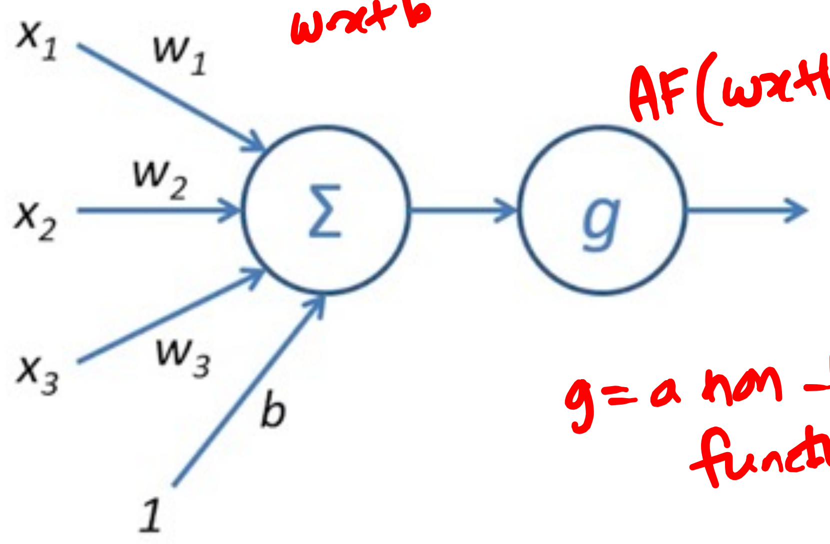
Activation Functions

NN Training is all about tuning weights & biases.

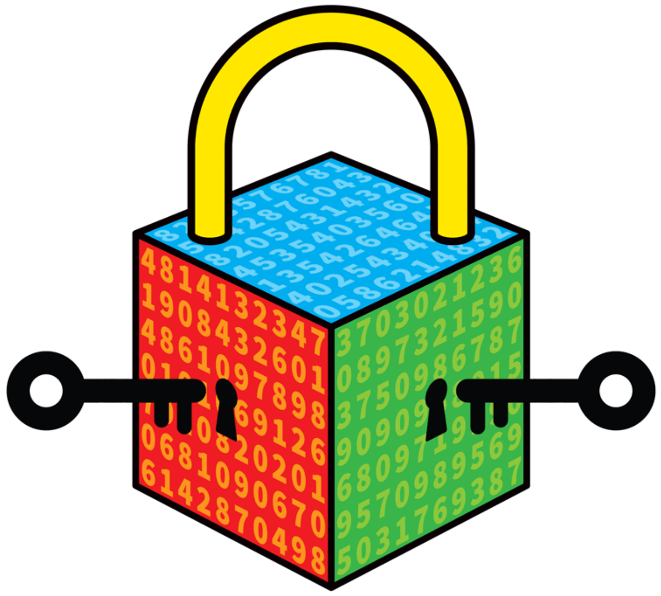
- This is the secret of the Neural Network.



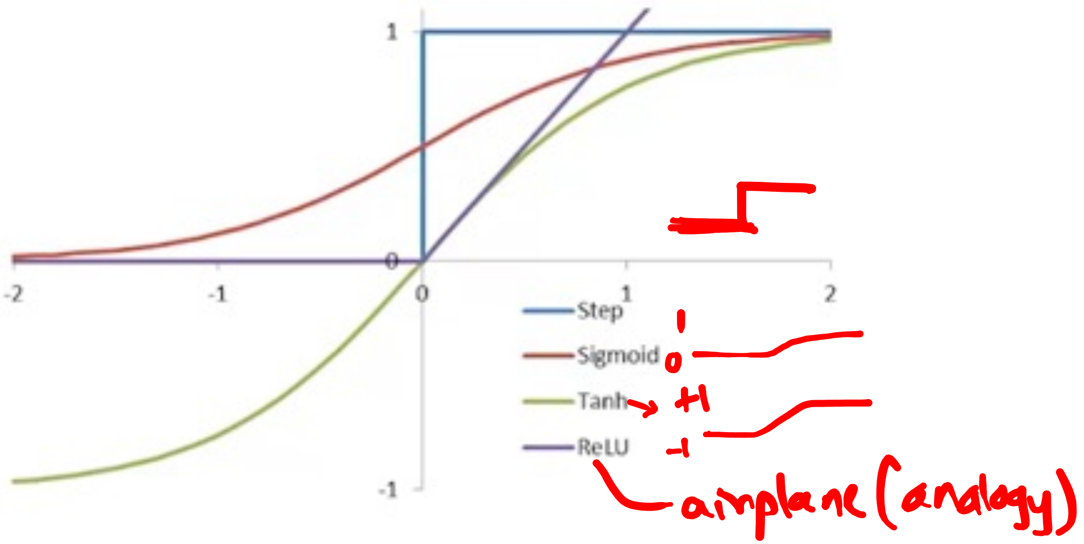
$wx+b$

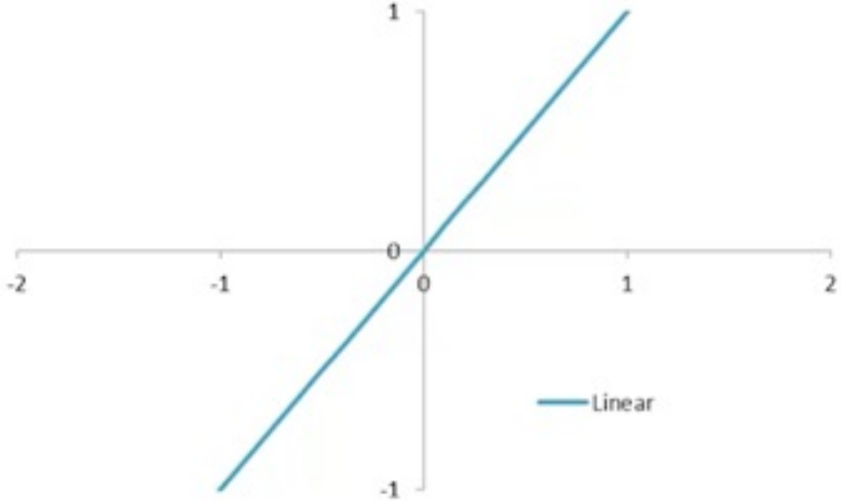
$AF(wx+b)$

g = a non-linear function



Types of Activation Functions





Types of Activation Functions

was used very popularly by researchers for Binary Classification problems

Stair

$wx+b>0 \rightarrow 1$

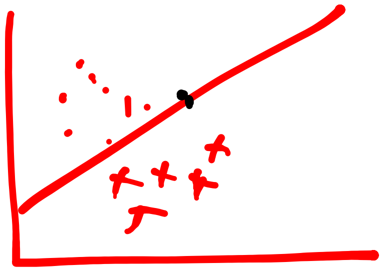
$wx+b<0,\ 0$

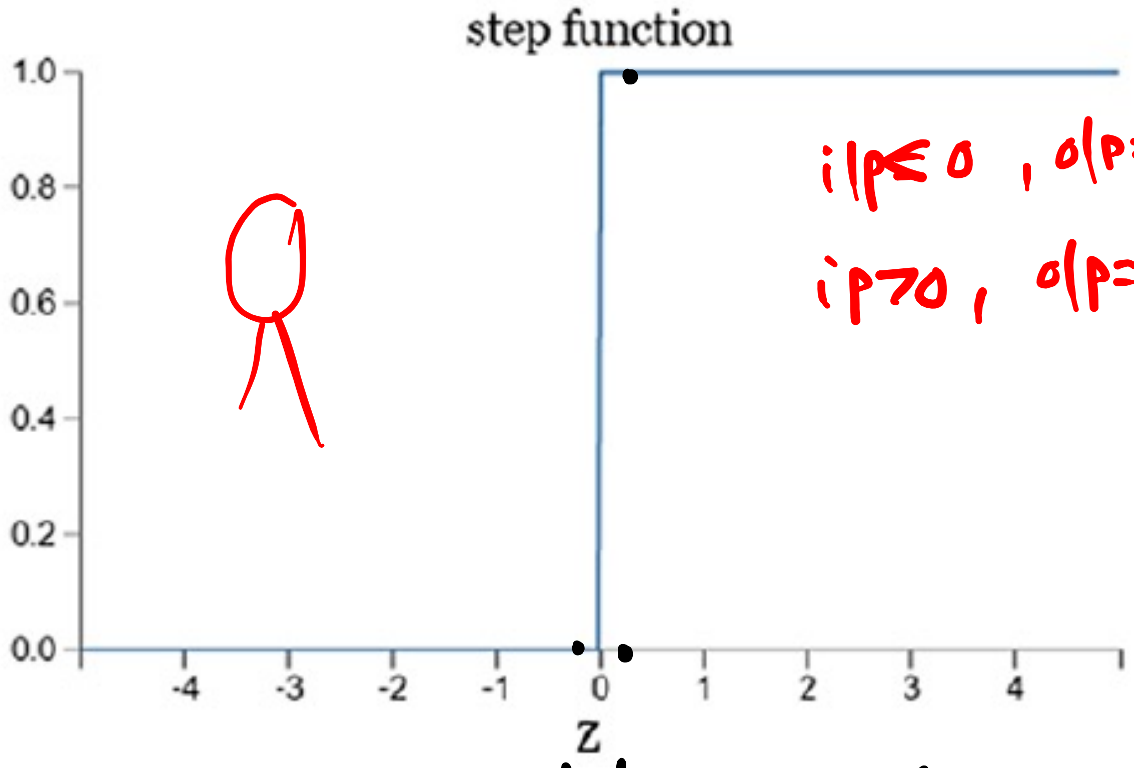
Step: Classification & Region Bifurcation

- **Step**: original concept behind classification and region bifurcation. Not used anymore

– No one uses this today, its obsolete

$$o/p = \sum_j w_j x_j + bias$$





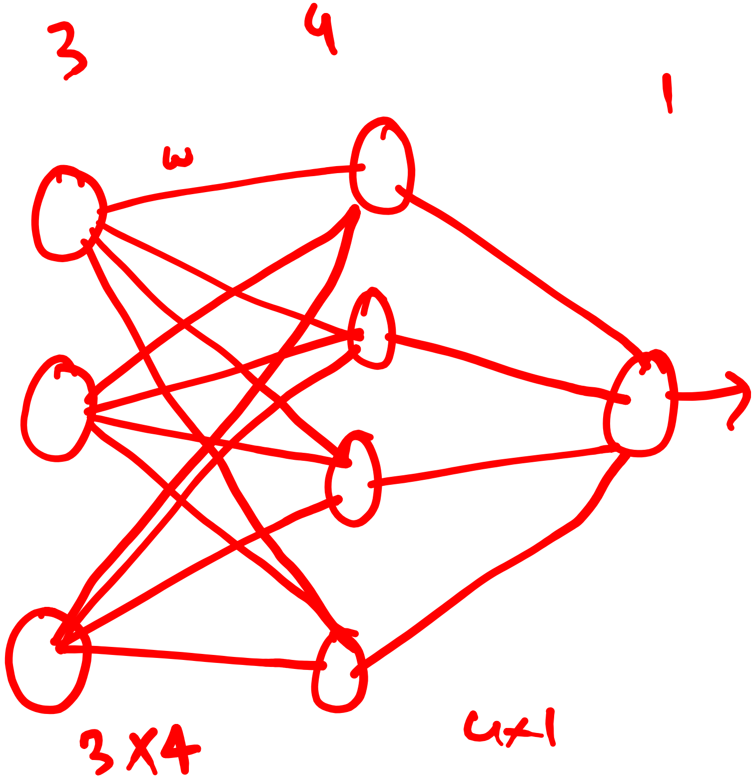
i/p ≤ 0, o/p=0

ip>0, o/p=1

abrupt change

Step function

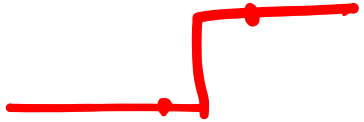
Problem with Perceptron?



Output + ΔOutput

0→1 or 1→0

w=5
↓
w=5.1



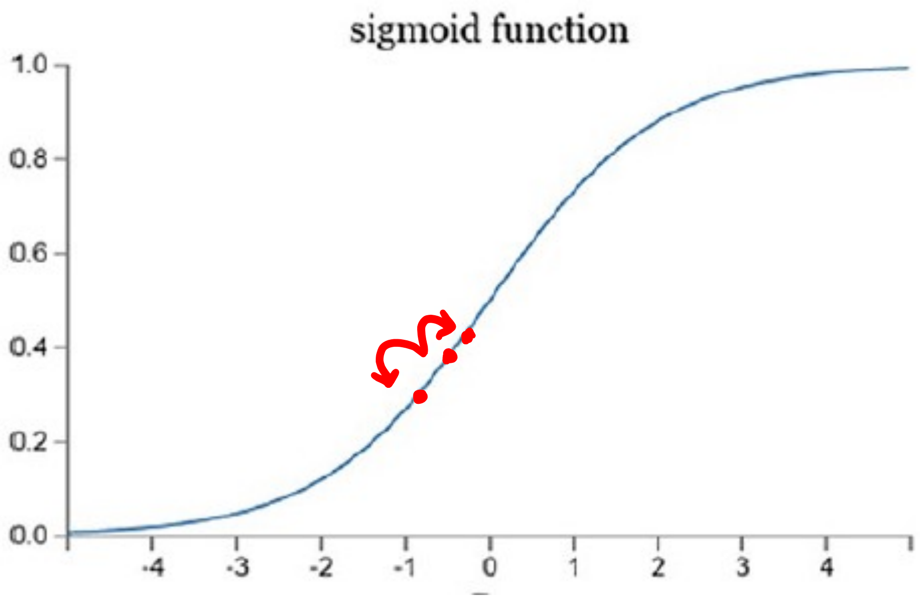
Types of Activation Functions

Sigmoid: is smoother than Step.

- i/p's can take any value b/w 0 & 1 instead of being binary.

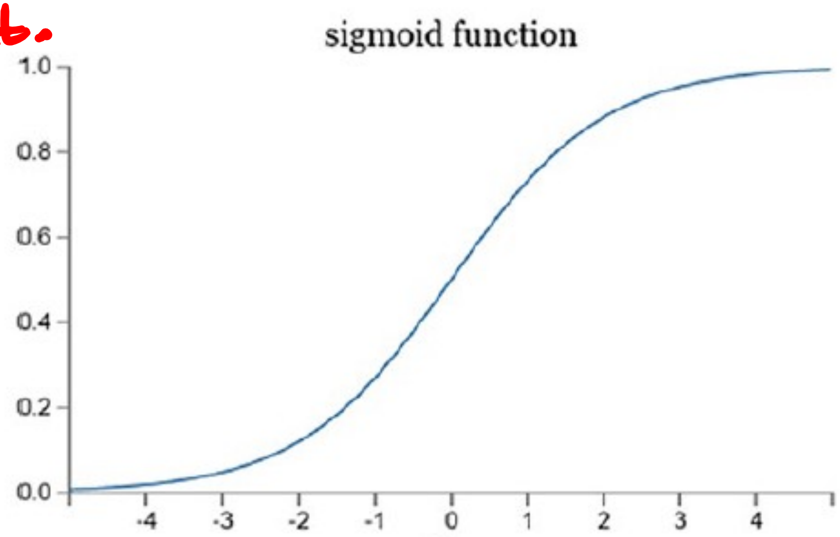
$$\sigma = \frac{1}{1+\exp^{-\sum_j w_j \cdot x_j + b}}$$

i.e. $\sigma = \frac{1}{1+\exp^{-mx+c}}$



Sigmoid are similar to Perceptrons

Sigmoids are modified step functions, so that small changes in the weight & bias cause only small change in their output.



0.627

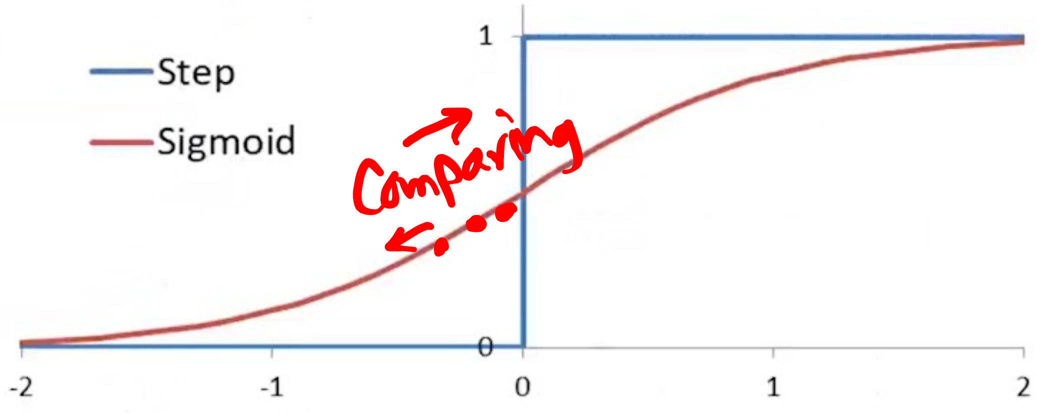
$\sigma(wx+b)$

i.e. $\sigma(z) \equiv \dfrac{1}{1+e^{-z}}$

$= \dfrac{1}{1+\exp^{-(\sum_j w_j x_j + b)}}$

Types of Activation Functions

- The sigmoid function is a smoother step function.
- Smoothness ensures that there is more information about the direction in which to change the weights if there are any errors.
- Sigmoid function is also mathematically linked to Logistic Regression, which is theoretically well-backed linear classifier.



$w_1 \rightarrow 0.39$

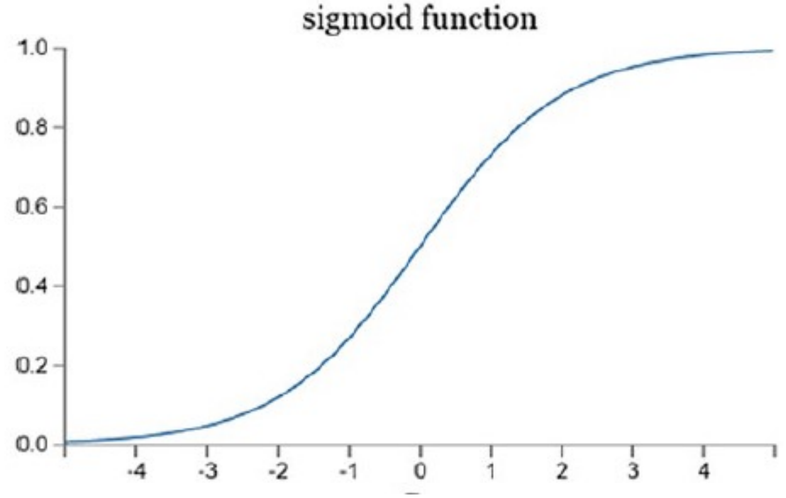
$w_1 + \Delta w \rightarrow 0.49$

$w_1 - \Delta w \rightarrow 0.21$

To understand the similarity to perceptron model

$\sigma : [0-1]$

$\sigma = \dfrac{1}{1+e^{-z}}$ whr $z = wx+b$



① $z = wx+b$ is a large +ve no.
then $e^{-z} \approx 0$ $\therefore \boxed{\sigma(z) \approx 1}$

② $z = wx+b$ is a large -ve no.
then $e^{-z} \to \infty$ $\therefore \boxed{\sigma(z) \approx 0}$

③ Only & only when wx+b is of a modest size, there's much deviation from the perceptron model.

Problem with Sigmoid: Vanishing Gradient

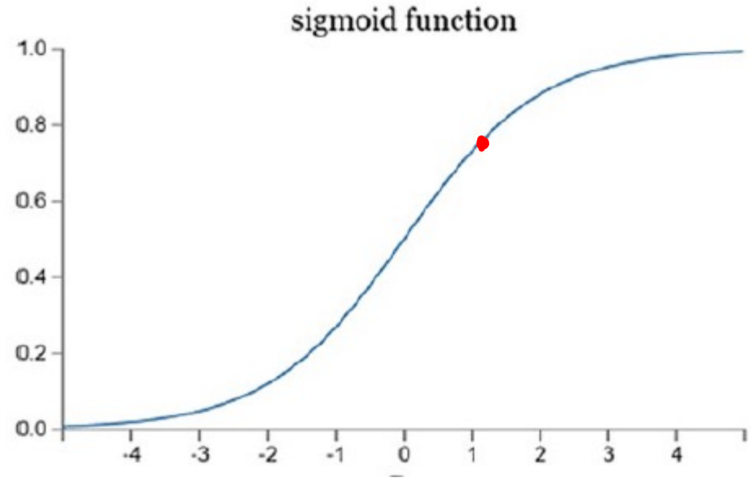
I/P : $[-\infty, +\infty]$

$-\infty$ .... $+\infty$

$\downarrow$

$[0, 1]$

0 1



Sigmoid has very steep gradient.

Thus, there remains large regions of input space, where even large changes produce very small change in output. This problem is called Vanishing Gradient Problem.

$\therefore$ Note: Sigmoids are used only at o/p nodes.

TANH Activation Function

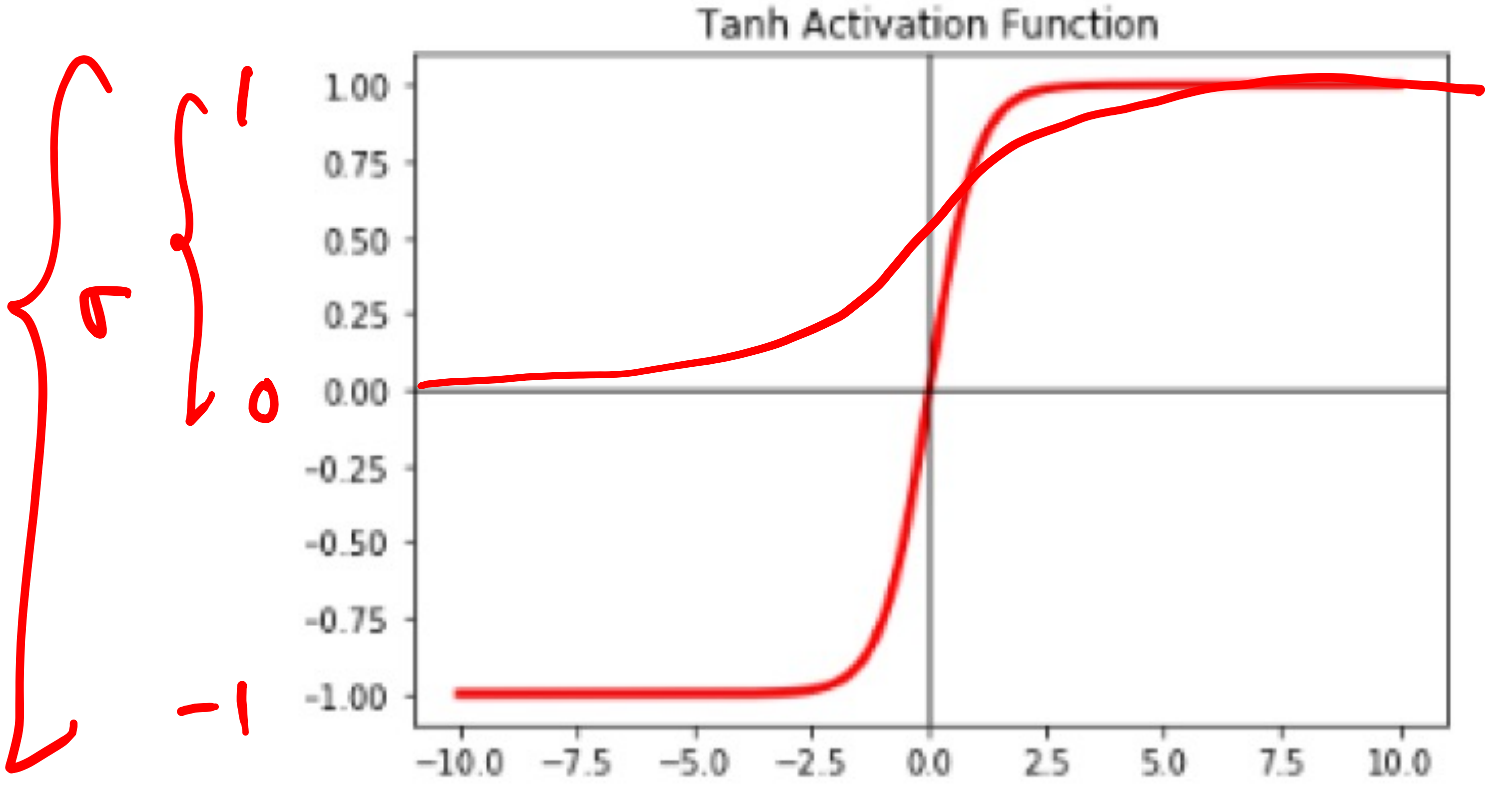
Sigmoid & tanh are qualitatively the same AF.

$\sigma = [0,1]$

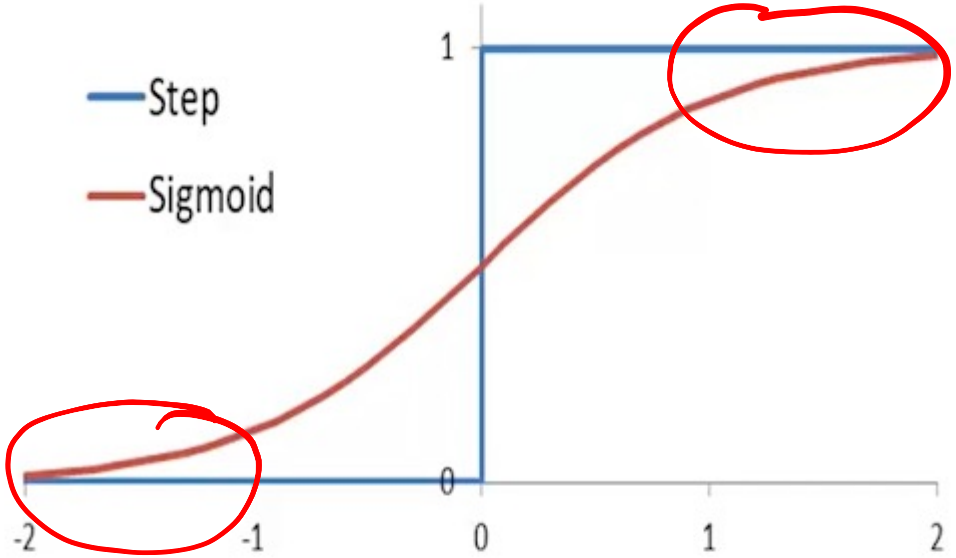
$\tanh[-1, +1]$

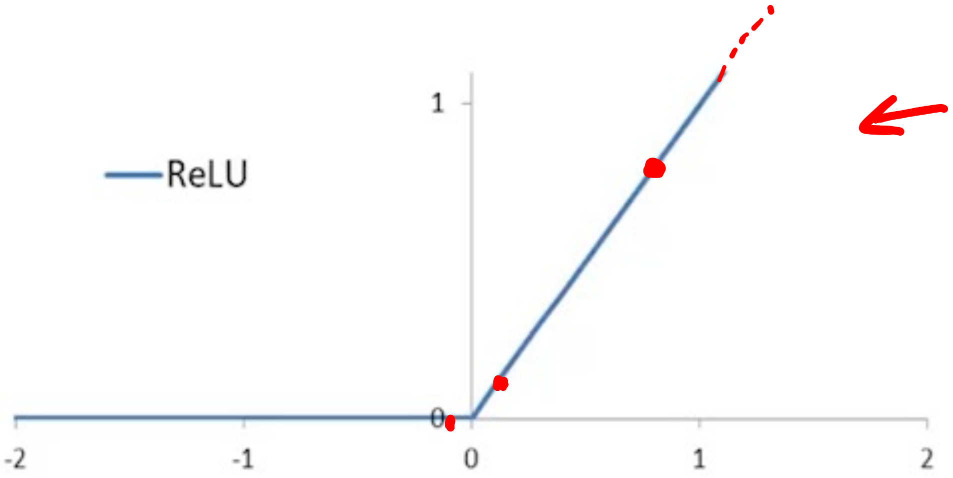
tanh

The tanh(z) function is a rescaled version of the sigmoid, and its output range is [– 1,1] instead of [0,1].



Problem with Sigmoid: Near zero gradient on both extremes.





← Hero

ReLU: Rectified Linear Unit.

- has a constant gradient for almost half of the inputs.
- ∴ ReLU cannot give a meaningful output.
- Used popularly in Hidden neurons.

ReLU Activation Function

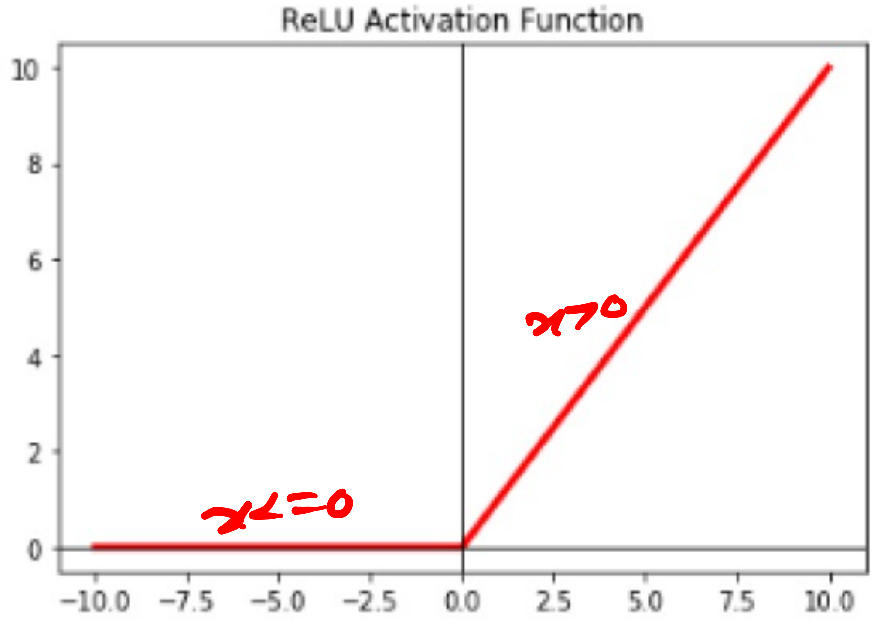
$z_j = f_j(x_j) = \max(0, x_j)$

whr x_j - j^{th} input value

z_j - corresponding output value after ReLU function.

Why are ReLU's so preferred?

– Due to fast convergence

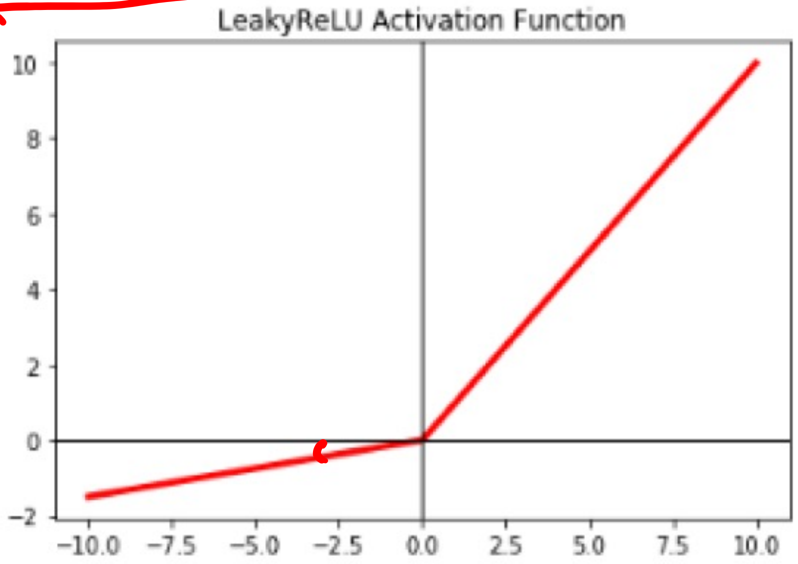


ReLU function graph

22022022

Problem with ReLU

often face the issue of dying, especially when the learning rate is set to a higher value, as this triggers weight updating.



Leaky ReLU: mitigate issue of dying ReLUs by introducing a marginally reduced slope (~ 0.01) for all $x<0$.

Note: LReLUs do offer successful scenarios, although not always.

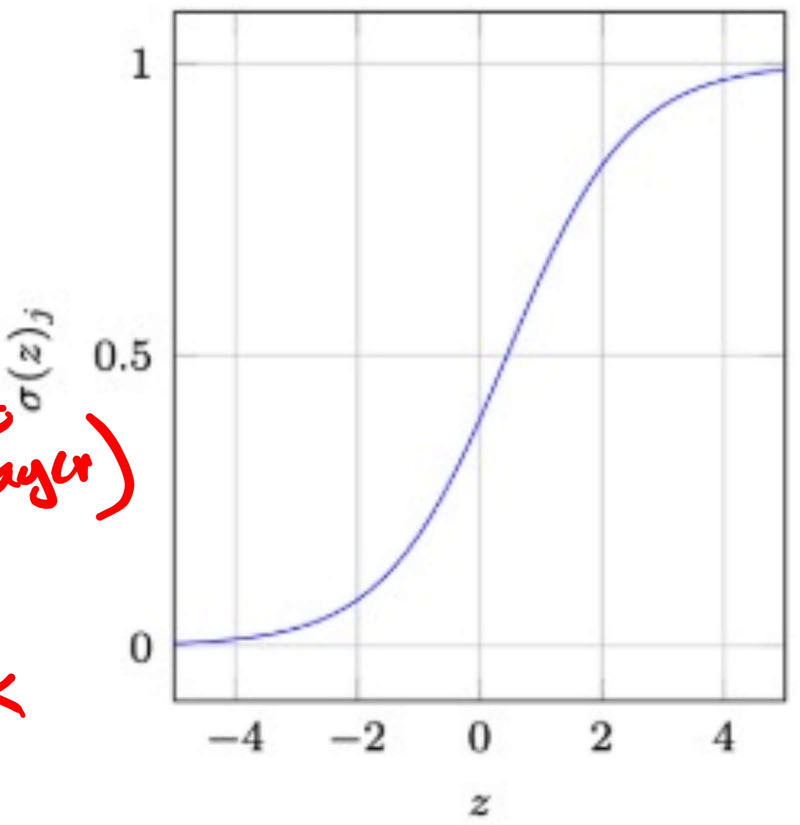
Softmax Activation Function

or Normalized Exponential

This function will transform a set of given real values in range (0,1) such that the combined sum=1.

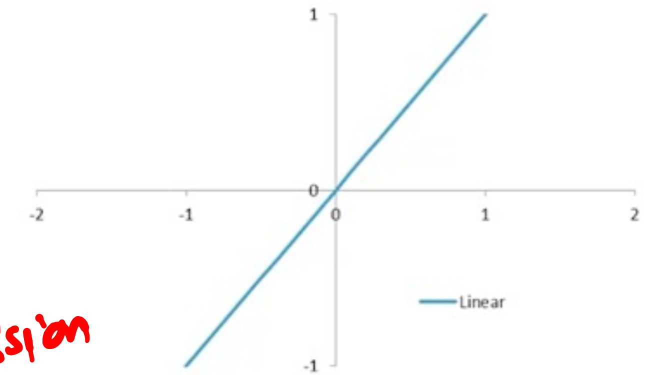
- Softmax — Multi Class Classification (Output layer)
- Sigmoid — Binary Classification.

Softmax: $\sigma(z)_j = \frac{e^{z_k}}{\sum_{k=}^{k} e^{z_k}}$ for $j = 1, \dots K$



Types of Activation Functions

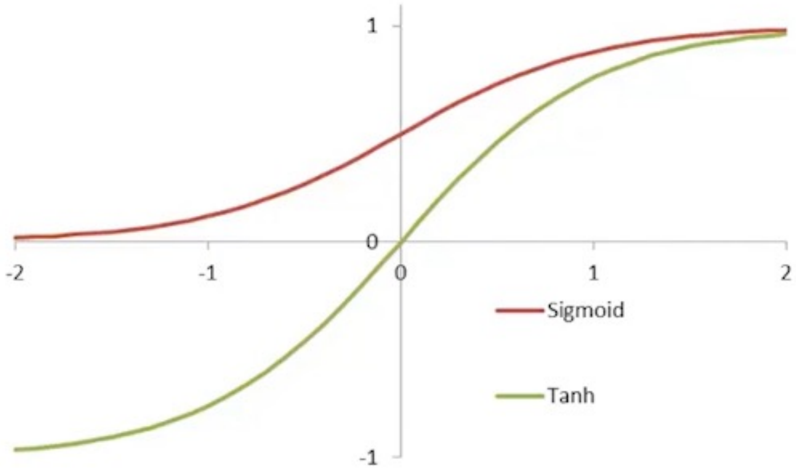
$wx+b$

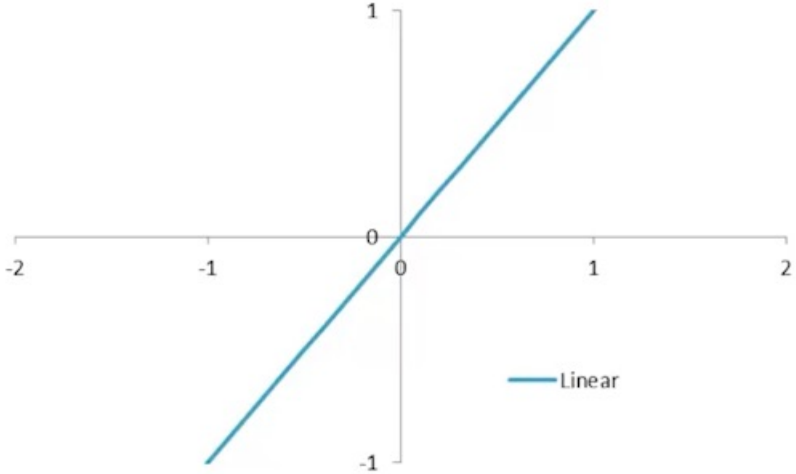


- Linear is nothing but whatever input you have,

  your output is the same as the input. $wx+b$ → Regression Problems
- Among all Activation Functions we have seen so far, this is the only linear one, rest all are non-linear Activation functions.
- The other thing is that the o/p of a linear function can be a large +ve value or a large –ve value, whereas for other Activations functions, the o/p was restricted.
- So basically the Linear function is useful when you want your output to have any value which happens a lot when you have regression problem.

Output activation functions can only be of the following kinds





Sigmoid: Binary Classification output

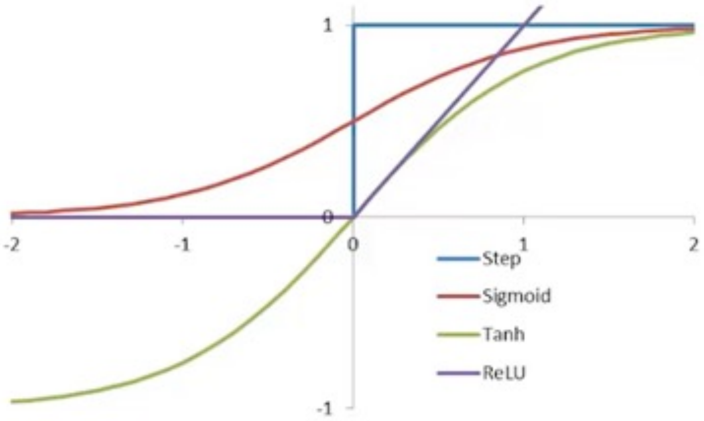
tanh: used in RNN (NLP) [-1, +1]

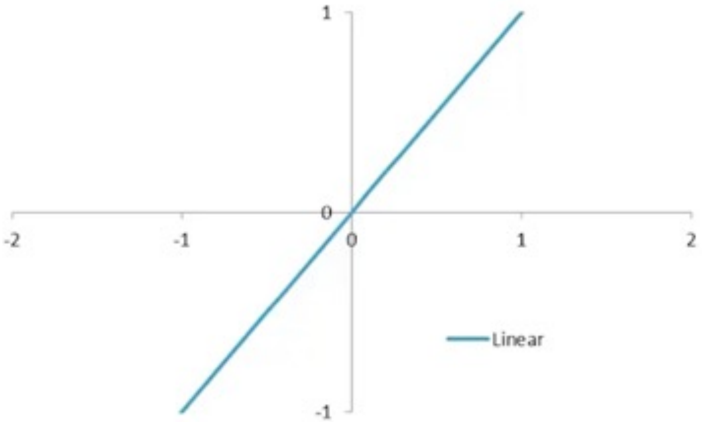
Softmax: It generalizes Sigmoid to many classification (Multiclass)

Linear: Regression

ReLU - used for internal nodes only (hidden layers neurons).

Types of Activation Functions





- **Step**: $g(x) = \frac{\text{sign}(x)+1}{2}$
- **Sigmoid**: $g(x) = \frac{1}{1+e^{-x}}$
- **Tanh**: $g(x) = \tanh(x)$
- **ReLU**: $g(x) = \max(0, x)$
- **Softmax**: $g(x_i) = \frac{e^{x_i}}{\sum_i e^{x_i}}$
- **Linear**: $g(x) = x$