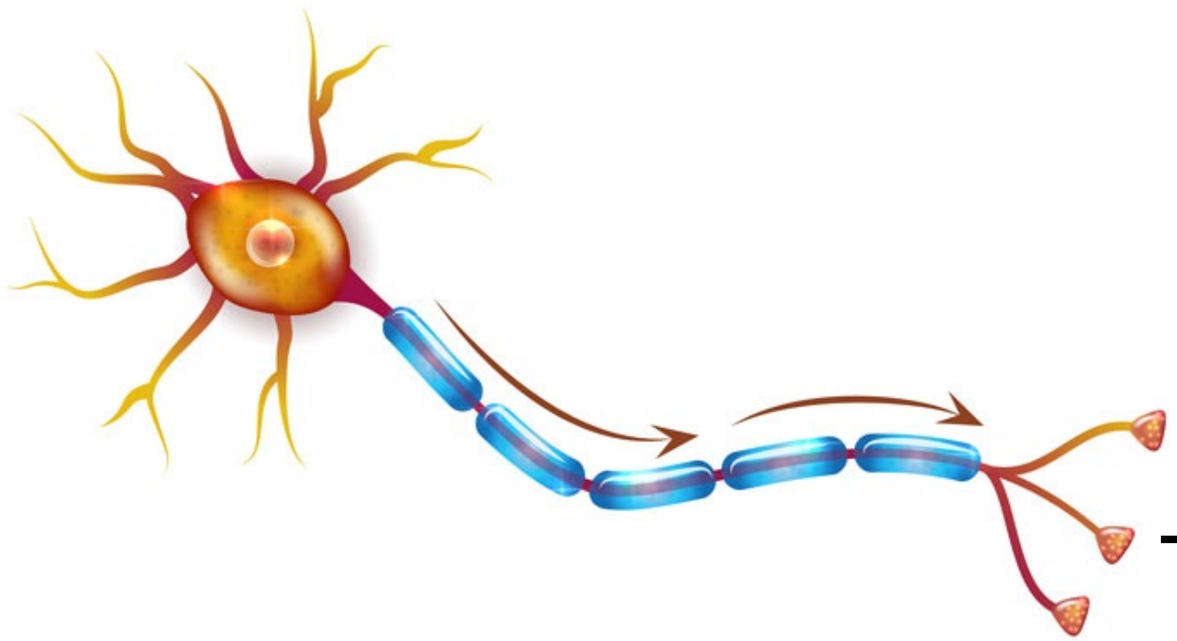
Activation Functions

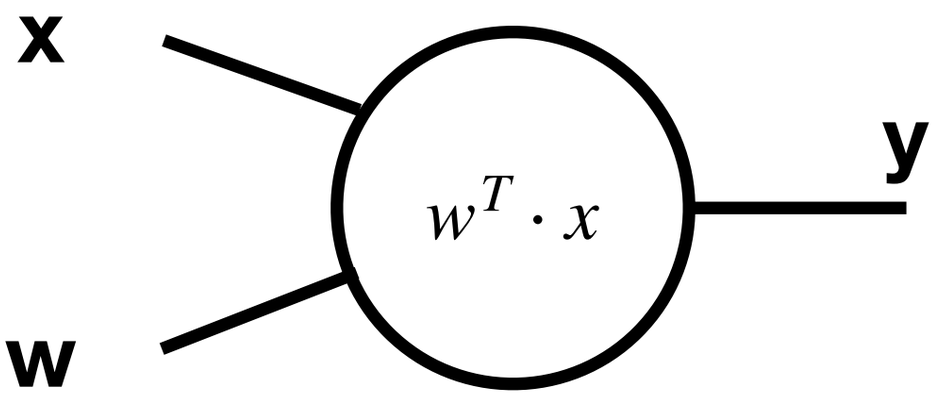
Activation function

Activation functions apply a non-linear transformation and decide whether a neuron should be activated or not.

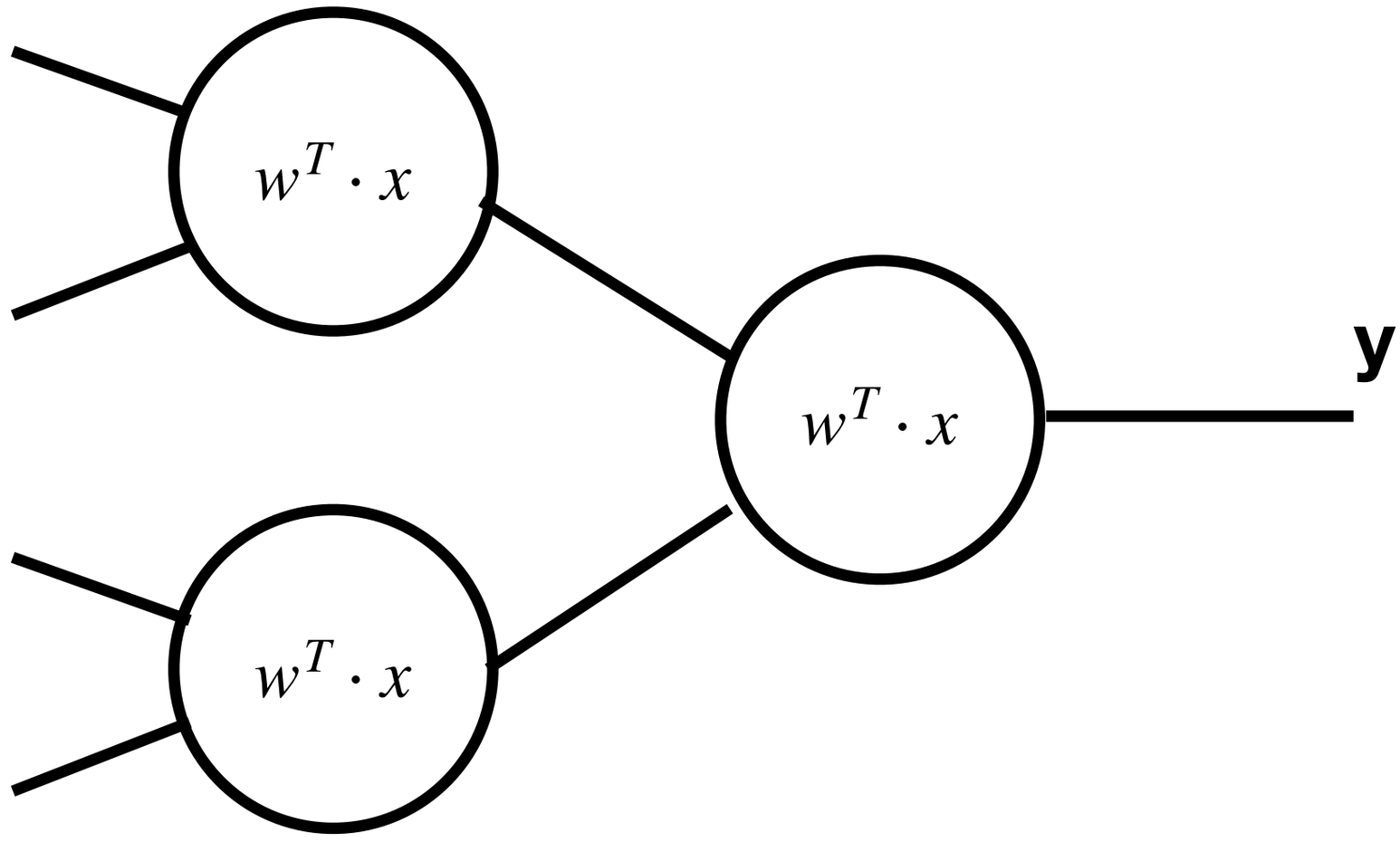


—> Activated?

Why?

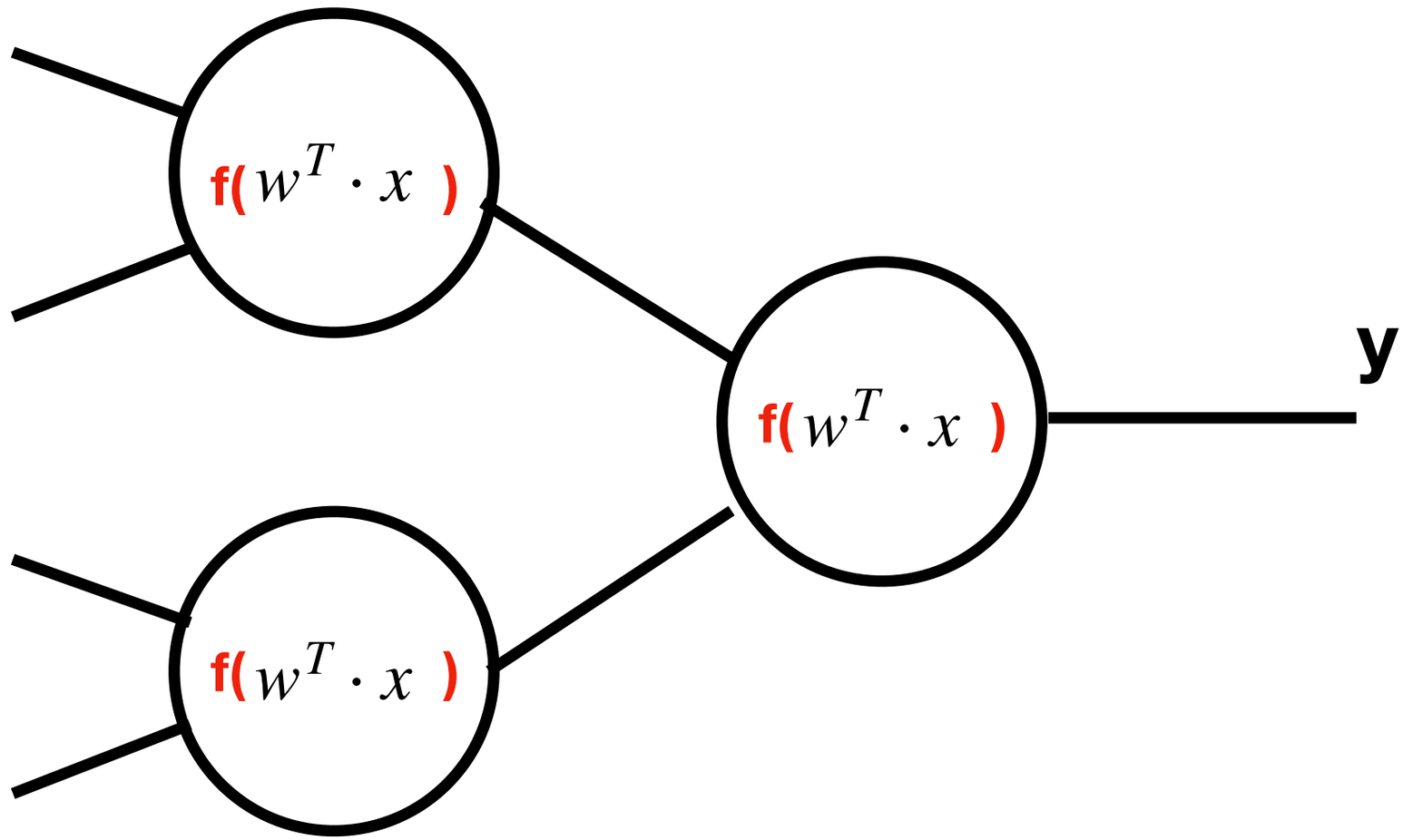


Why?



Without activation functions our network is basically just a stacked linear regression model

Why?



—> With non-linear transformations our network can learn better and perform more complex tasks!

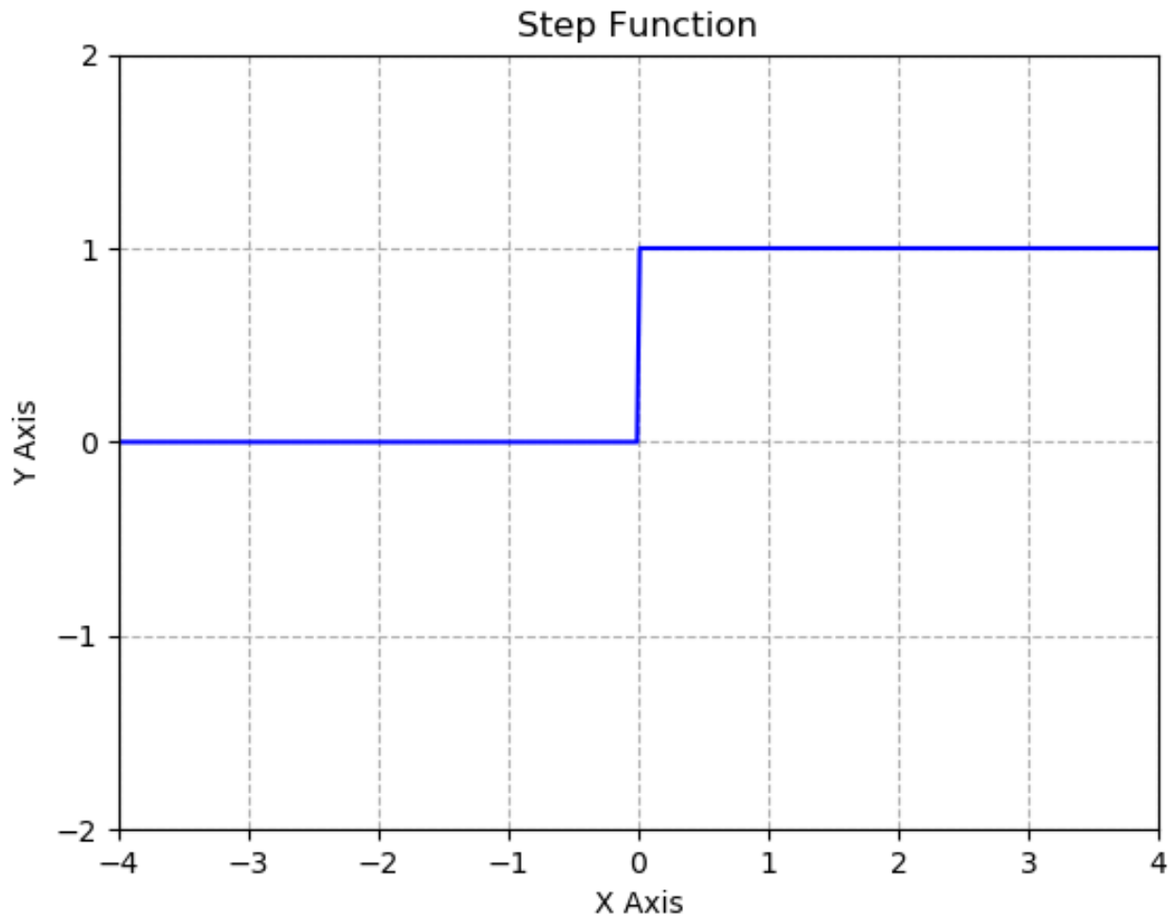
—> After each layer we typically use an activation function!

Most popular activation functions

1. **Step function**
2. **Sigmoid**
3. **TanH**
4. **ReLU**
5. **Leaky ReLU**
6. **Softmax**

Step Function

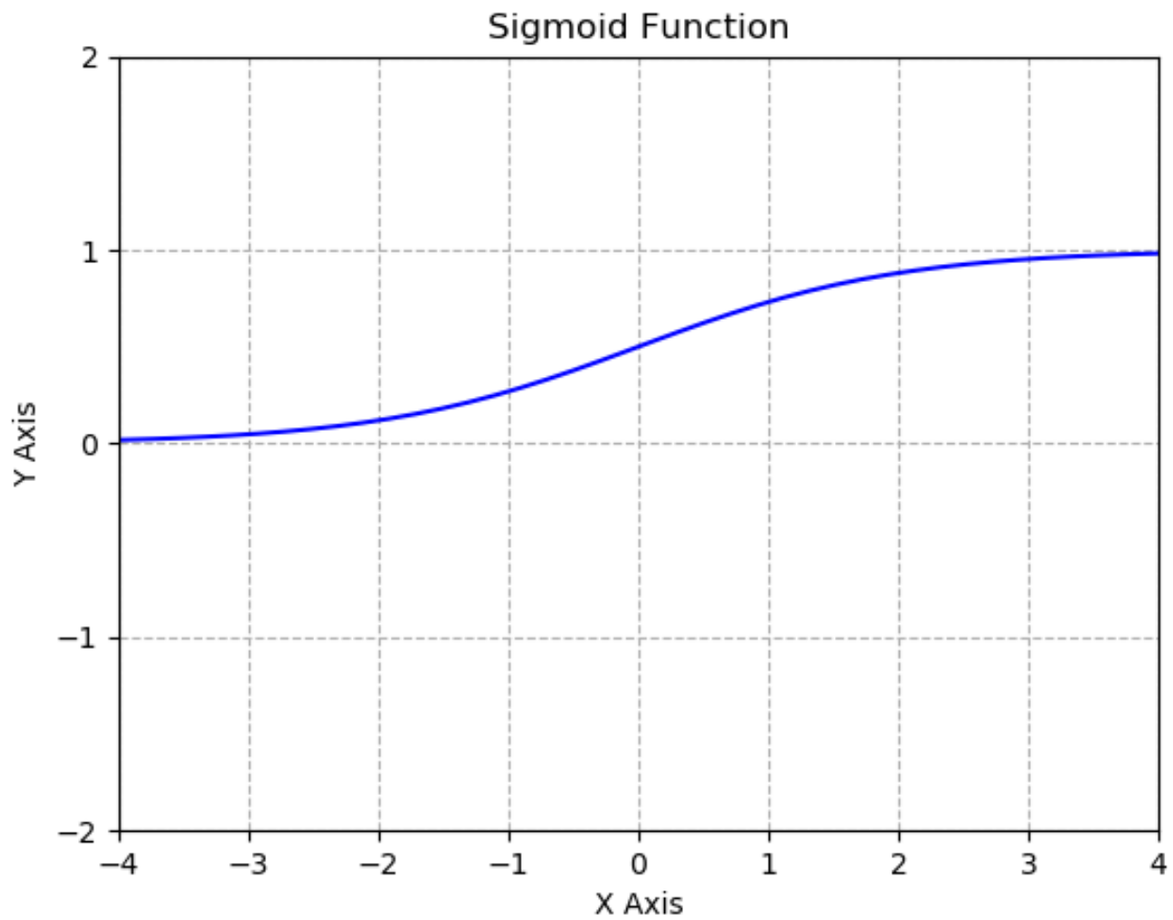
$$f(x) = \begin{cases} 1 & \textbf{if } x \geq \theta \\ 0 & \textbf{otherwise}. \end{cases}$$



—> Not used in practice

Sigmoid Function

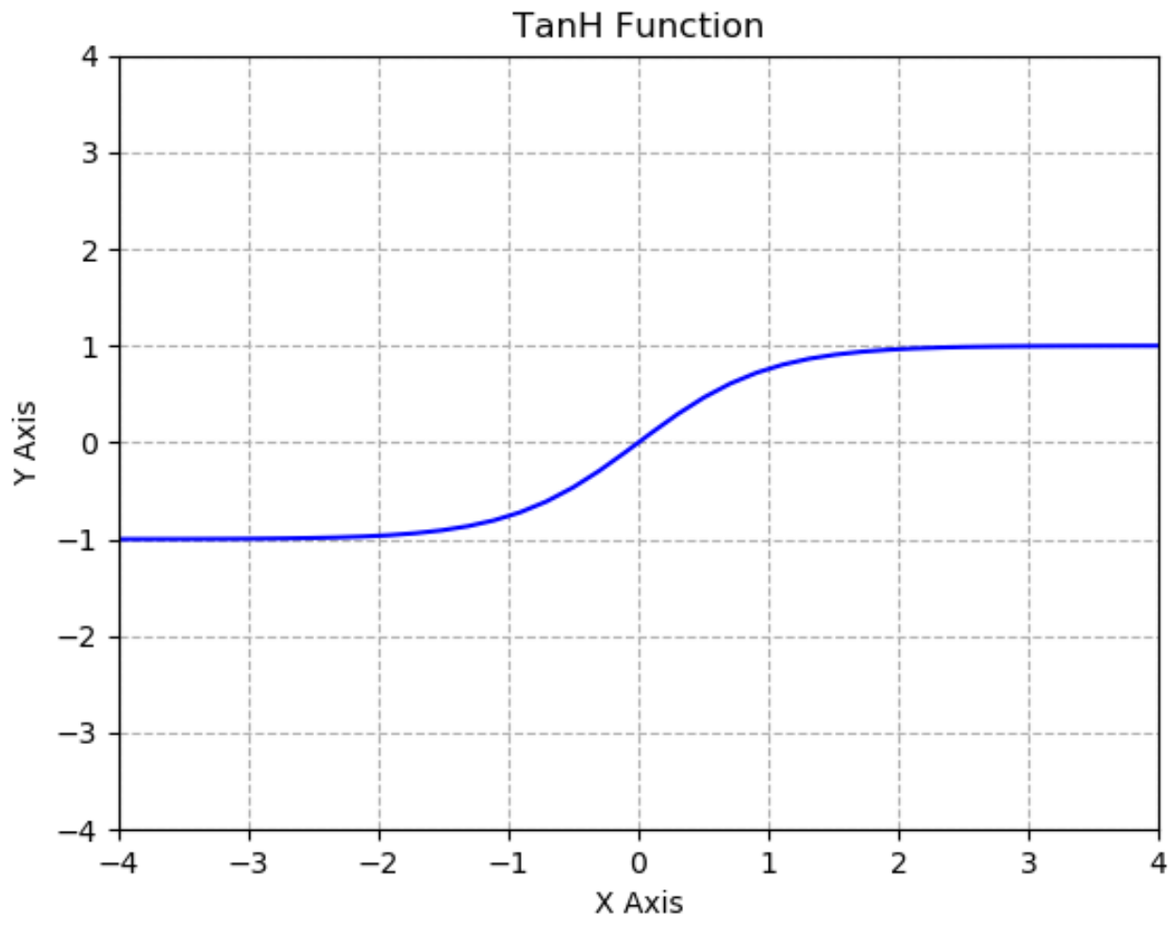
$$f(x) = \frac{1}{1 + e^{-x}}$$



—> Typically in the last layer of a binary classification problem

TanH Function

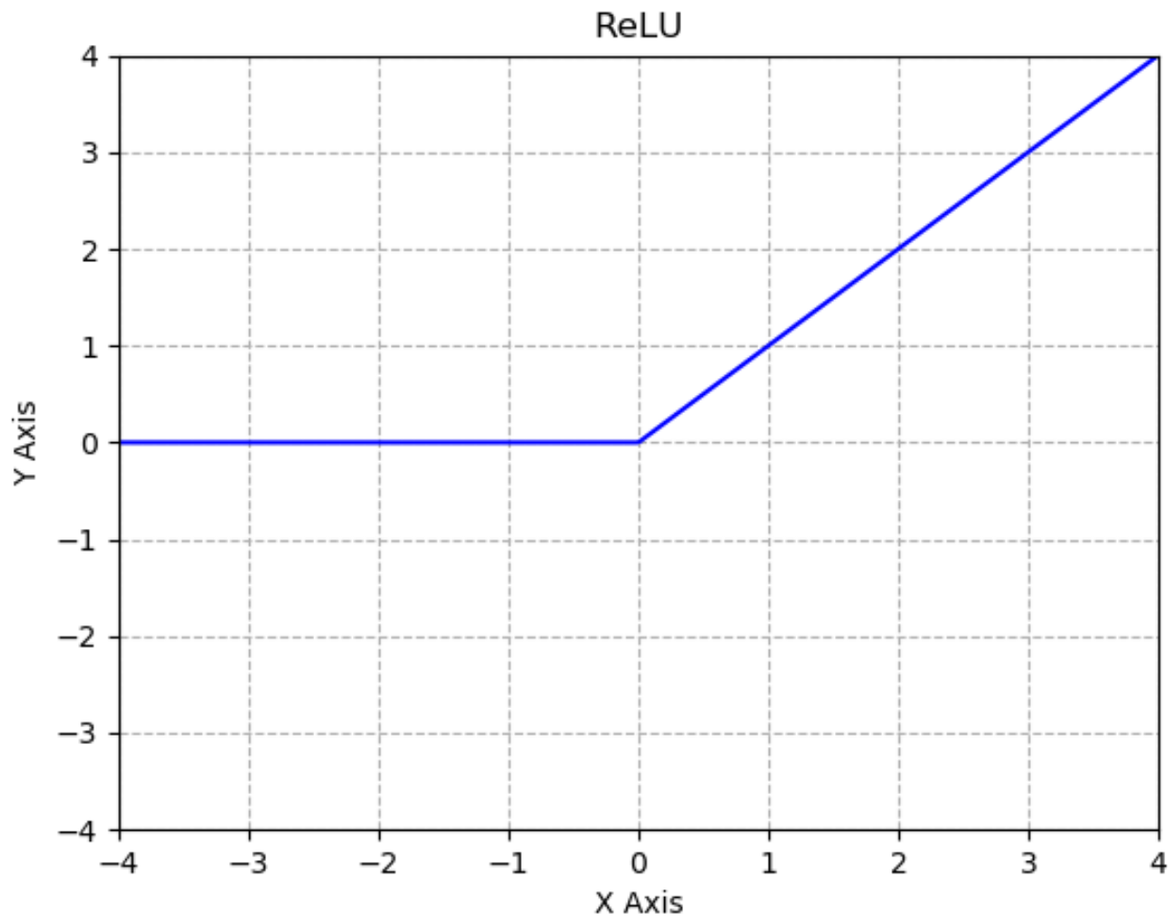
$$f(x) = \frac{2}{1 + e^{-2x}} - 1$$



—> Hidden layers

ReLU Function

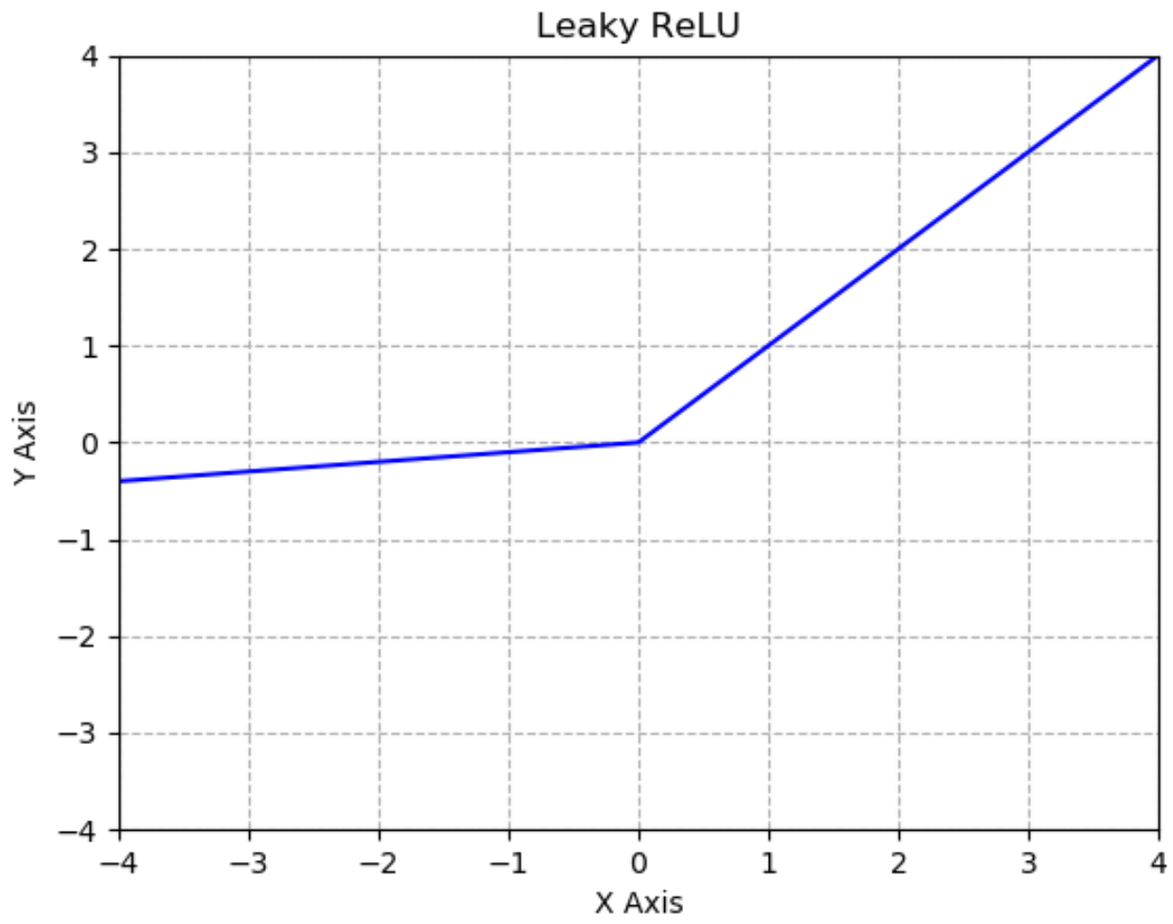
$$f(x) = max(0,\ x)$$



—> If you don’t know what to use, just use a ReLU for hidden layers

Leaky ReLU Function

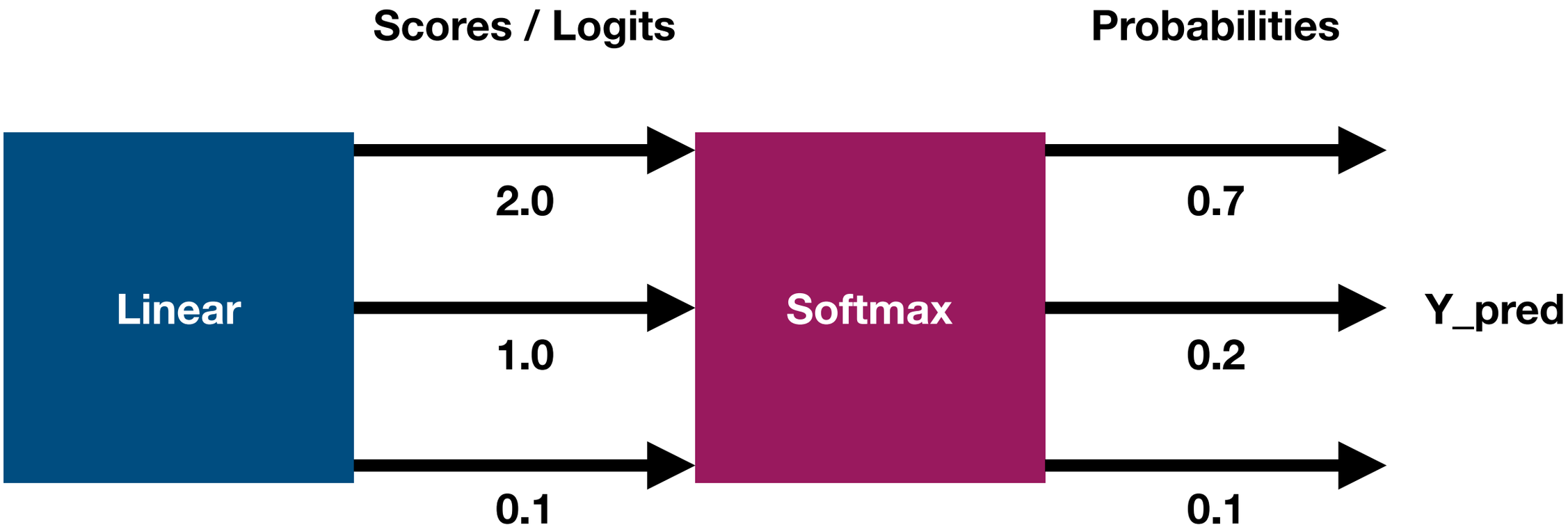
$$f(x) = \begin{cases} x & \textbf{if } x \geq 0 \\ a \cdot x & \textbf{otherwise}. \end{cases}$$



—> Improved version of ReLU. Tries to solve the vanishing gradient problem

Softmax

$$S(y_i) = \frac{e^{y_i}}{\sum e^{y_j}}$$



—> Good in last layer in multi class classification problems