Deep Learning

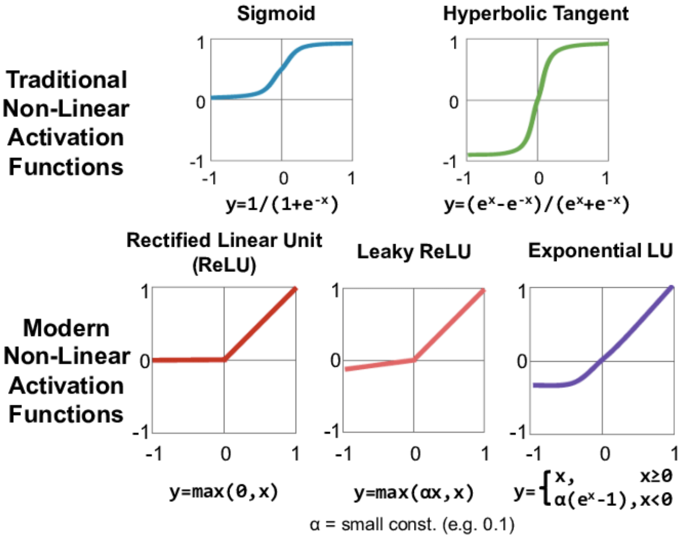
Non-linear activations

Edgar Roman-Rangel.
edgar.roman@itam.mx

Department of Computer Science.
Instituto Tecnológico Autónomo de México, ITAM.

Outline

Comparison


Sigmoid
Hyperbolic Tangent
Traditional Non-Linear Activation Functions
y=1/(1+e^-x)
y=(e^x-e^-x)/(e^x+e^-x)
Rectified Linear Unit (ReLU)
Leaky ReLU
Exponential LU
Modern Non-Linear Activation Functions
y=max(0,x)
y=max(αx,x)
y={x, x≥0; α(e^x-1), x<0
α = small const. (e.g. 0.1)

Linear

$$a = \sum_{n=0}^{N} x_n \omega_n.$$

Derivative:

$$\frac{\partial a}{\partial \omega_i} = x_i.$$

Sigmoid

$$a = \sigma(s) = \frac{1}{1 + e^{-s}}.$$

Derivative:

$$\sigma'(s) = \sigma(s)(1 - \sigma(s)).$$

Tanh

$$a = \frac{e^s - e^{-s}}{e^s + e^{-s}}.$$

Derivative:

$$a' = 1 - a^2.$$

ReLU

Logistic functions suffer of the so-called *vanishing gradient* issue. Rectified Linear Units (ReLU) are an alternative:

$$a = \begin{cases} 0, & s < 0, \\ s, & s \geq 0. \end{cases}$$

Derivative:

$$a' = \begin{cases} 0, & s < 0, \\ 1, & s \geq 0. \end{cases}$$

They are also more efficient computationally speaking.

Leaky ReLU

$$a = \begin{cases} \alpha s, & s < 0, \\ s, & s \geq 0, \end{cases}$$

with $0 < \alpha < 1$.

Derivative:

$$a' = \begin{cases} \alpha, & s < 0, \\ 1, & s \geq 0. \end{cases}$$

ELU: Exponential Linear Unit

$$a = \begin{cases} \alpha(e^s - 1), & s < 0, \\ s, & s \geq 0, \end{cases}$$

with $0 < \alpha < 1$.

Derivative:

$$a' = \begin{cases} a + \alpha, & s < 0, \\ 1, & s \geq 0. \end{cases}$$

SELU: Scaled Exponential Linear Units

$$a = \begin{cases} \lambda s, & s \geq 0, \\ \lambda\alpha(e^s - 1), & s < 0, \end{cases}$$

with $\lambda = 1.0507$ and $\alpha = 1.6733$.

Derivative:

$$a' = \begin{cases} \lambda, & s \geq 0, \\ \lambda\alpha e^s, & s < 0. \end{cases}$$

Softmax

All previous functions are element-wise operations. Softmax is a vector normalizer.

$$a_i = \frac{e^{s_i}}{\sum_j e^{s_j}}.$$

Derivative:

$$a_i' = a_i(1 - a_i).$$

Used to exaggerate the most probable of the elements of the vector. Useful in output layers for multi-class classification problems.

Common use scenarios

Activation	Use
ReLU (or variants)	All hidden layers in all scenarios.
Sigmoid (tanh)	Output layer for binary classification.
Sigmoid	Output layer for regression with $0 \leq y \leq 1$.
Tanh	Output layer for regression with $-1 \leq y \leq 1$.
Linear	Output layer for unbounded regression.
Softmax	Output layer for multi-class classification.

To know more

Bishop, C. M. and Bishop, H. "Deep Learning Foundations and Concepts". Chapter 6. Springer. 2024.

Q&A

Thank you!

edgar.roman@itam.mx