Neural Network Activation Functions: a small subset!

ReLU	GELU /	PReLU
$\max(0,x)$	$\frac{x}{2}\left(1+\tanh\left(\sqrt{\frac{2}{\pi}}\right)(x+ax^3)\right)$	$\max(0,x)$
$\begin{cases} x \text{ if } x > 0 \end{cases}$	Swish	SELU $\alpha(\max(0,x)+$
$\alpha(x \exp x - 1) \text{ if } x < 0$ SoftPlus	$1 + \exp{-x}$ Mish	$\min(0, \beta(\exp x - 1)))$
$\frac{1}{\beta}\log\left(1+\exp(\beta x)\right)$	$x \tanh\left(\frac{1}{\beta}\log\left(1 + \exp(\beta x)\right)\right)$	$\begin{cases} x \text{ if } x \ge 0\\ ax \text{ if } x < 0 \text{ with } a \sim \Re(l, u) \end{cases}$
HardSwish $\begin{cases} 0 \text{ if } x \le -3 \\ x \text{ if } x \ge 3 \\ x(x+3)/6 \text{ otherwise} \end{cases}$	Sigmoid $\frac{1}{1 + \exp(-x)}$	SoftSign x $1 + x $
Tanh $tanh(x)$	Hard tanh $a \text{ if } x \ge a$ $b \text{ if } x \le b$ $x \text{ otherwise}$	Hard Sigmoid $\begin{cases} 0 \text{ if } x \leq -3 \\ 1 \text{ if } x \geq 3 \\ x/6 + 1/2 \text{ otherwise} \end{cases}$
Tanh Shrink	Soft Shrink	Hard Shrink
$x - \tanh(x)$	$\begin{cases} x - \lambda \text{ if } x > \lambda \\ x + \lambda \text{ if } x < -\lambda \\ 0 \text{ otherwise} \end{cases}$	$\begin{cases} x \text{ if } x > \lambda \\ x \text{ if } x < -\lambda \\ 0 \text{ otherwise} \end{cases}$

Neural Network Activation Functions:

1 ReLU (Rectified Linear Unit):

Formula: f(x) = max(0, x)

Pros: Simple, computationally efficient, helps mitigate the vanishing gradient problem.

Cons: Can lead to "dying ReLUs" where neurons become inactive and stop learning.

2 GELU (Gaussian Error Linear Unit):

Formula: $f(x) = x * P(X \le x)$, where P is the cumulative distribution function of the standard normal distribution.

Pros: Smooth, differentiable, combines properties of ReLU and dropout, improves performance in NLP tasks.

Cons: Computationally more intensive compared to ReLU.

3 Sigmoid:

Formula: $f(x) = 1 / (1 + \exp(-x))$

Pros: Outputs probabilities, useful for binary classification.

Cons: Prone to vanishing gradient problem, slow convergence.

4 Tanh (Hyperbolic Tangent):

Formula: f(x) = (exp(x) - exp(-x)) / (exp(x) + exp(-x))

 $\label{pros:pros:zero-centered} Pros: Zero-centered, less likely to saturate than sigmoid.$

Cons: Still suffers from vanishing gradients, slower training.

5 Leaky ReLU:

Formula: f(x) = x if x > 0 else alpha * x

Pros: Addresses the "dying ReLU" problem, allows a small gradient when inactive.

Cons: Introduces a small negative slope, which may not always be optimal.

6 ELU (Exponential Linear Unit):

Formula: f(x) = x if x > 0 else alpha * (exp(x) - 1)

Pros: Smooth, reduces the bias shift by pushing mean activations closer to zero.

Cons: More computationally intensive than ReLU, introduces an additional hyperparameter.

7 Swish:

Formula: f(x) = x / (1 + exp(-x))

Pros: Smooth, differentiable, improves performance in deep networks.

Cons: More computationally intensive than ReLU.

8 Softplus:

Formula: f(x) = log(1 + exp(x))

Pros: Smooth, always positive, avoids the problem of zero gradients.

Cons: Computationally intensive, can lead to vanishing gradients for large negative inputs.

Choosing the right activation function can significantly impact your model's performance.