Activation Functions in Neural Network

Role of Activation Function

- Even though our neural network has a very complex configuration of weights, it will not be able to solve a problem without the activation function. The reason for this lies in the concept of Non Linearity.
- The activation function plays a crucial role in artificial neural networks (ANNs) by introducing non-linearity into the model.
- Following are the key aspects of its role:

1. **Non-Linearity Introduction:** The primary role of the activation function is to introduce non-linear properties to the network.
2. **Decision Making:** The activation function determines the output of a neuron given an input or set of inputs. It decides whether a neuron should be activated or not, based on the input received and the weights applied. This decision-making process helps the network to approximate complex functions and make better predictions.
3. **Output Mapping:** Activation functions map the input signal to an output signal in a specific range. For example, the sigmoid function maps inputs to a range between 0 and 1.

The Impact of Not Using Activation Functions in Neural Networks

- If we do not use any activation function in a neural network, the output of each neuron would simply be a linear function of its inputs. In other words, the network would be reduced to a **linear regression model**, which can only model linear relationships between input and output data.
- Activation functions are essential in neural networks because they introduce nonlinearity into the network, enabling it to model complex, nonlinear relationships in the data. Without activation functions, the network would be limited to modeling only linear relationships, which would greatly restrict its ability to learn and recognize complex patterns in the data.

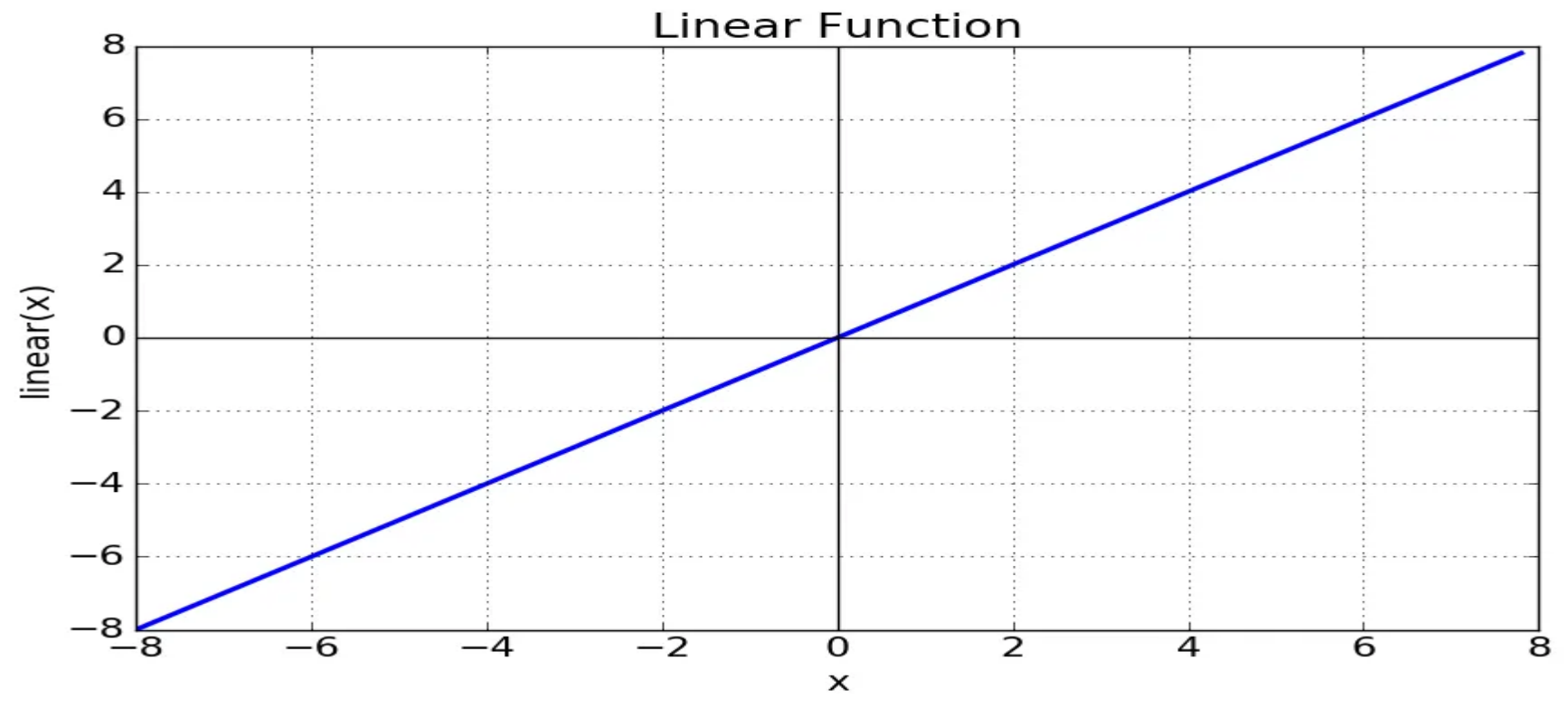
Types of Activation Functions

- Activation functions play a crucial role in determining the output of a neural network, helping it make decisions like “yes” or “no”.
- These functions also map resulting values to specific ranges, such as 0 to 1 or -1 to 1, depending on the chosen function.
- There are two main types of activation functions: Linear Activation Functions and Non-linear Activation Functions.
- Non-linear activation functions play important role in the classification model of the neural networks to understand complex and non-linear relationship among data.

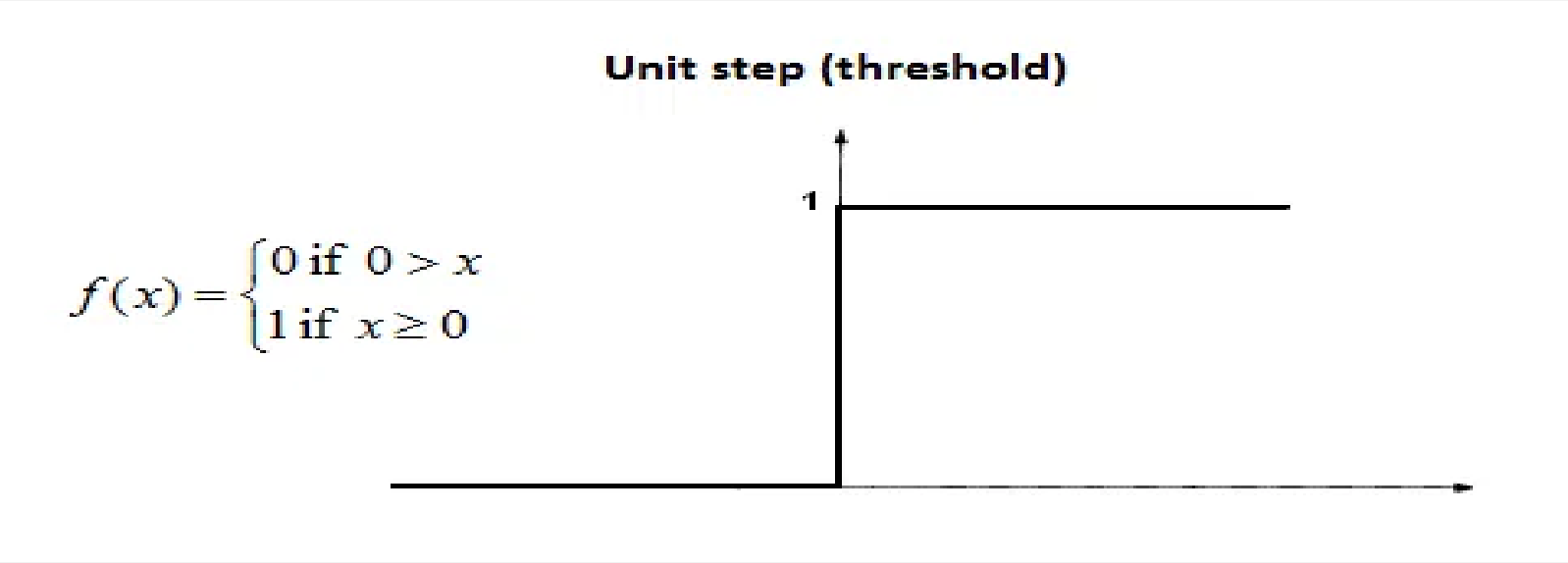
Linear Activation Function

- It is a simple straight line activation function where our function is directly proportional to the weighted sum of neurons or input.
- It is better in giving wide-range of activation function and a line of positive slope may increase the firing rate as input rate increases.
- Generalized equation: $f(x) = x$ and $f`(x) = 1$
- Ranges from -∞ to +∞

- Limitation:
- Gradient is constant.
- It is not possible to use back-propagation as the derivative of a function since its constant.



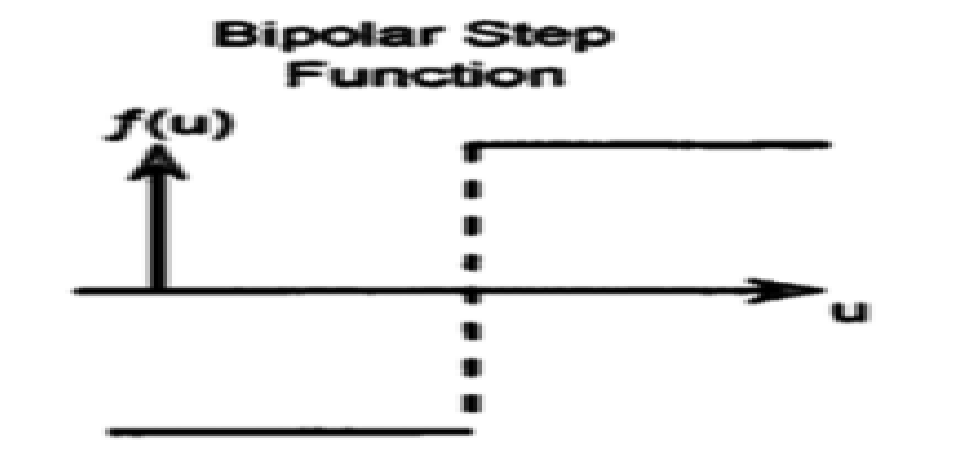
Binary Step function (Heaviside Step Function or Unit Step Function)



- If the value of Y is above a certain value, declare it activated. If it's less than the threshold, then say it's not.
- It is great for binary classifier.
- It is not suitable for Gradient Descent algorithm for weight optimization.
- It cannot be used for multi-class classification.

Bipolar Step Fuction

- In the Bipolar Step Function, if the value of Y is above a certain value known as the threshold, the output is +1and if it’s less than the threshold then the output is -1.
- It has bipolar outputs (+1 to -1). It can be utilized in single-layer networks.



$$f(x) = \begin{cases} 1 \text{ if } x >= t \\ -1 \text{ if } x < t \end{cases}$$

Widely used Activation Functions

- There are several commonly used activation functions in neural networks, including:

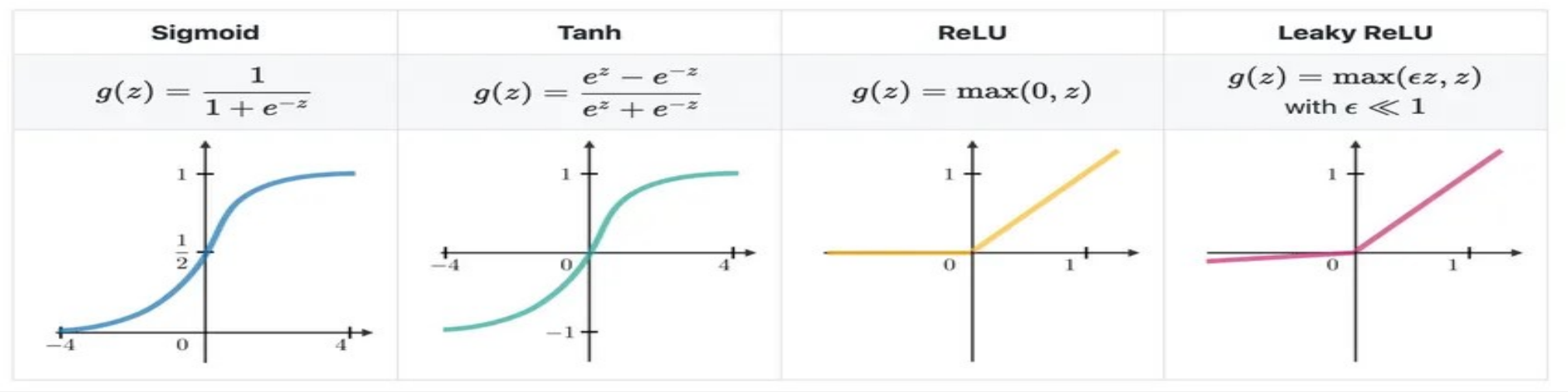
1. Sigmoid: The sigmoid activation function maps any input to the range of 0 and 1, producing an output that can be interpreted as a probability.
2. Tanh (Hyperbolic Tangent): The Tanh activation function maps its inputs to the range of -1 and 1, producing outputs with zero mean and unit variance. This makes it useful for normalizing the output of a neuron, which can improve the performance of the network.

Continued (Activation Functions)

3. Softmax: The softmax activation is typically used as the final activation function in a neural network for multiclass classification problems. It maps its inputs to a probability distribution over multiple classes.
4. ReLU (Rectified Linear Unit): The ReLU activation function sets any negative input to 0 and retains positive inputs unchanged. This function has become widely popular in deep learning due to its computational efficiency and ability to avoid the vanishing gradient problem.

Continued (Activation Functions)

5. Leaky ReLU: It is similar to the ReLU function but allows a small gradient for negative inputs, preventing neurons from dying (i.e., outputting zero for all inputs).
6. Swish: The Swish is a recent activation function that has been shown to outperform ReLU on some tasks. It is defined as x * sigmoid(x).



Sigmoid activation Function

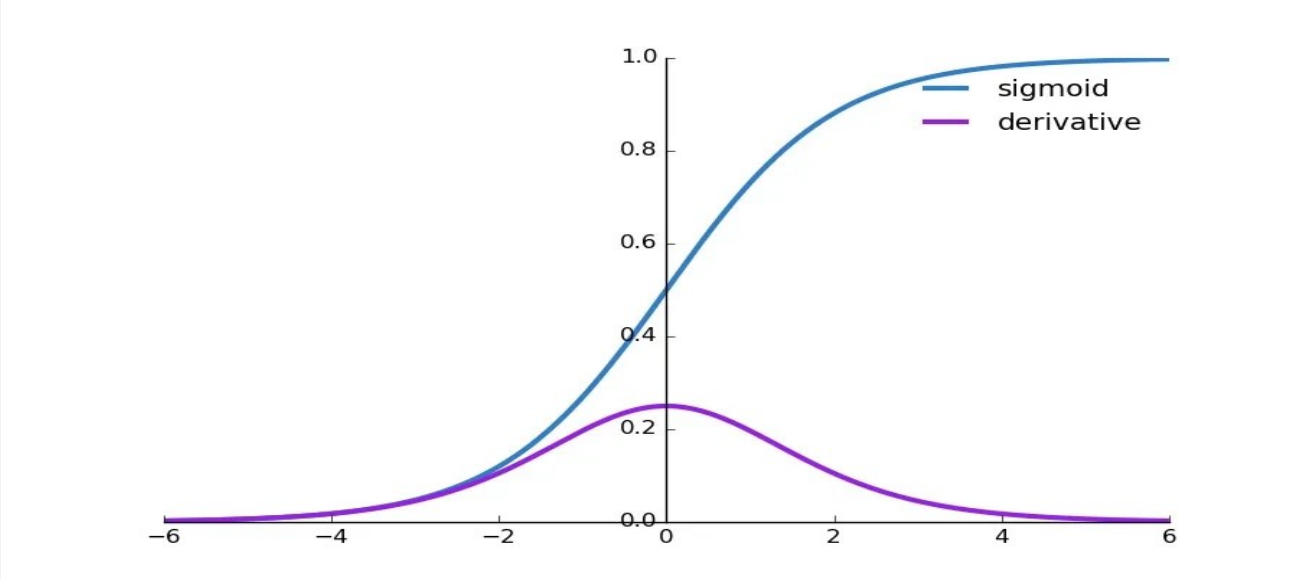
- The sigmoid function is a mathematical function that maps any input value to a value between 0 and 1. It is often used as an activation function in artificial neural networks to introduce nonlinearity into the model.
- The sigmoid function has the following mathematical form:

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

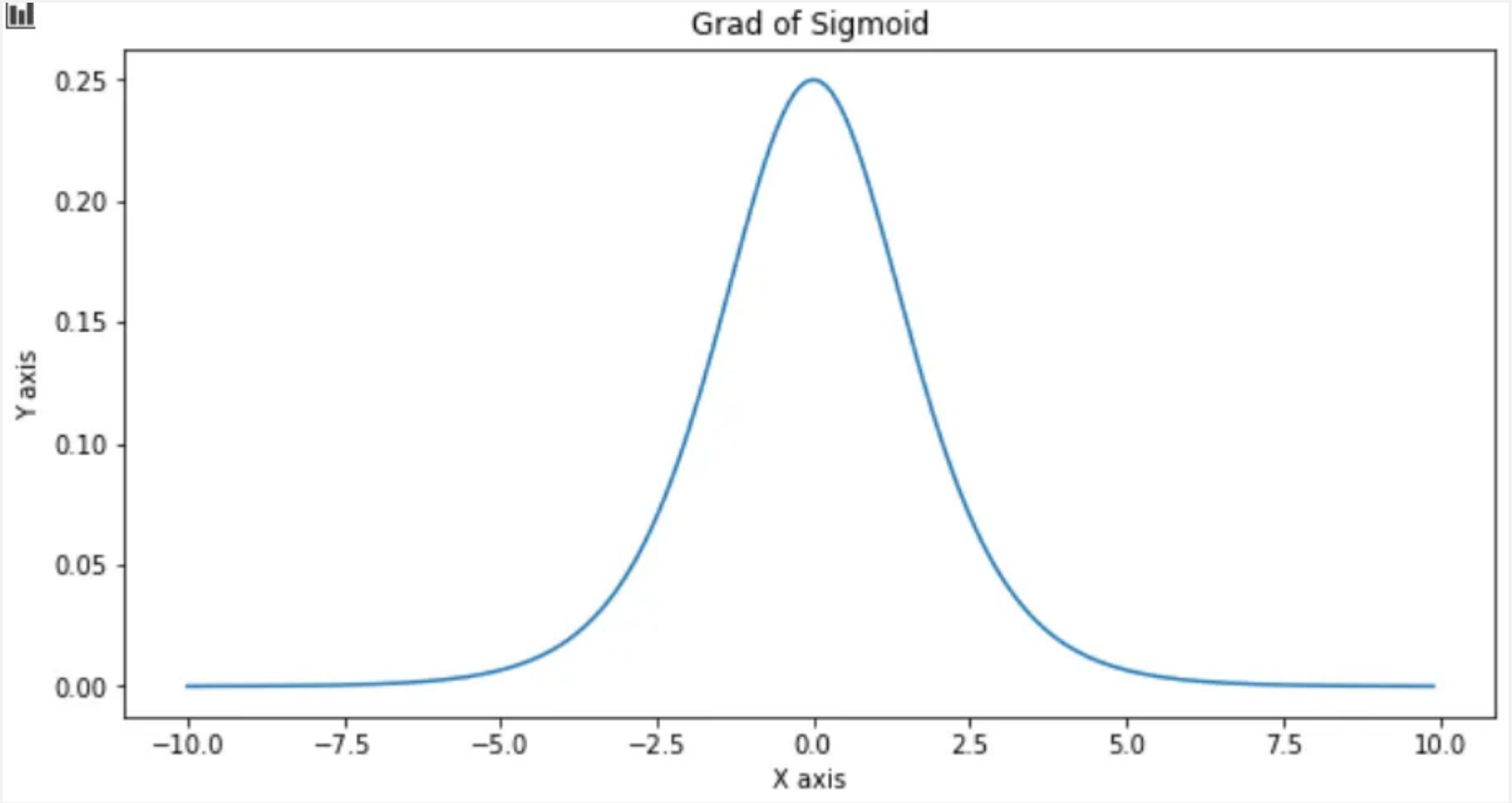
- where x is the input to the function, e is the base of the natural logarithm, and f(x) is the output of the function.

Sigmoid Function Derivative

- The sigmoid function can be used to introduce nonlinearity into the model and allow the neural network to make probabilistic predictions for binary classification problems.



Gradient of Sigmoid



It is seen in the gradient graph, Sigmoid suffers from vanishing gradient problem. The input changes from -10 to -5 and 5 to 10, but the gradient output doesn’t change.

Advantages of the Sigmoi Activation Function:

- Non-linear: The sigmoid function is a non-linear function, which means it can model complex relationships between inputs and outputs. This is useful in machine learning where we often encounter non-linear relationships in data.
- Smoothness: The sigmoid function is a smooth function that has a continuous derivative. This means that it is easy to optimize using gradient descent, a popular optimization algorithm in machine learning.
- Interpretability: The sigmoid function outputs a probability value between 0 and 1, which can be interpreted as the likelihood of an event occurring. This makes it easy to interpret the output of a machine learning model that uses the sigmoid function.

Disadvantages of the Sigmoid Function:

- Vanishing gradient: The derivative of the sigmoid function is maximum at the center point of the curve (where the output is 0.5) and decreases as we move towards the extremes of 0 and 1. This can cause the gradient to become very small (i.e., “vanish”) during backpropagation, which can slow down or even prevent learning in deep neural networks.
- Susceptible to saturation: The sigmoid function can saturate (i.e., output values very close to 0 or 1), which can make it difficult to optimize the network. This is particularly a problem in deep neural networks where the inputs to the sigmoid function can become very large or small, causing the function to saturate. In this case, other activation functions such as ReLU or LeakyReLU may be more effective.

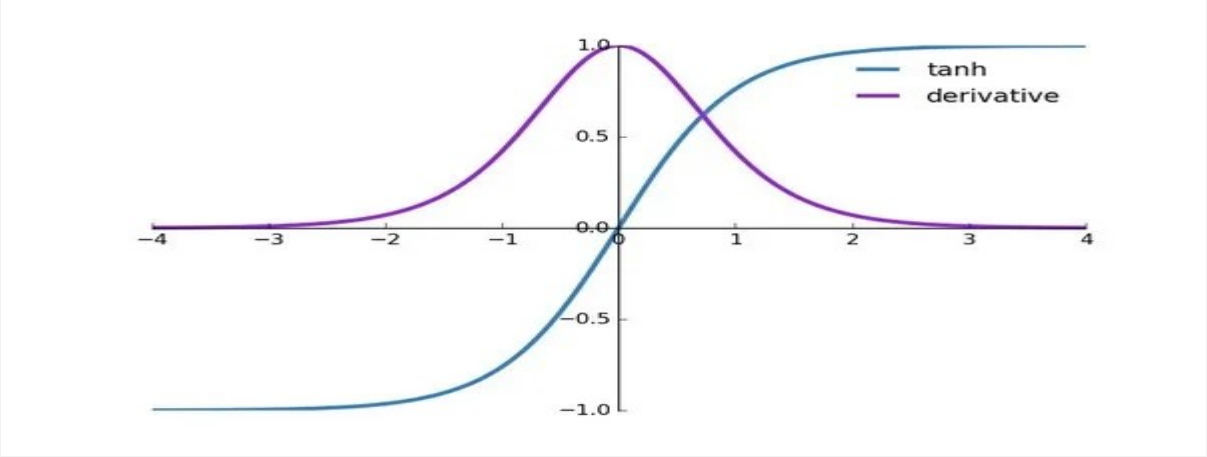
Hyperbolic tangent activation function (tanh)

- The hyperbolic tangent activation function, also known as “tanh”, is a widely used activation function in artificial neural networks.
- It is a non-linear function that takes a real-valued number as input and maps it to a value between -1 and 1.
- The formula for tanh is:

$$\frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

- The shape of the tanh function is similar to the sigmoid function, but the range is from -1 to 1 instead of 0 to 1.
- The tanh function is symmetric around the origin, which means that for negative inputs, it produces negative outputs, and for positive inputs, it produces positive outputs. The output is 0 when the input is 0.

- The derivative of tanh with respect to its input x is given by: $f'(x) = \tanh'(x) = 1 - \tanh^2(x) = 1 - f(x)^2$
- The derivative of the tanh function is steeper than the sigmoid function around the origin, which can make it more suitable for certain types of neural networks, such as those used for image processing.
- The tanh function is often used as an activation function in the hidden layers of neural networks because it is differentiable, non-linear, and has a bounded output. It can help neural networks learn complex non-linear relationships between input and output data.

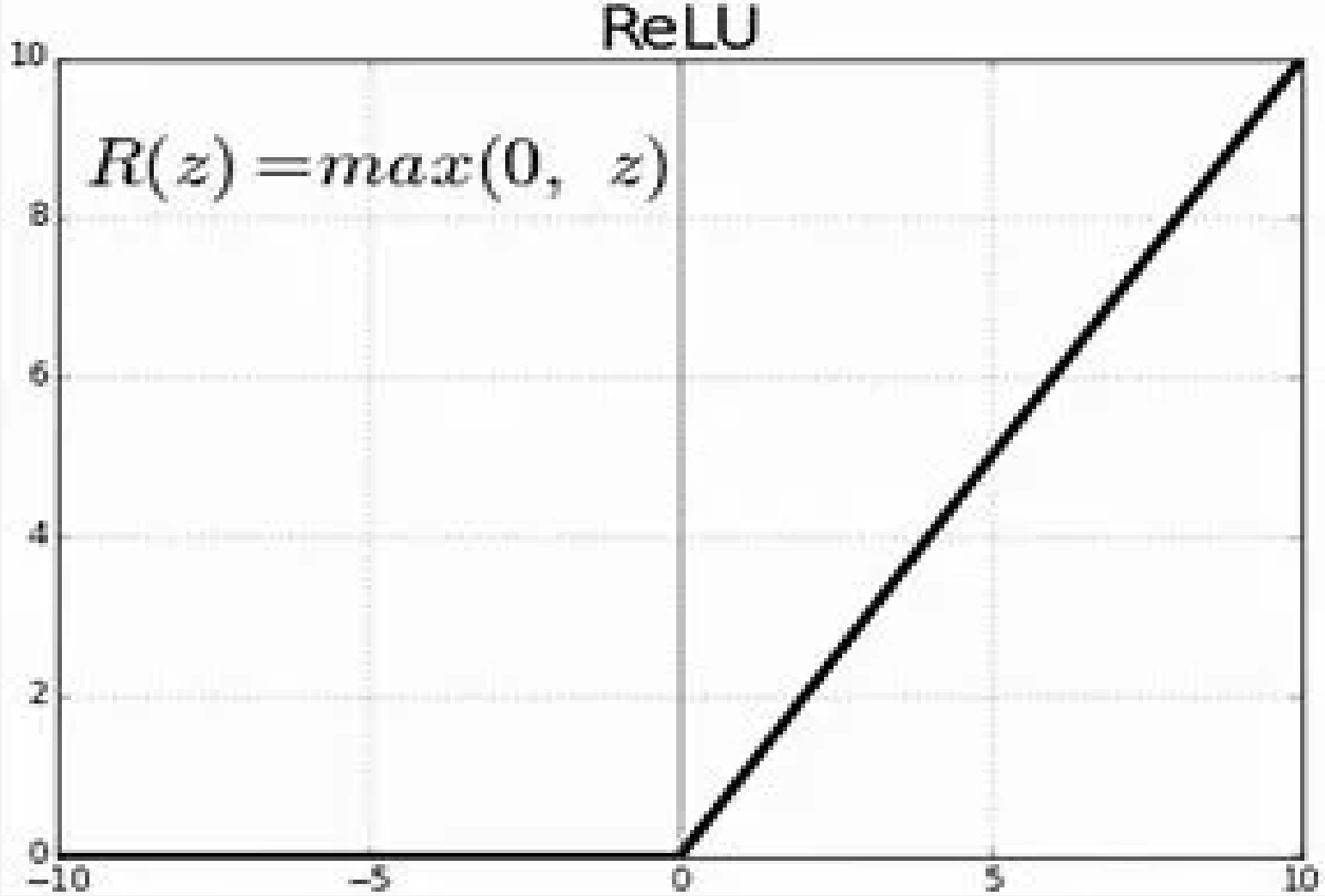


- The derivative of the tanh function is smooth and well-behaved, which can help improve the stability and convergence of neural network training algorithms.
- However, one disadvantage of the tanh function is that it suffers from the “vanishing gradient” problem, which means that the gradients become very small as the absolute value of the input increases.
- This can make it difficult for neural networks to learn effectively with the tanh activation function in deep networks. To address this issue, other activation functions, such as the Rectified Linear Unit (ReLU) and its variants, have been developed.

The ReLU (Rectified Linear Unit)

- The ReLU (Rectified Linear Unit) activation function is a commonly used activation function in deep learning, particularly in neural networks.
- It is a simple yet effective activation function that is computationally efficient and has been shown to perform well in many applications.
- The ReLU function is defined as follows: $f(x) = \max(0, x)$
- In other words, if the input value x is greater than or equal to zero, then the output of the function is simply x. If the input value is negative, however, the output is zero.
- The function is flat for negative input values, and linear for positive input values.

ReLU Activation Function

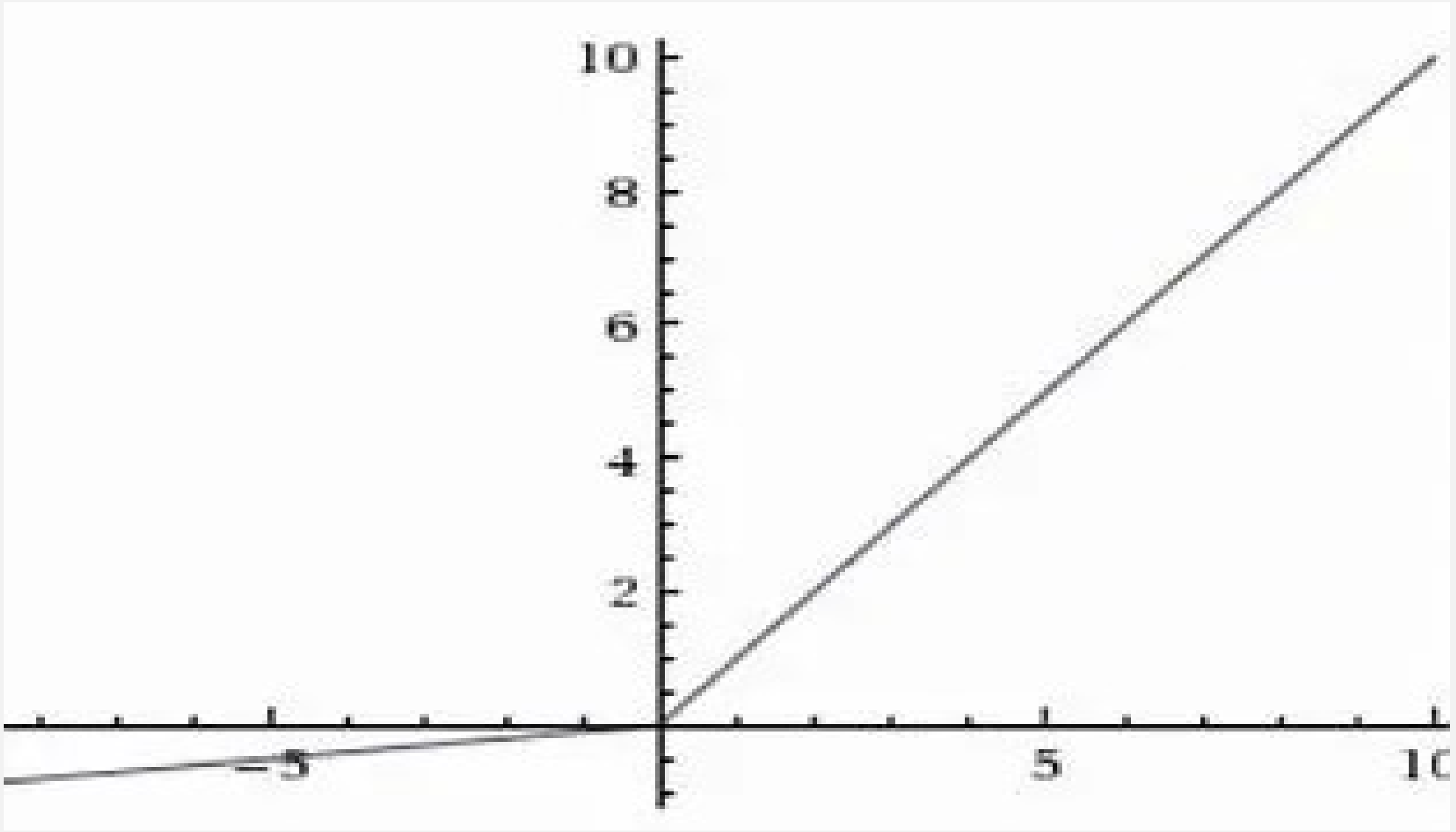


- The main advantage of the ReLU function is that it helps to address the problem of vanishing gradients that can occur when using other activation functions, such as the sigmoid or tanh functions.
- When using these functions, the gradients can become very small as the input values become very large or very small, which can slow down the learning process and make it difficult to train deep neural networks.
- By contrast, the ReLU function has a constant gradient of 1 for positive input values, which helps to speed up the learning process and make it easier to train deep neural networks.
- Additionally, the ReLU function is computationally efficient and easy to implement, which makes it a popular choice for many applications.

- However, one potential drawback of the ReLU function is that it can lead to “dead” neurons, where the output of the function is always zero for a particular neuron.
- This can occur if the weights of the neuron become very negative, which can cause the input to the ReLU function to always be negative.
- The “**dying ReLU**” problem: is a common issue that can occur when using the ReLU (Rectified Linear Unit) activation function in deep neural networks. It happens when the output of a ReLU neuron becomes zero, and the neuron is no longer able to produce a gradient during backpropagation. This means that the weights of the neuron are no longer updated during training, effectively making the neuron “dead” and unable to contribute to the network’s output.
- To address this problem, several variants of the ReLU function have been proposed, such as the leaky ReLU and the exponential ReLU, which can help to prevent dead neurons from occurring.

Leaky ReLU

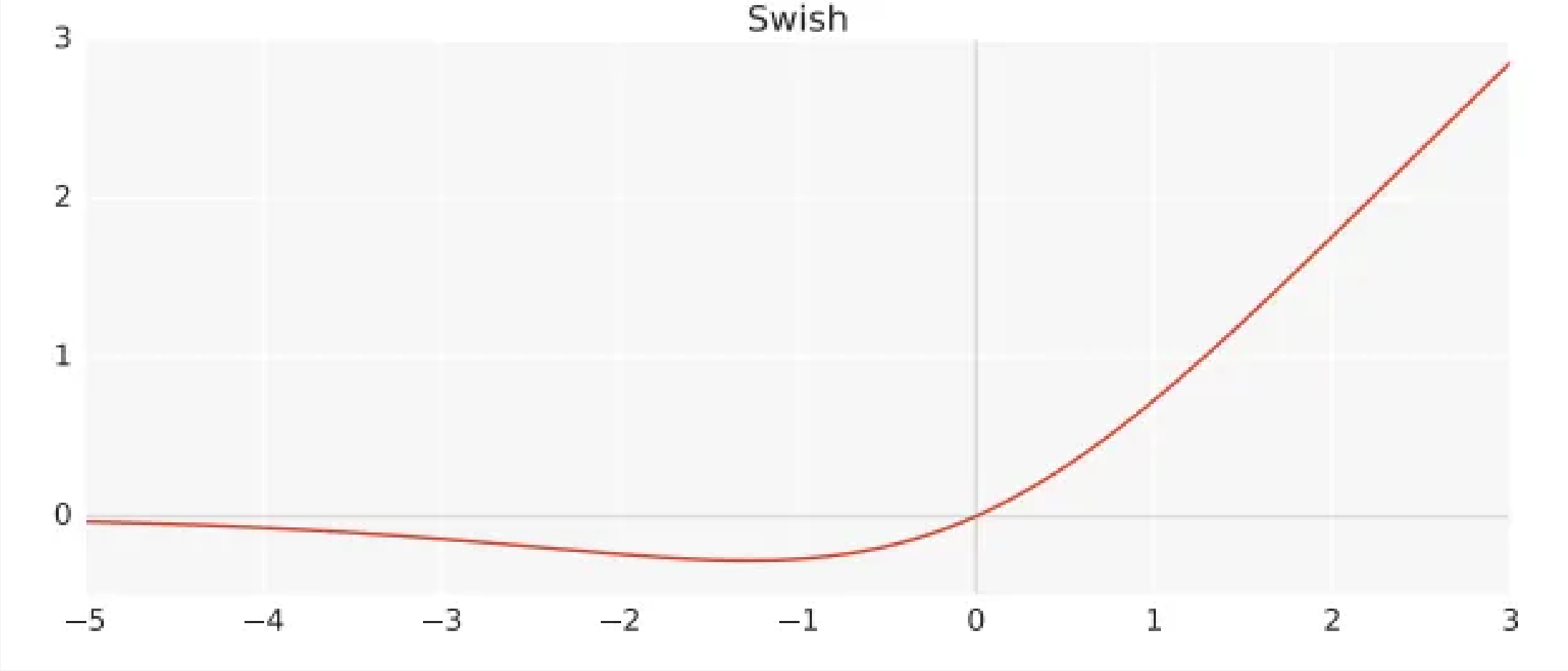
- Leaky ReLU is a variant of the ReLU function that avoids the dying ReLU problem by allowing negative values to have a small slope instead of being entirely flat.
- The formula for Leaky ReLU is $f(x) = \max(ax, x)$, where a is a small constant (typically 0.01).
- This ensures that even if the input is negative, there will still be a small gradient, allowing for learning to occur.


10
8
6
4
2
−5
5
10

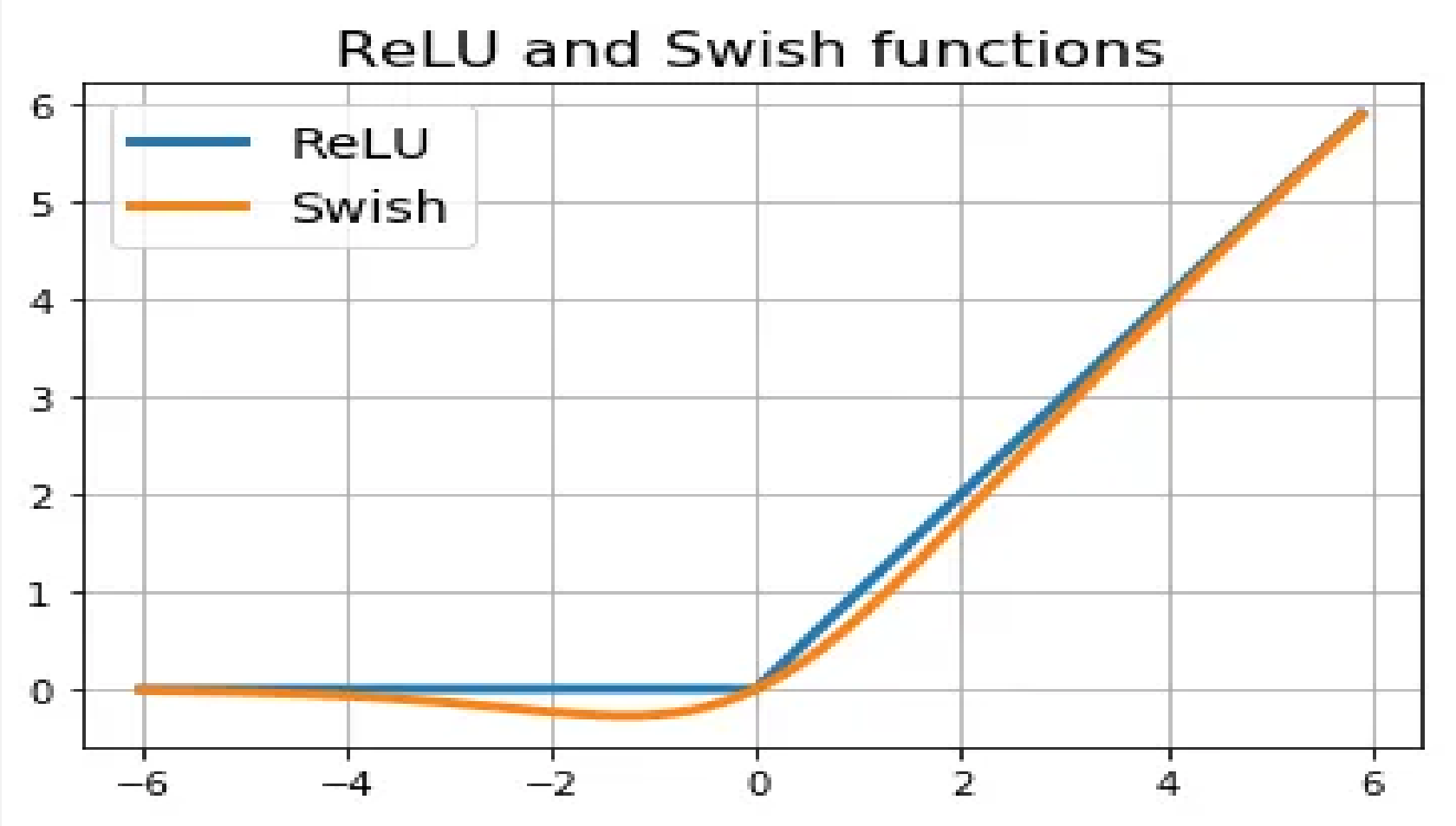
Swish Activation Function

- Google Brain Team has proposed a new activation function, named Swish.
- The formula for Swish: $f(x) = x \cdot sigmoid(x)$
- Their experiments show that Swish tends to work better than ReLU on deeper models across a number of challenging data sets.
- The simplicity of Swish and its similarity to ReLU make it easy for practitioners to replace ReLUs with Swish units in any neural network.
- Swish is a smooth, non-monotonic function that consistently matches or outperforms ReLU on deep networks.

Swish Activation Graph



ReLU and Swish

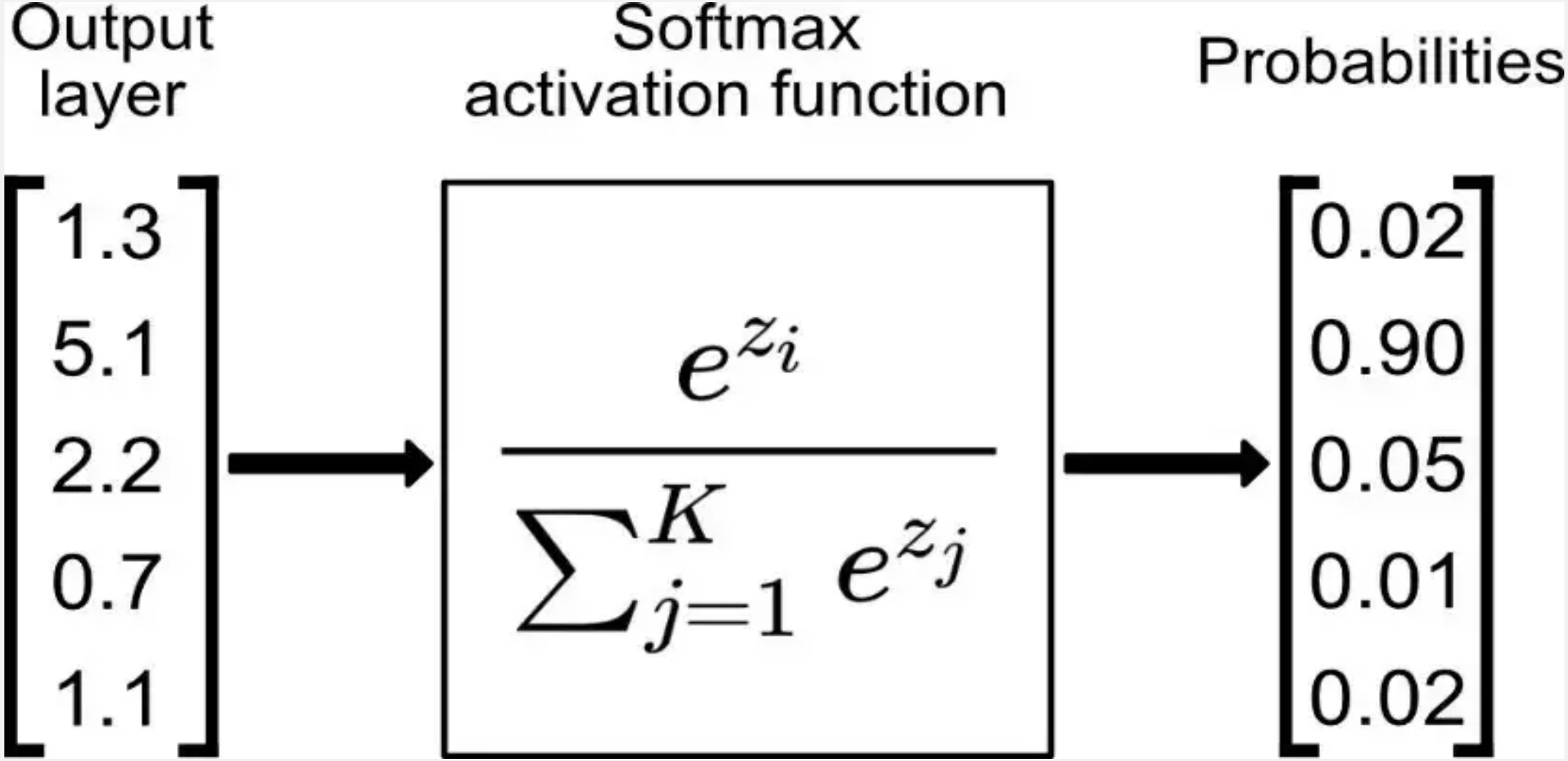


Softmax Activation Function

- Softmax activation function is used at output layer of the neural network model that converts a vector of raw values into a probability distribution.
- Softmax function describes as a combination of multiple sigmoid function.
- Softmax is used at multi-class classification problems.
- Softmax is differentiable, making it suitable for use in gradient-based optimization methods such as backpropogation algorithm.

$$p_i = \left(\frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}\right)$$

Softmax Activation Function



Summary of Activation Functions

- Activation Functions are used to introduce non-linearity in the network.
- A neural network will almost always have the same activation function in all hidden layers and a different one on the output layer depending upon the type of output of the model.
- ReLU is the most commonly used activation function for hidden layers.
- Regarding the output layer, we must always consider the expected value range of the predictions. If it can be any numeric value (as in the case of the regression problem) you can use the linear activation function or ReLU.
- Use Softmax or Sigmoid function for the classification problems.