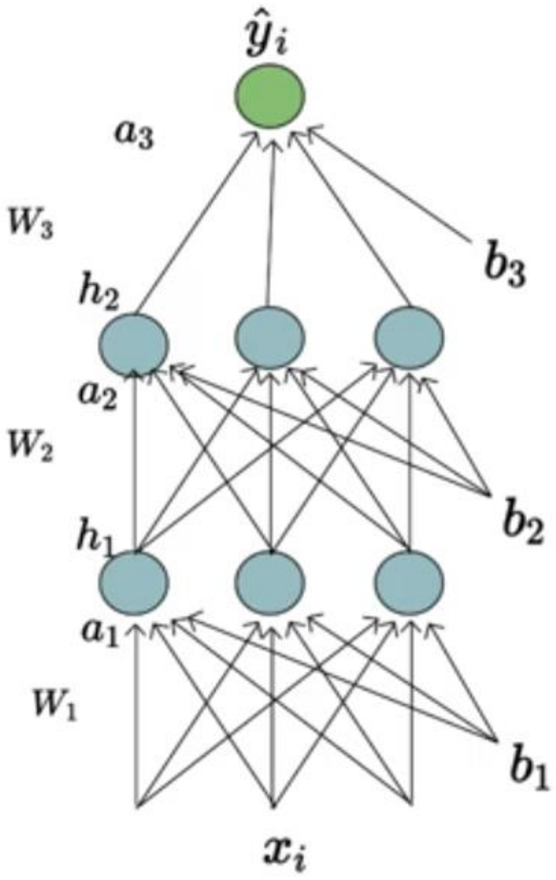
Deep Learning: Activation Function

Course Instructor:
Dr. Bam Bahadur Sinha
Assistant Professor
Computer Science & Engineering
National Institute of Technology Sikkim

Why are Activation Functions Important?



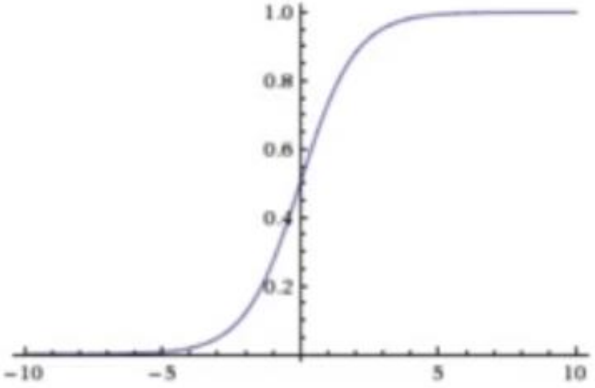
What happens if there are no non-linear activation functions in the network ?

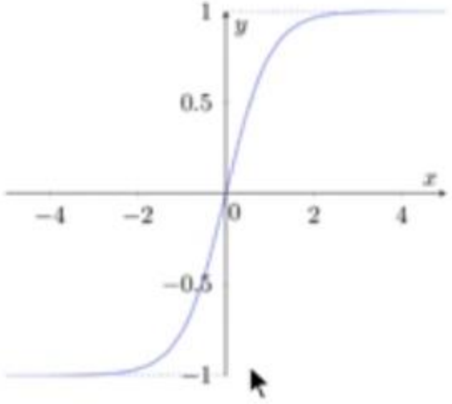
$$\hat{y}_i = W_3(W_2(W_1 x_i)) = W x_i$$

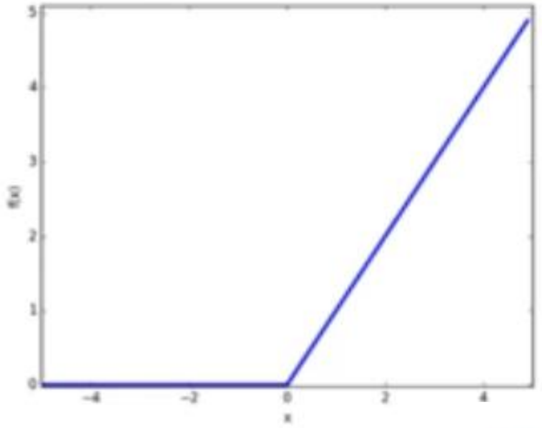
- Can only represent linear relations between x and y
- UAT does not hold!

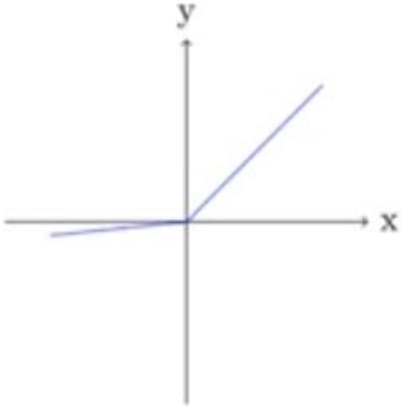
The representation power of a deep neural network is due to its non-linear activation functions.

Commonly used Non-Linear Activation Functions





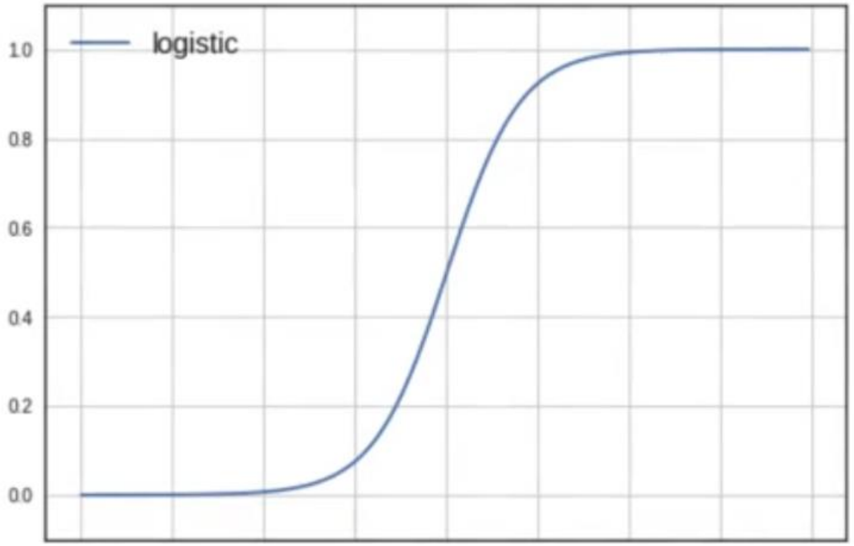




- logistic
- tanh
- ReLU
- Leaky ReLU

Logistic Function

Problem 1: Saturation



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

$$f'(x) = \frac{\partial f(x)}{\partial x} = f(x) * (1 - f(x))$$

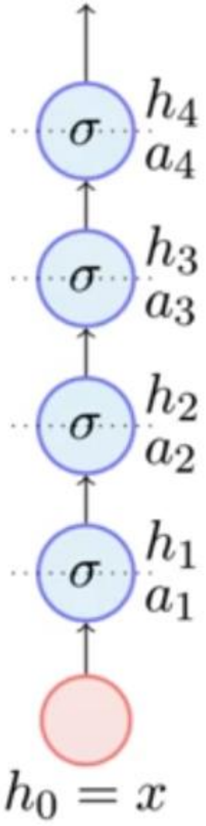
Saturation:

when $f(x) = 0$ *or* 1

and hence $f'(x) = 0$

When do we say a sigmoid neuron is saturated and what is its implications?

Implications of Saturated Neurons - Vanishing Gradient Problem



$$a_3 = w_2 h_2$$
$$h_3 = \sigma(a_3)$$

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

$$f'(x) = \frac{\partial f(x)}{\partial x} = f(x) * (1 - f(x))$$

Saturation:

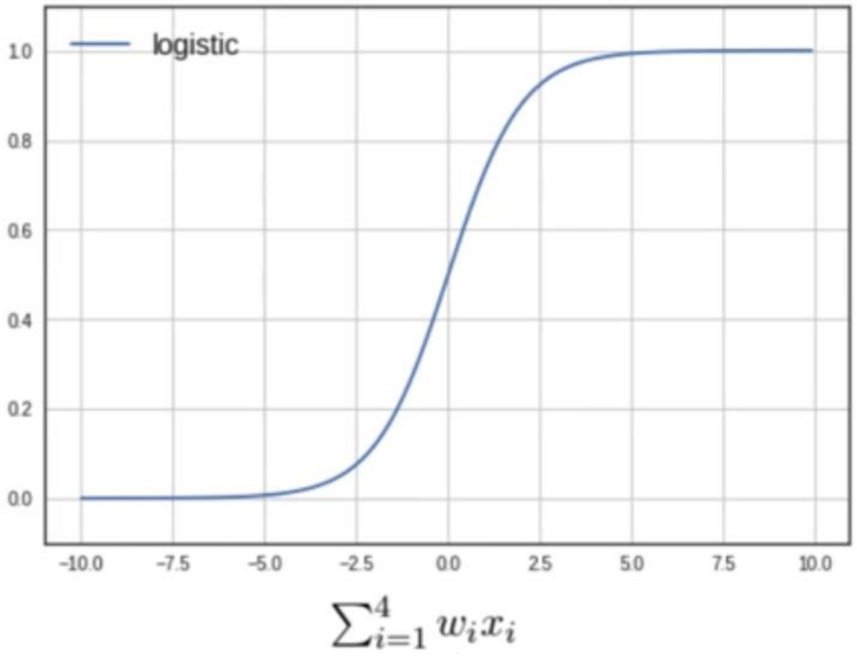
when $f(x) = 0$ *or* 1

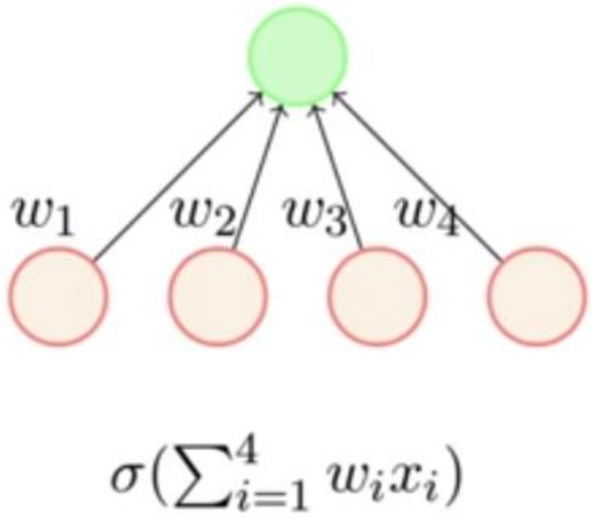
and hence $f'(x) = 0$

Saturated neurons cause the gradients to vanish

The neurons generally saturate due to very large value of weights in positive or negative.

Why would neurons saturate?





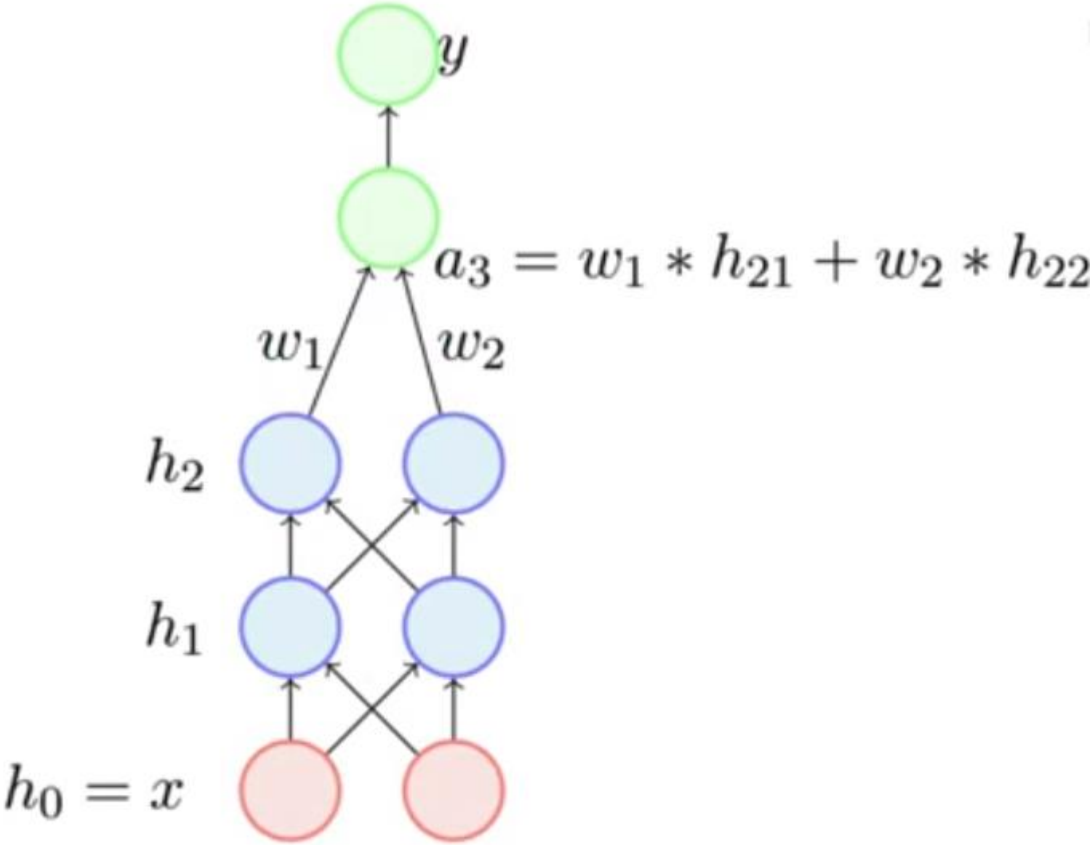
Make sure to initialize the weights with small values.

Problem 2: Not Zero-Centered

logistic function is not zero-centered

$$\nabla w_1 = \frac{\partial \mathscr{L}(\mathbf{w})}{\partial y} \frac{\partial y}{h_3} \frac{\partial h_3}{\partial a_3} h_{21}$$

$$\nabla w_2 = \frac{\partial \mathscr{L}(\mathbf{w})}{\partial y} \frac{\partial y}{h_3} \frac{\partial h_3}{\partial a_3} h_{22}$$



The gradients w.r.t. all the weights connected to the same neuron are either all +ve or all -ve

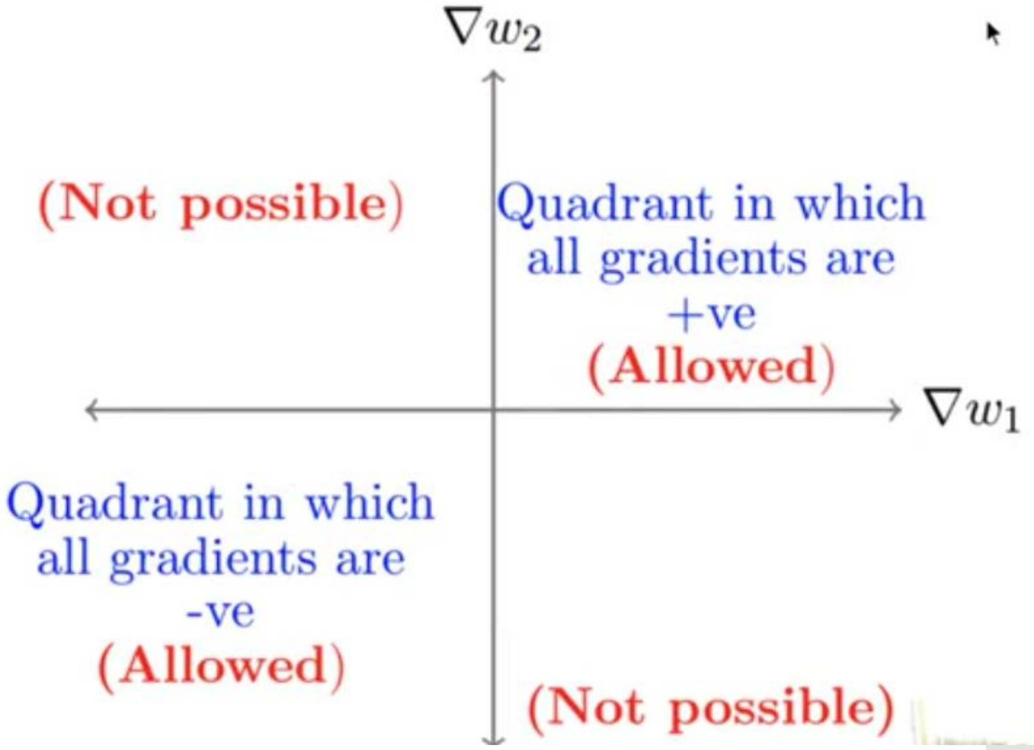
Problem 2: Not Zero-Centered

logistic function is not zero-centered

$$\nabla w_1 = \frac{\partial \mathscr{L}(\mathbf{w})}{\partial y} \frac{\partial y}{h_3} \frac{\partial h_3}{\partial a_3} h_{21}$$

$$\nabla w_2 = \frac{\partial \mathscr{L}(\mathbf{w})}{\partial y} \frac{\partial y}{h_3} \frac{\partial h_3}{\partial a_3} h_{22}$$

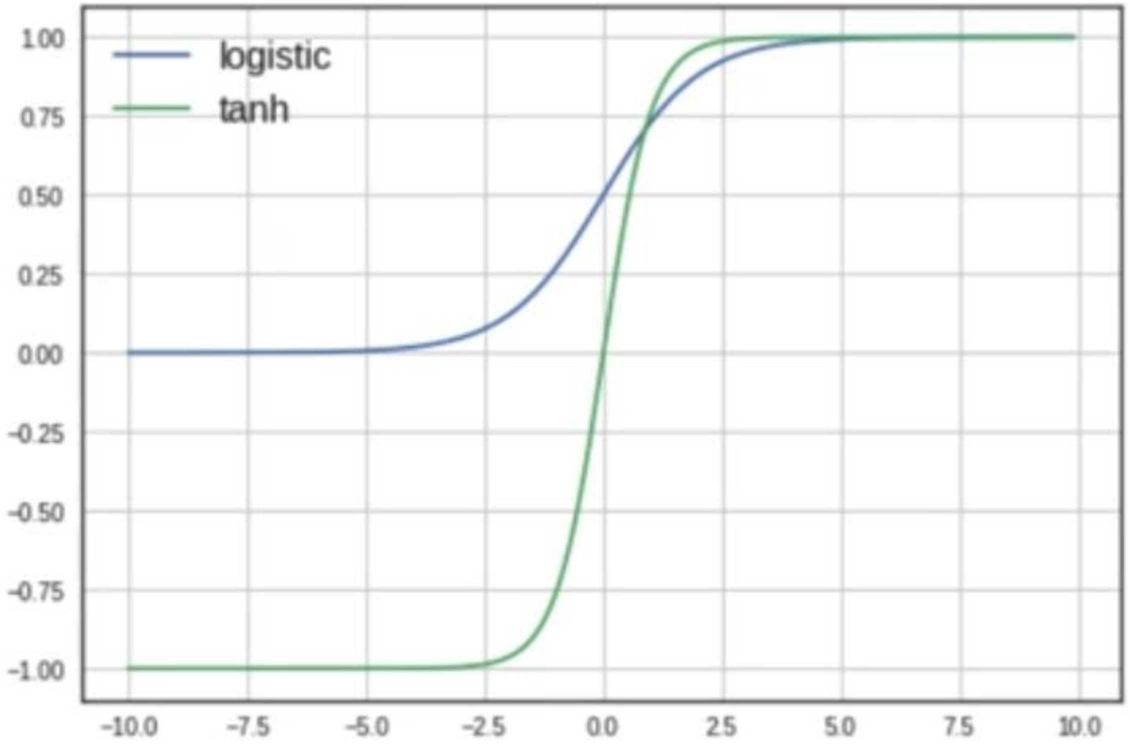
The gradients w.r.t. all the weights connected to the same neuron are either all +ve or all -ve



Issues with sigmoid / logistic neuron

- Saturated logistic neurons cause the gradients to vanish.
- Logistic Function is not Zero-Centered.
- Logistic function is computationally expensive due to the computation of exponential term.

Tanh Function



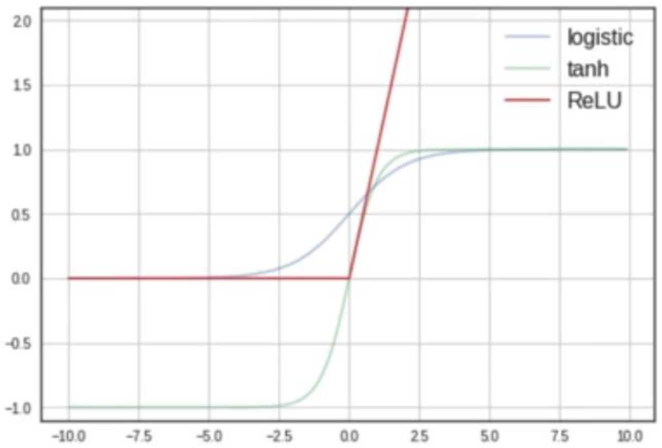
$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

$$f'(x) = \frac{\partial f(x)}{\partial x} = (1 - (f(x))^2)$$

Implications of tanh

- tanh cause the gradients to vanish.
- tanh function is Zero-Centered.
- Tanh function is computationally expensive due to the computation of exponential term.
- Tanh performs better than logistic function

ReLU

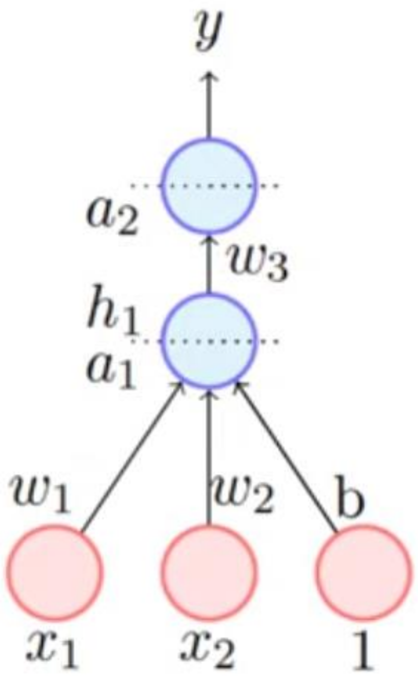


$$f(x) = max(0, x)$$

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

- Does not saturate in the positive region
- Not Zero-centered
- Easy to Compute (no expensive exponential computation)

Issues with ReLU



$$\nabla w_1 = \frac{\partial \mathscr{L}(\theta)}{\partial y} \cdot \frac{\partial y}{\partial a_2} \cdot \frac{\partial a_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial a_1} \cdot \frac{\partial a_1}{\partial w_1}$$

$$\begin{aligned} h_1 &= ReLU(a_1) = max(0, a_1) \\ &= max(0, w_1 x_1 + w_2 x_2 + b) \end{aligned}$$

What happens if b takes on a large negative value due to a large negative update (∇b) at some point ?

$$w_1 x_1 + w_2 x_2 + b < 0 \quad [if \quad b << 0]$$

$$\implies h_1 = 0 \quad [dead\ neuron]$$

$$\implies \frac{\partial h_1}{\partial a_1} = 0$$

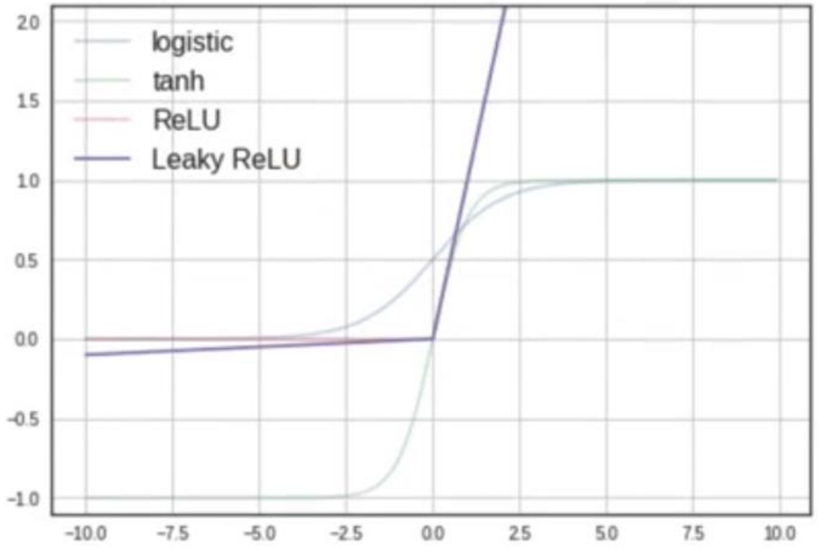
$$\implies w_1, w_2, b\ remain\ unchanged$$

$$\implies the\ neuron\ stays\ dead\ forever$$

Key takeaways for ReLU

- A large fraction of ReLU units can die if the learning rate is set too high.
- It is advised to initialize the bias to a positive value.
- Use other variants of ReLU

Leaky ReLU



$$f(x) = max(0.01x, x)$$

$$f'(x) = \frac{\partial f(x)}{\partial x} = \begin{cases} 0.01 \quad if \quad x < 0 \\ 1 \quad if \quad x > 0 \end{cases}$$

- Does not saturate in positive/negative region.
- Will not die (0.01 x ensures that at least a small gradient will flow through)
- Easy to compute (no exponential term)
- Close to zero-centered outputs.