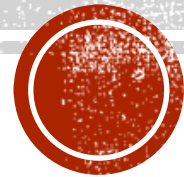


Indian Institute of Information Technology, Allahabad



Introduction to Machine Learning Activation Functions



By

Dr. Shiv Ram Dubey

Associate Professor

Computer Vision and Biometrics Lab

Department of Information Technology

Indian Institute of Information Technology, Allahabad

Web: <https://profile.iita.ac.in/srdubey/>

DISCLAIMER

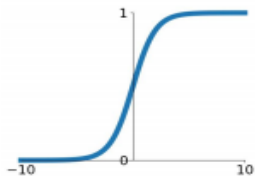
The content (text, image, and graphics) used in this slide are adopted from many sources for academic purposes. Broadly, the sources have been given due credit appropriately. However, there is a chance of missing out some original primary sources. The authors of this material do not claim any copyright of such material.

Non-linearity Layer

Activation Functions

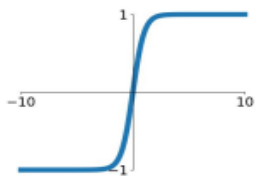
Sigmoid

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



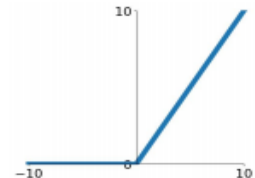
tanh

$$\tanh(x)$$



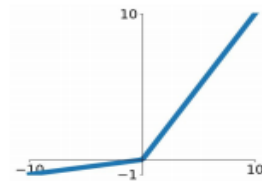
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

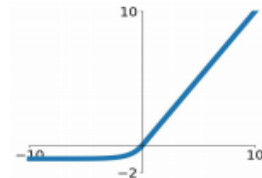


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

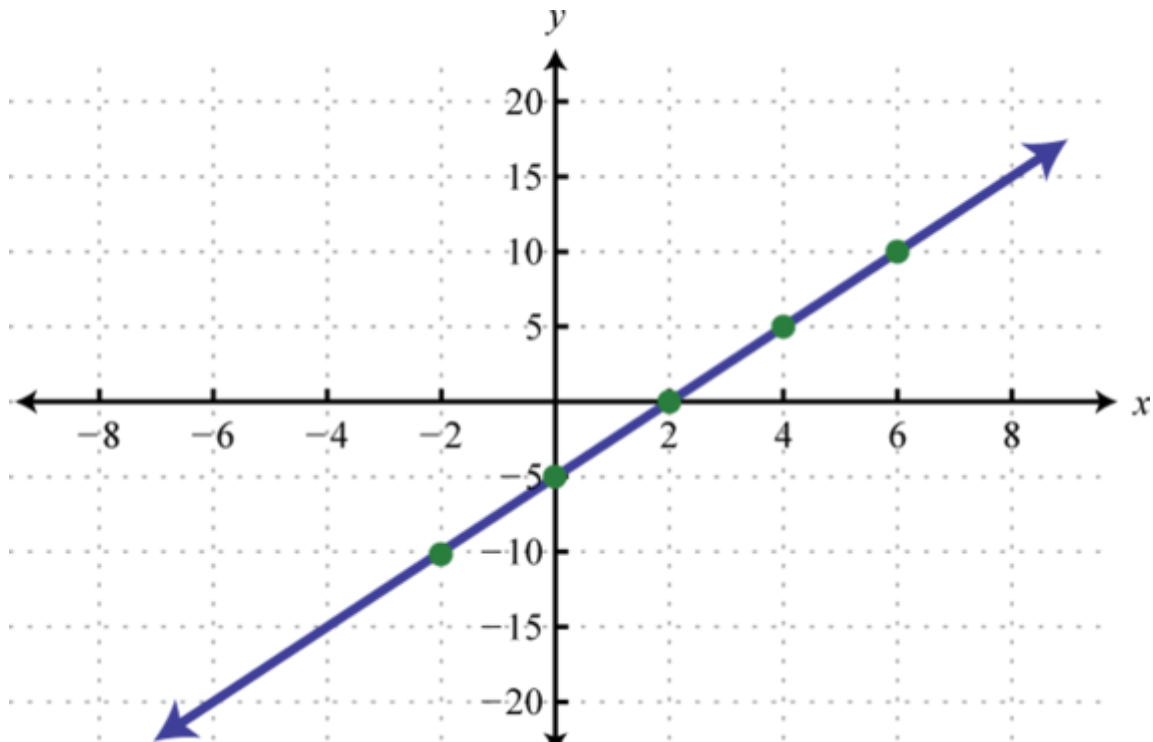
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



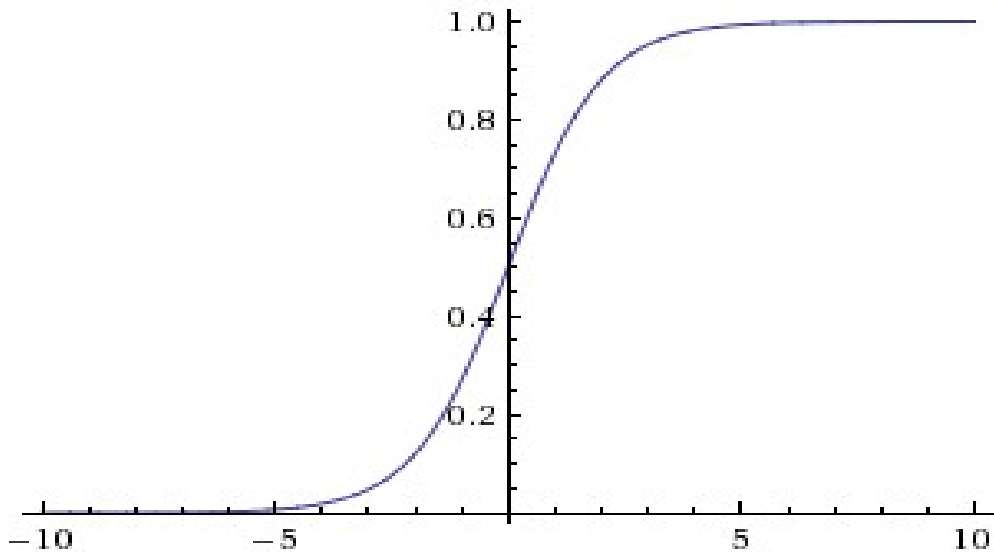
Activation Functions: Linear

- Simplest activation function
- Does not include any non-linearity.



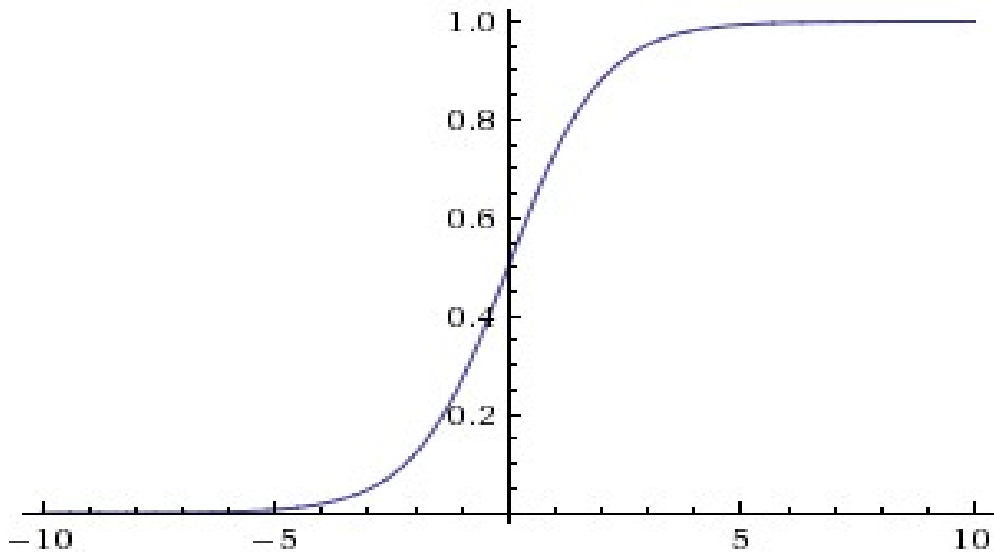
Activation Functions: Sigmoid

$$\sigma(x) = 1/(1 + e^{-x})$$



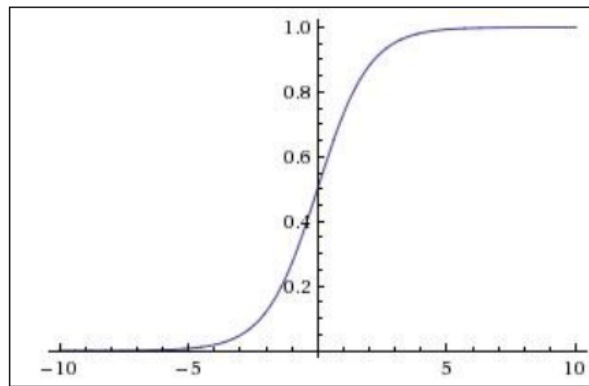
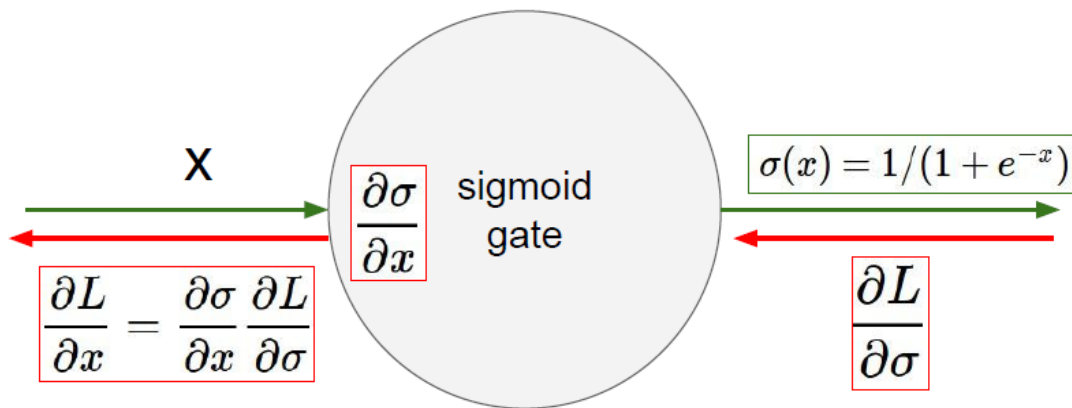
Activation Functions: Sigmoid

$$\sigma(x) = 1/(1 + e^{-x})$$



- Sigmoids saturate and kill gradients.

Activation Functions: Sigmoid



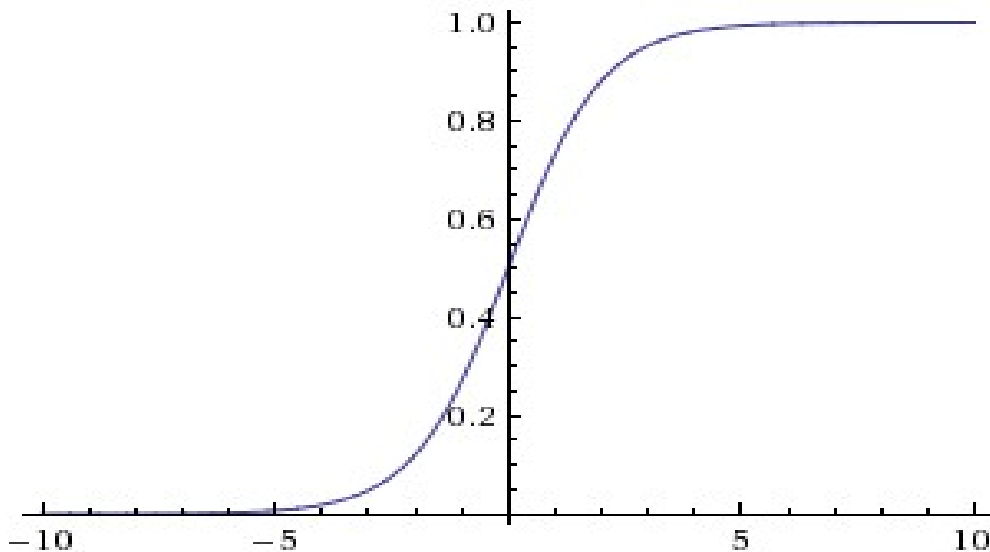
What happens when $x = -10$?

What happens when $x = 0$?

What happens when $x = 10$?

Activation Functions: Sigmoid

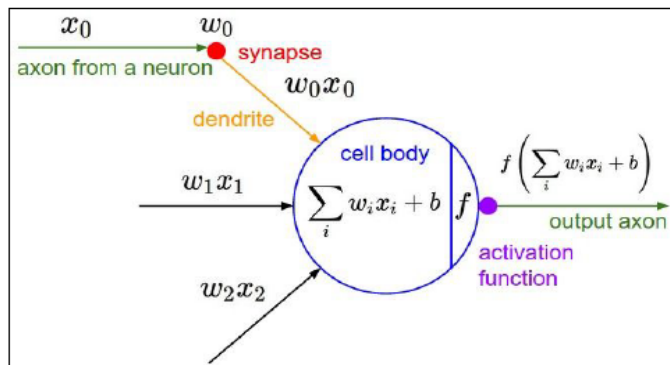
$$\sigma(x) = 1/(1 + e^{-x})$$



- Sigmoids saturate and kill gradients.
- Sigmoid outputs are not zero-centered.

Activation Functions: Sigmoid

Consider what happens when the input to a neuron (x) is always positive:



$$f\left(\sum_i w_i x_i + b\right)$$

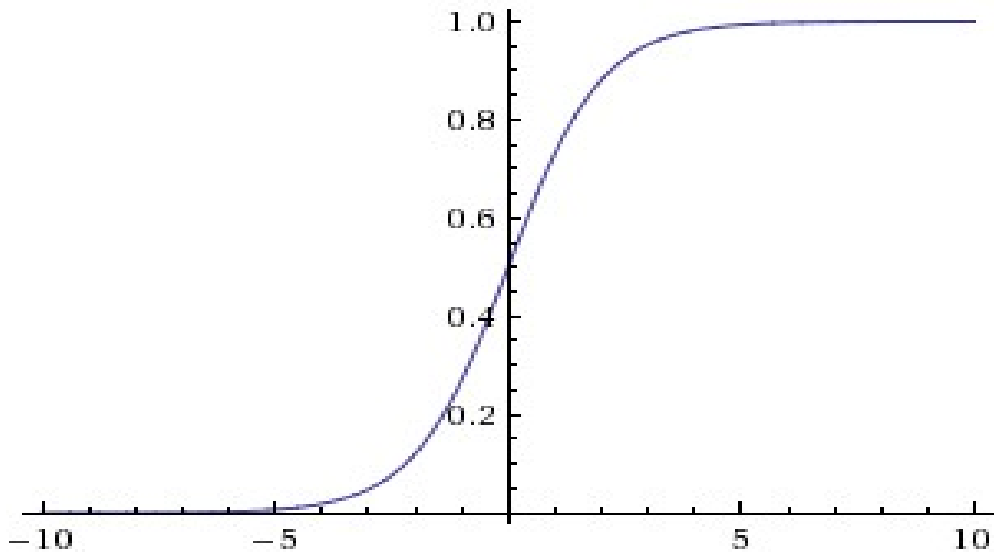
What can we say about the gradients on $\mathbf{w}$?

Always all positive or all negative

(this is also why you want zero-mean data!)

Activation Functions: Sigmoid

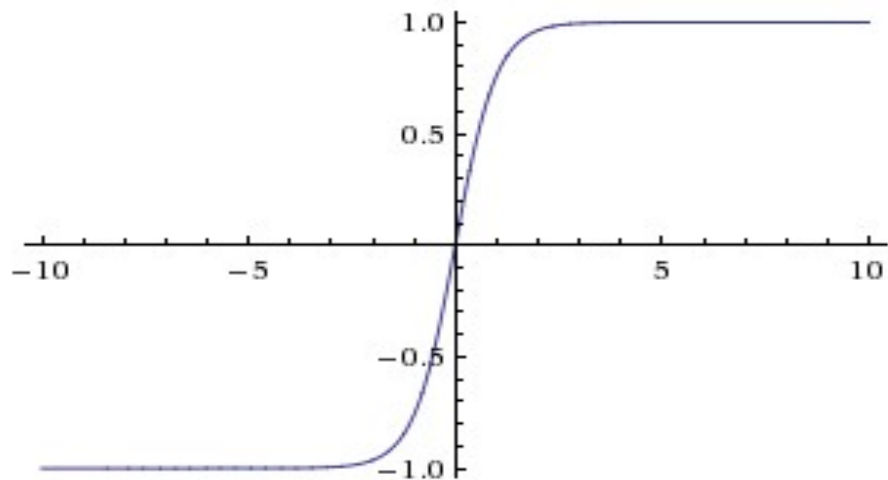
$$\sigma(x) = 1/(1 + e^{-x})$$



- Sigmoids saturate and kill gradients.
- Sigmoid outputs are not zero-centered.
- `Exp()` is a bit compute expensive.

Activation Functions: Tanh

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



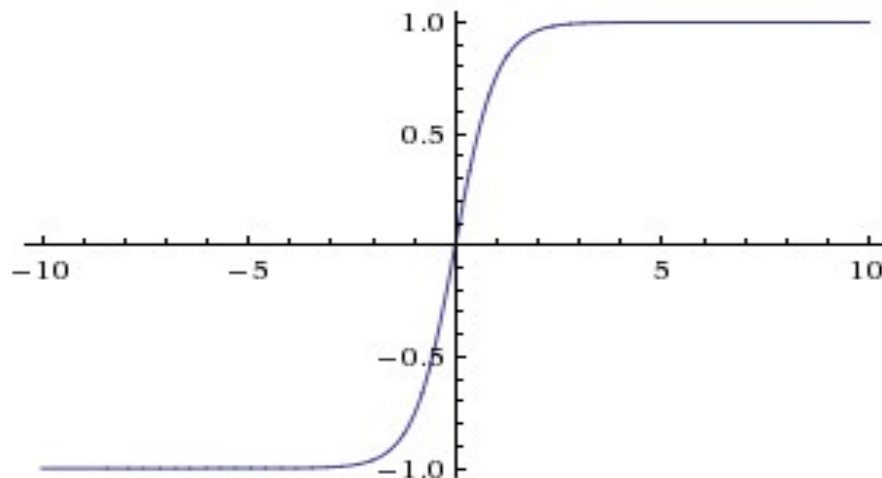
Activation Functions: Tanh

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

tanh neuron is simply a scaled sigmoid neuron

$$\tanh(x) = 2\sigma(2x) - 1.$$

↑
Sigmoid



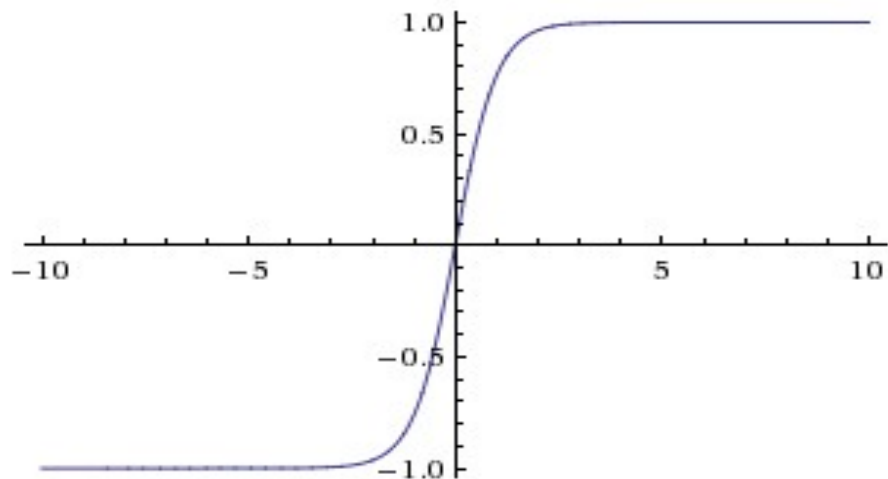
Activation Functions: Tanh

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

tanh neuron is simply a scaled sigmoid neuron

$$\tanh(x) = 2\sigma(2x) - 1.$$

↑
Sigmoid



Like the sigmoid neuron, its activations saturate.

Unlike the sigmoid neuron its output is zero-centered.

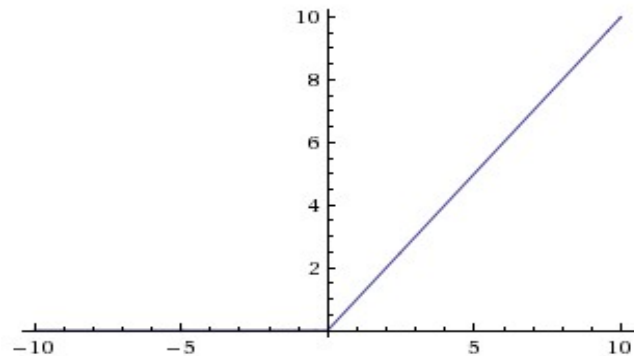
In practice the *tanh non-linearity is always preferred to the sigmoid nonlinearity.*

[LeCun et al., 1991]

Source: <http://cs231n.github.io>

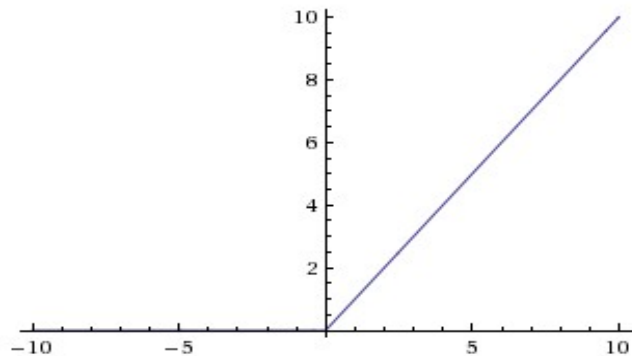
Activation Functions: ReLU

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



Activation Functions: ReLU

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

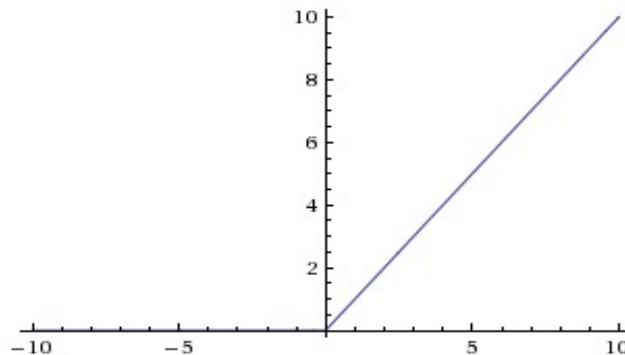


ReLU is 6 times faster in the convergence of stochastic gradient descent compared to the sigmoid/tanh ([Krizhevsky et al.](#)).

ReLU is simple as compared to tanh/sigmoid that involve expensive operations (exponentials, etc.)

Activation Functions: ReLU

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



ReLU is 6 times faster in the convergence of stochastic gradient descent compared to the sigmoid/tanh ([Krizhevsky et al.](#)).

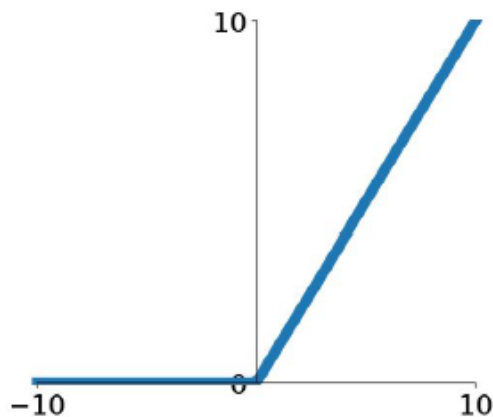
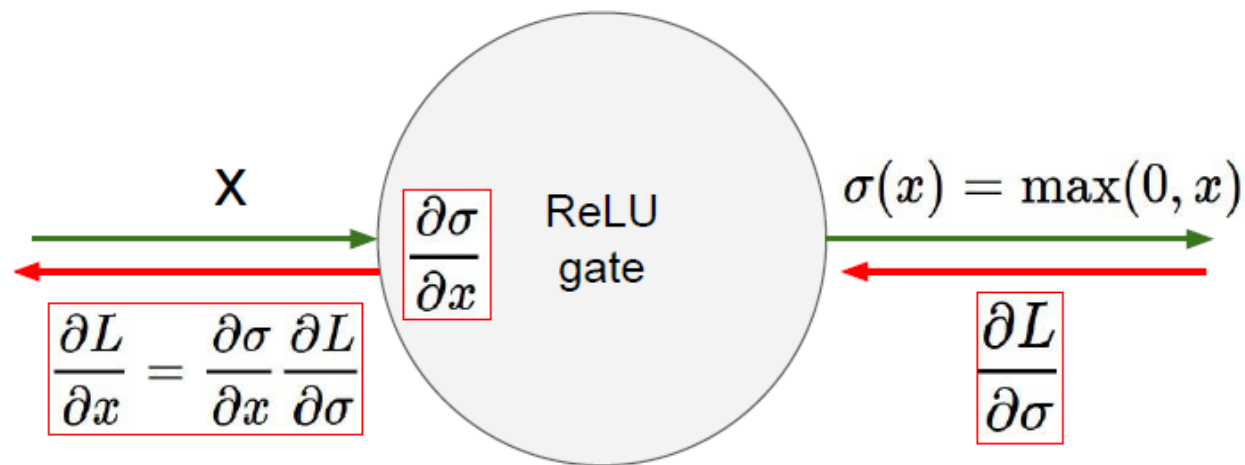
ReLU is simple as compared to tanh/sigmoid that involve expensive operations (exponentials, etc.)

Dying ReLU problem: a large gradient flowing through a ReLU neuron could cause the weights to update in such a way that the neuron will never activate on any datapoint again.

[Krizhevsky et al., 2012]

Source: <http://cs231n.github.io>

Activation Functions: ReLU



What happens when $x = -10$?

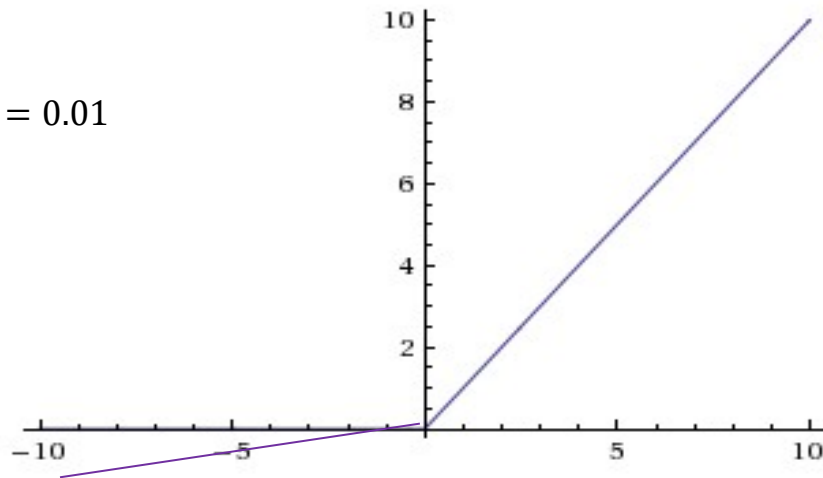
What happens when $x = 0$?

What happens when $x = 10$?

Activation Functions: Leaky ReLU

$$f(x) = \begin{cases} \alpha x, & x < 0 \\ x, & x \geq 0 \end{cases}$$

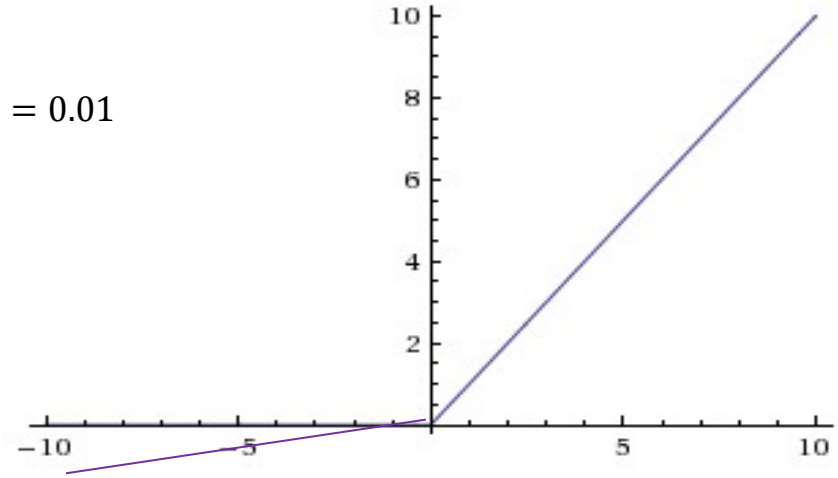
$$\alpha = 0.01$$



Activation Functions: Leaky ReLU

$$f(x) = \begin{cases} \alpha x, & x < 0 \\ x, & x \geq 0 \end{cases}$$

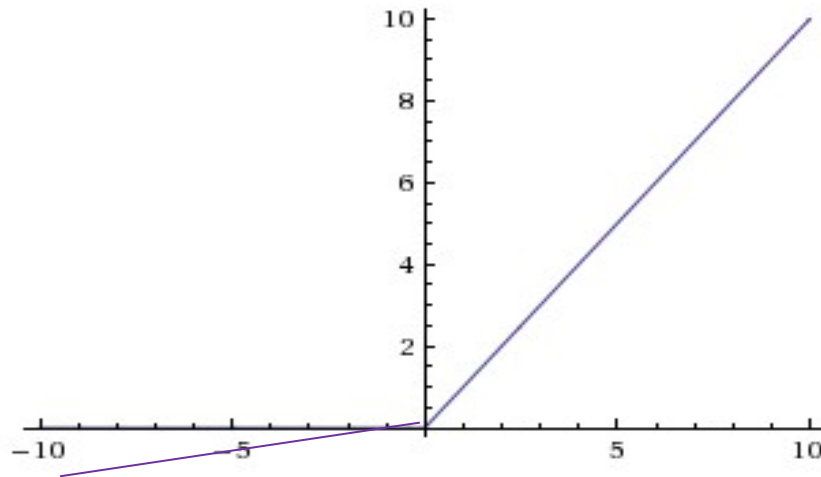
$$\alpha = 0.01$$



Succeeded in some cases, but the results are not always consistent.

Activation Functions: Parametric ReLU

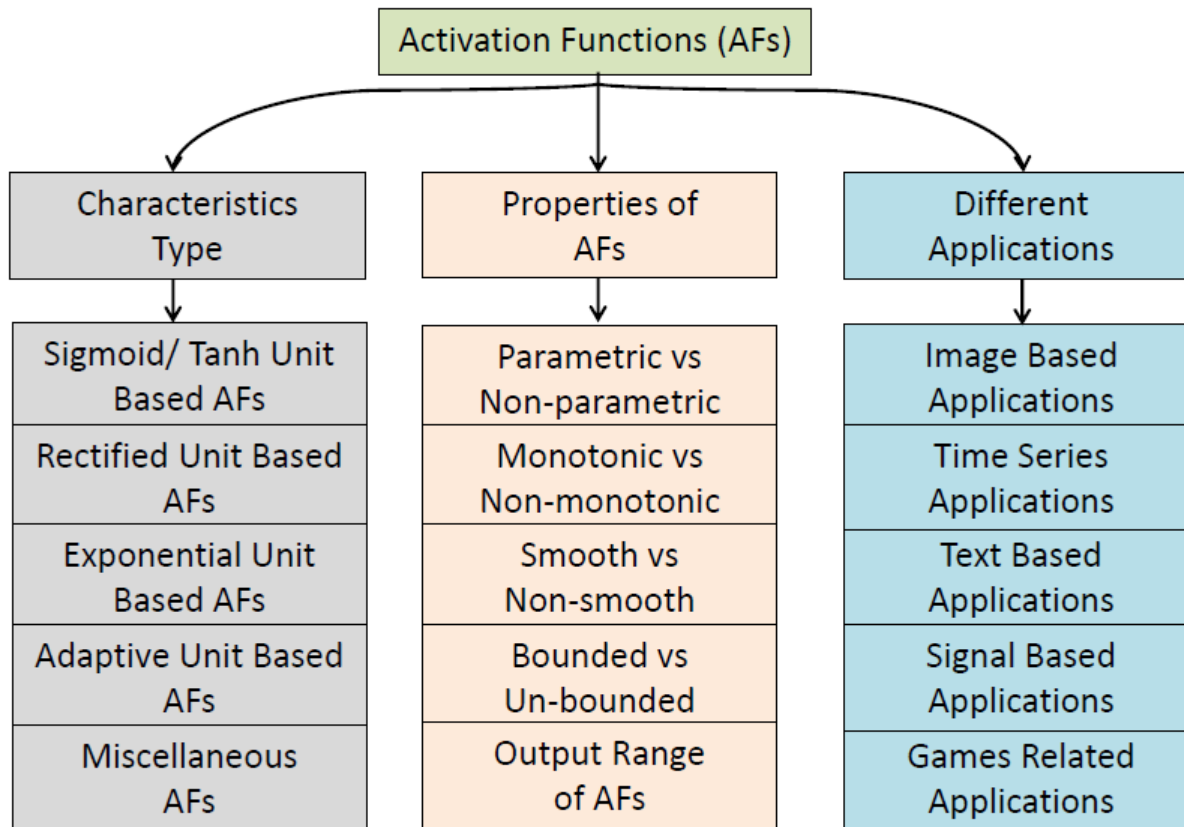
$$f(x) = \begin{cases} \alpha x, & x < 0 \\ x, & x \geq 0 \end{cases}$$



In PReLU, the slope in the negative region is considered as a parameter of each neuron and learnt from data.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *IEEE international conference on computer vision (CVPR)*.

Activation Functions: Classification



Activation Functions: In Practice

- Use ReLU. Be careful with your learning rates
- Try out Leaky ReLU but performance might not be stable
- Try out tanh but don't expect much
- Don't use sigmoid

Acknowledgement

Thanks to the following courses and corresponding researchers for making their teaching/research material online

- Convolutional Neural Networks for Visual Recognition, Stanford University
- Deep Learning, Stanford University
- Introduction to Deep Learning, University of Illinois at Urbana-Champaign
- Introduction to Deep Learning, Carnegie Mellon University
- Natural Language Processing with Deep Learning, Stanford University
- And Many More

THANK YOU