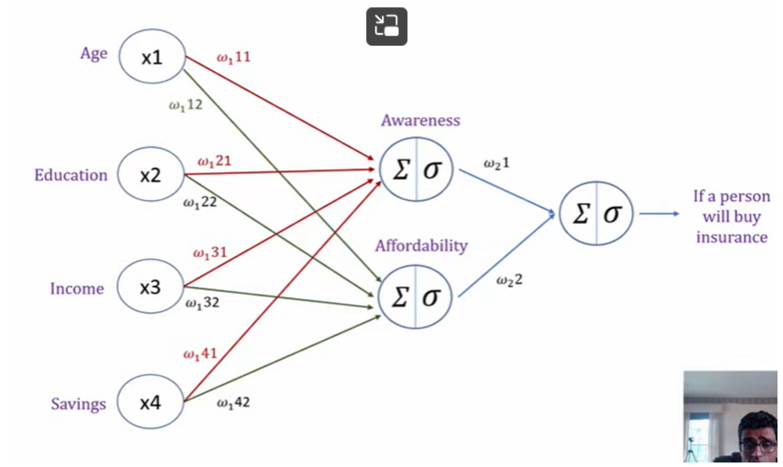
▼ Further research 🔎

▼ Resources

Activation Functions | Deep Learning Tutorial 8 (Tensorflow Tutorial, Keras & Python) - YouTube

📉 Why Activation Function?

- Helpful in output layer
    - The sigmoid function helps us see if the neuron will fire or not **(0 or 1 ?)**
- Helpful in hidden layer
    - To make the neuron output **non-linear**



🧠 Examples of activation functions

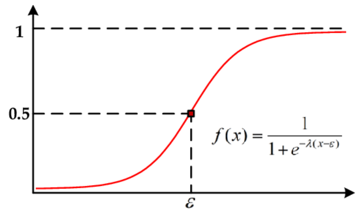
Generally in binary classifications:

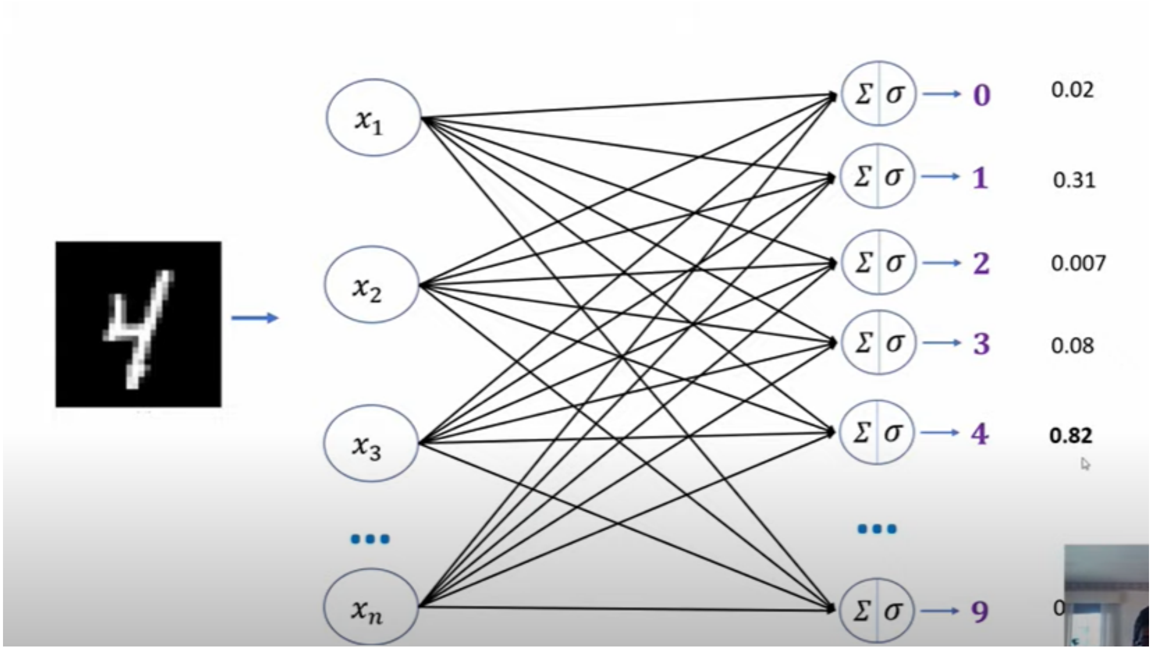
📝 Use **tanh** in all hidden layers, and **sigmoid** in output layer

📝use **ReLU** is the default choice for hidden layer activation function

▼ sigmoid (number between 0 and 1)

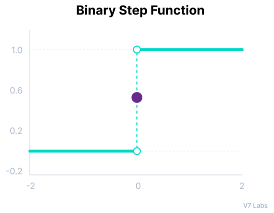
- Better than the step function because you choose the one with higher probability

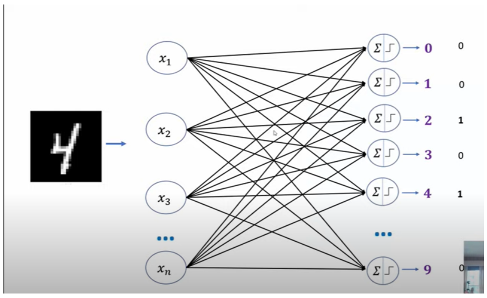




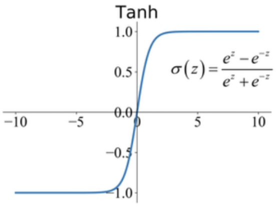
- Has the vanishing gradient problem as you get closer to the asymptote the learning starts to get so slow

▼ step function (0 or 1)



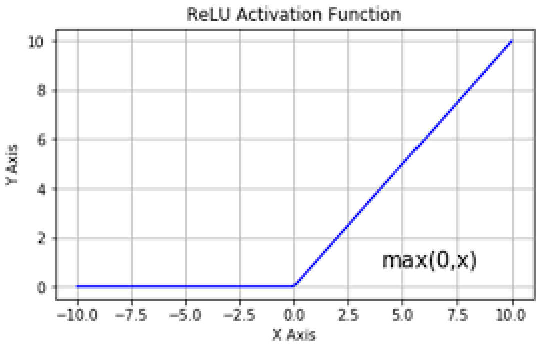


▼ tanh



- Has the vanishing gradient problem as you get closer to the asymptote the learning starts to get so slow

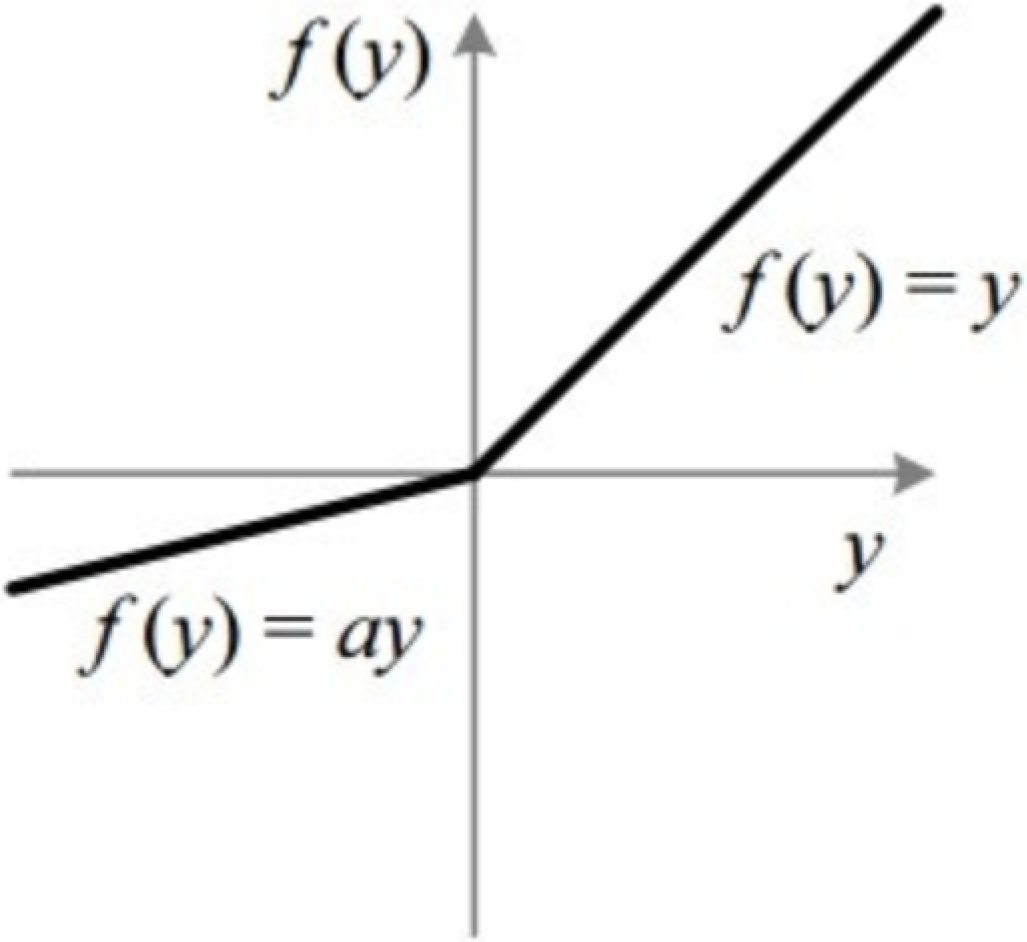
▼ ReLU



- solves the **vanishing gradient problem** as its always increasing, but still same problem in decreasing.
- very light-weight function

▼ Leaky ReLU

- solves the vanishing gradient as you increase and decrease


$f(y)$
$f(y) = y$
y
$f(y) = ay$