**Different types of Activation Functions**

Activation functions play a critical role in neural networks by introducing non-linearity, which allows networks to learn complex patterns. Here are some common types of activation functions:

### 1. \*\*Sigmoid (Logistic) Activation Function\*\*

- \*\*Formula\*\*: \[ f(x) = \frac{1}{1 + e^{-x}} \]

- \*\*Range\*\*: (0, 1)

- \*\*Description\*\*: The sigmoid function squashes input values into a range between 0 and 1, making it useful for binary classification problems.

- \*\*Drawbacks\*\*: Can suffer from the \*\*vanishing gradient problem\*\* and has slow convergence since gradients near 0 are small.

### 2. \*\*Hyperbolic Tangent (Tanh) Activation Function\*\*

- \*\*Formula\*\*: \[ f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \]

- \*\*Range\*\*: (-1, 1)

- \*\*Description\*\*: The tanh function is similar to sigmoid but outputs between -1 and 1. It’s often preferred over sigmoid as it centers the data, helping to mitigate the vanishing gradient problem slightly.

- \*\*Drawbacks\*\*: Still can suffer from the vanishing gradient problem with deep networks.

### 3. \*\*Rectified Linear Unit (ReLU) Activation Function\*\*

- \*\*Formula\*\*: \[ f(x) = \max(0, x) \]

- \*\*Range\*\*: \[ [0, \infty) \]

- \*\*Description\*\*: ReLU is a popular activation function due to its simplicity and effectiveness. It outputs the input directly if it’s positive; otherwise, it outputs zero.

- \*\*Advantages\*\*: Reduces the likelihood of the vanishing gradient problem, making training faster.

- \*\*Drawbacks\*\*: Can lead to \*\*"dying ReLU"\*\* where neurons get stuck in the zero region and stop learning.

### 4. \*\*Leaky ReLU Activation Function\*\*

- \*\*Formula\*\*: \[ f(x) = \begin{cases}

x & x > 0 \\

\alpha x & x \leq 0

\end{cases} \]

- \*\*Range\*\*: \[ (-\infty, \infty) \]

- \*\*Description\*\*: A variant of ReLU that allows a small, non-zero gradient when \( x \leq 0 \), where \( \alpha \) is a small constant (e.g., 0.01).

- \*\*Advantages\*\*: Helps address the dying ReLU problem by allowing small negative values.

### 5. \*\*Parametric ReLU (PReLU) Activation Function\*\*

- \*\*Formula\*\*: \[ f(x) = \begin{cases}

x & x > 0 \\

\alpha x & x \leq 0

\end{cases} \]

- \*\*Range\*\*: \[ (-\infty, \infty) \]

- \*\*Description\*\*: Similar to Leaky ReLU but with the parameter \( \alpha \) learned during training.

- \*\*Advantages\*\*: Adaptive; the network learns the best \( \alpha \) for each neuron.

### 6. \*\*Exponential Linear Unit (ELU) Activation Function\*\*

- \*\*Formula\*\*: \[ f(x) = \begin{cases}

x & x > 0 \\

\alpha (e^x - 1) & x \leq 0

\end{cases} \]

- \*\*Range\*\*: \[ (-\alpha, \infty) \]

- \*\*Description\*\*: ELU functions are similar to ReLU but allow for negative saturation when \( x \leq 0 \). \( \alpha \) is typically set to 1.

- \*\*Advantages\*\*: ELU tends to converge faster and produce better accuracy than ReLU in deep networks due to reduced mean activations closer to zero.

### 7. \*\*Softmax Activation Function\*\*

- \*\*Formula\*\*: \[ f(x\_i) = \frac{e^{x\_i}}{\sum\_{j=1}^{n} e^{x\_j}} \]

- \*\*Range\*\*: (0, 1) for each output, with all outputs summing to 1

- \*\*Description\*\*: The softmax function is commonly used in the output layer for multiclass classification problems. It converts logits into a probability distribution.

- \*\*Advantages\*\*: Useful for multiclass classification because it provides probabilities for each class.

### 8. \*\*Swish Activation Function\*\*

- \*\*Formula\*\*: \[ f(x) = x \cdot \sigma(x) = x \cdot \frac{1}{1 + e^{-x}} \]

- \*\*Range\*\*: \((- \infty, \infty)\)

- \*\*Description\*\*: A newer activation function proposed by Google, where \( \sigma(x) \) is the sigmoid function.

- \*\*Advantages\*\*: Shows better performance in some deep networks due to its smoothness and non-linearity.

### 9. \*\*Gaussian Activation Function\*\*

- \*\*Formula\*\*: \[ f(x) = e^{-x^2} \]

- \*\*Range\*\*: \[ (0, 1] \]

- \*\*Description\*\*: Common in certain specialized neural networks, this function outputs values close to zero for large inputs, with the peak value at 1 for \( x = 0 \).

- \*\*Use Case\*\*: Often used in radial basis function (RBF) networks for specific types of pattern recognition tasks.

### Summary Table

| Activation Function | Range | Common Use Case |

|---------------------|-----------------|-------------------------------------|

| \*\*Sigmoid\*\* | (0, 1) | Binary classification |

| \*\*Tanh\*\* | (-1, 1) | Centered data, mitigates vanishing gradients |

| \*\*ReLU\*\* | [0, ∞) | General deep networks |

| \*\*Leaky ReLU\*\* | (-∞, ∞) | Solves dying ReLU problem |

| \*\*PReLU\*\* | (-∞, ∞) | Learnable alpha, more flexible |

| \*\*ELU\*\* | (-α, ∞) | Reduces mean activations near zero |

| \*\*Softmax\*\* | (0, 1) (sums to 1) | Multiclass classification |

| \*\*Swish\*\* | (-∞, ∞) | Some deep networks, smooth gradient |

| \*\*Gaussian\*\* | (0, 1] | RBF networks, specialized tasks |

Each activation function has strengths suited to different types of networks and problems, so choosing one depends on the architecture and task.