

2-3 激活函数

王中雷

厦门大学王亚南经济研究院和经济学院, 2025

内容摘要

1. Sigmoid激活函数

2. Tanh激活函数

3. ReLU激活函数

4. Leaky ReLU激活函数

为什么需要激活函数？

1. 神经网络模型中，每个神经元涉及两个运算
 - 线性变换
 - 激活（非线性变换）
2. 如果没有激活（非线性变换），神经网络就是一个简单的线性模型
3. 激活运算使神经网络能够学习复杂的函数关系

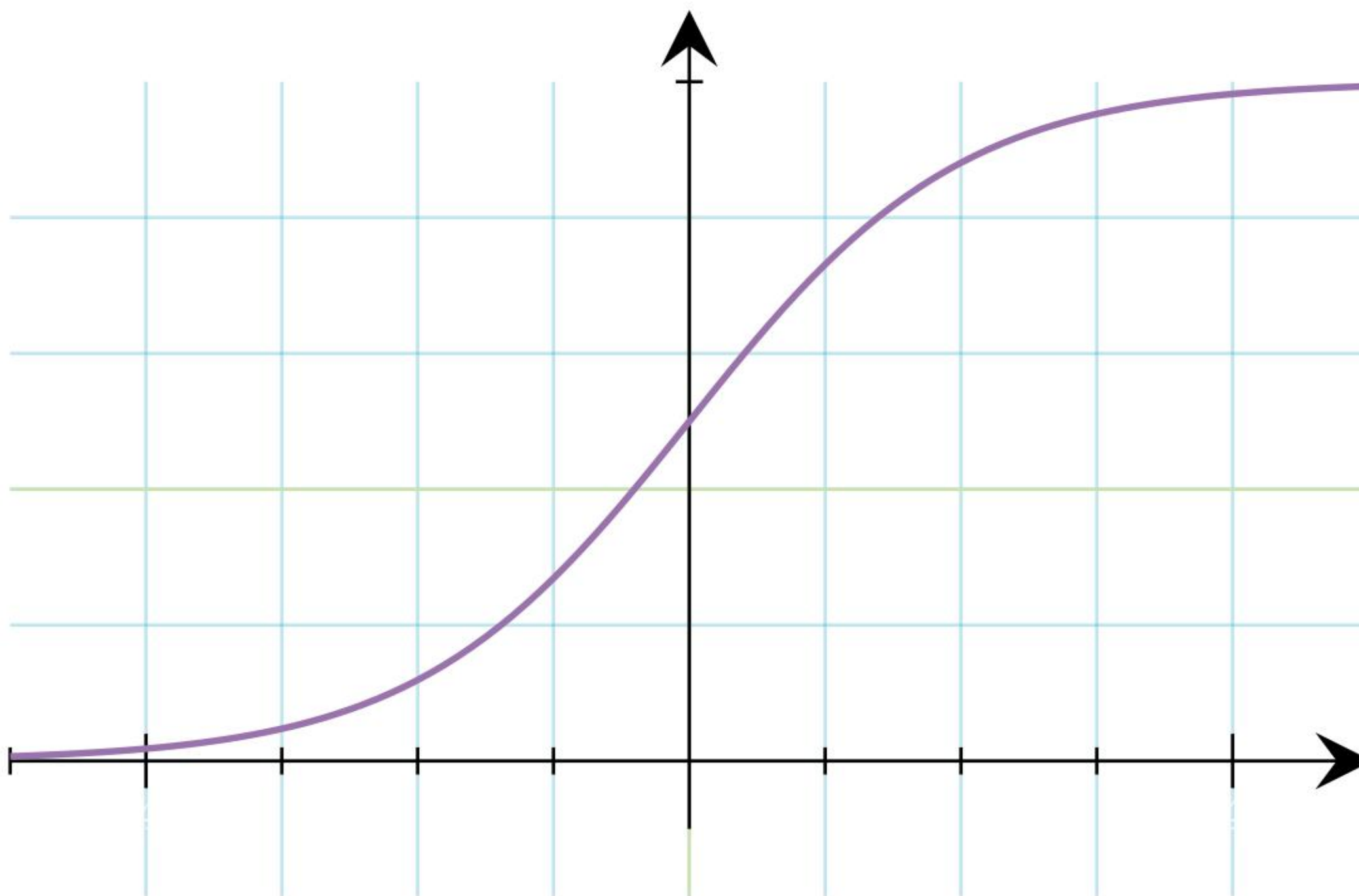
Sigmoid激活函数

1. 形式

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

- 值域: $(0, 1)$
- 导数: $\sigma'(z) = \sigma(z)\{1 - \sigma(z)\}$

Sigmoid激活函数



Sigmoid激活函数

1. 优点

- 值域为 $(0, 1)$ ，其通常用于二分类问题中对概率的估计（输出层）
- 当 z 比较大时，梯度消失（实际上不是太好）；因此，其对极端值比较稳健
- 对于后向传播而言，其导数计算简洁

2. 缺点

- 梯度消失，进而导致参数更新较慢
 - ▷ 对于一个 5 层神经网络模型，对应梯度会缩小 $0.25^5 \approx 0.001!$
- 输出不是关于 0 对称，可能导致“协变量”偏移
- 由于指数导致计算效率低下

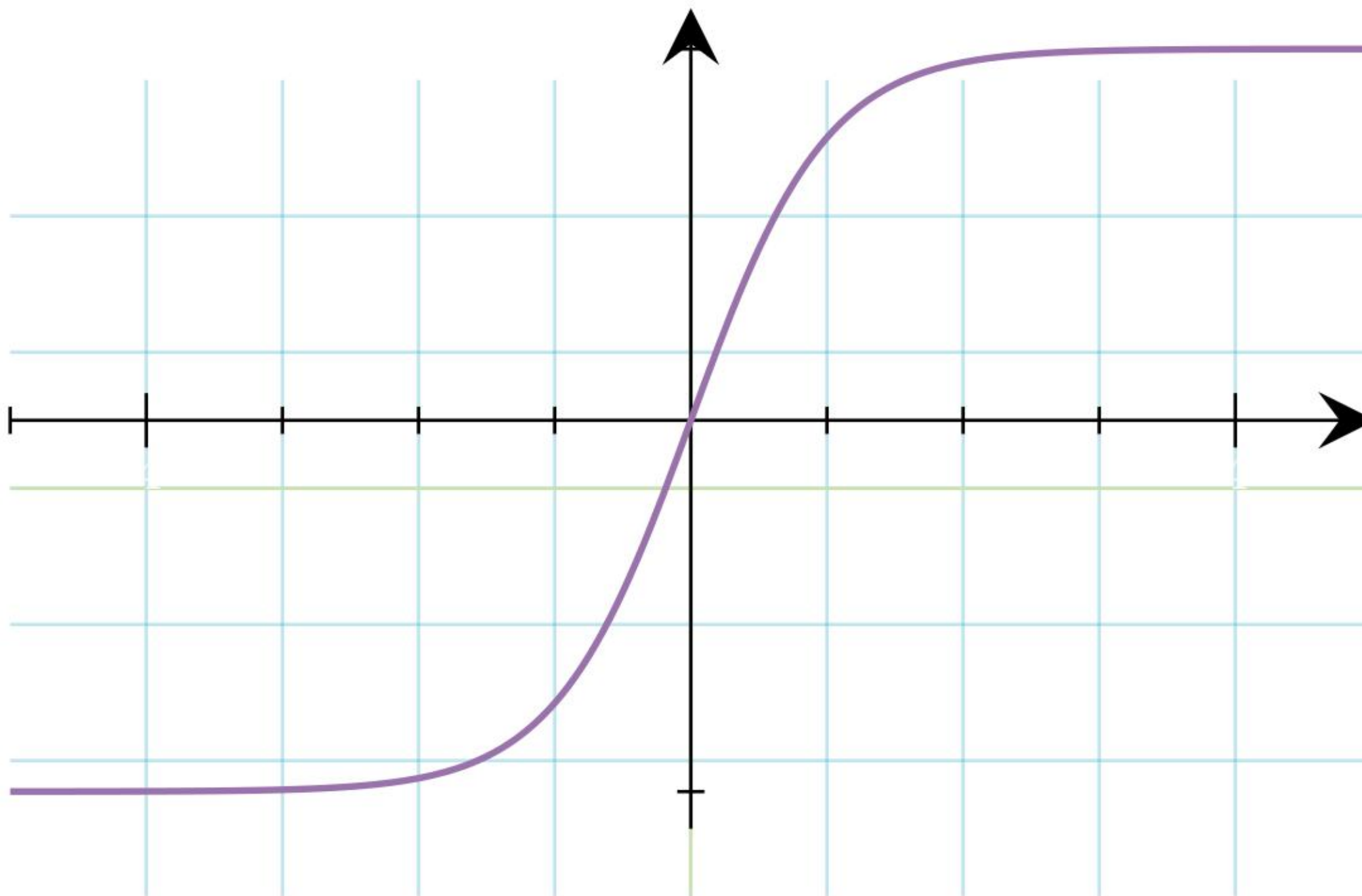
Tanh激活函数

1. 形式

$$\sigma(z) = \frac{2}{1 + \exp(-2z)} - 1$$

- 值域: $(-1, 1)$
- 导数: $\sigma'(z) = 1 - \sigma^2(z)$

Tanh 激活函数



Tanh 激活函数

1. 优点

- 激活函数关于 0 对称
- 激活值均值为 0
- 相对于 sigmoid 激活函数，Tanh 激活函数在 0 附近的导数更大.

2. 缺点

- 梯度消失
- 由于指数导致计算效率低下
- 没有稀疏化效果

ReLU激活函数

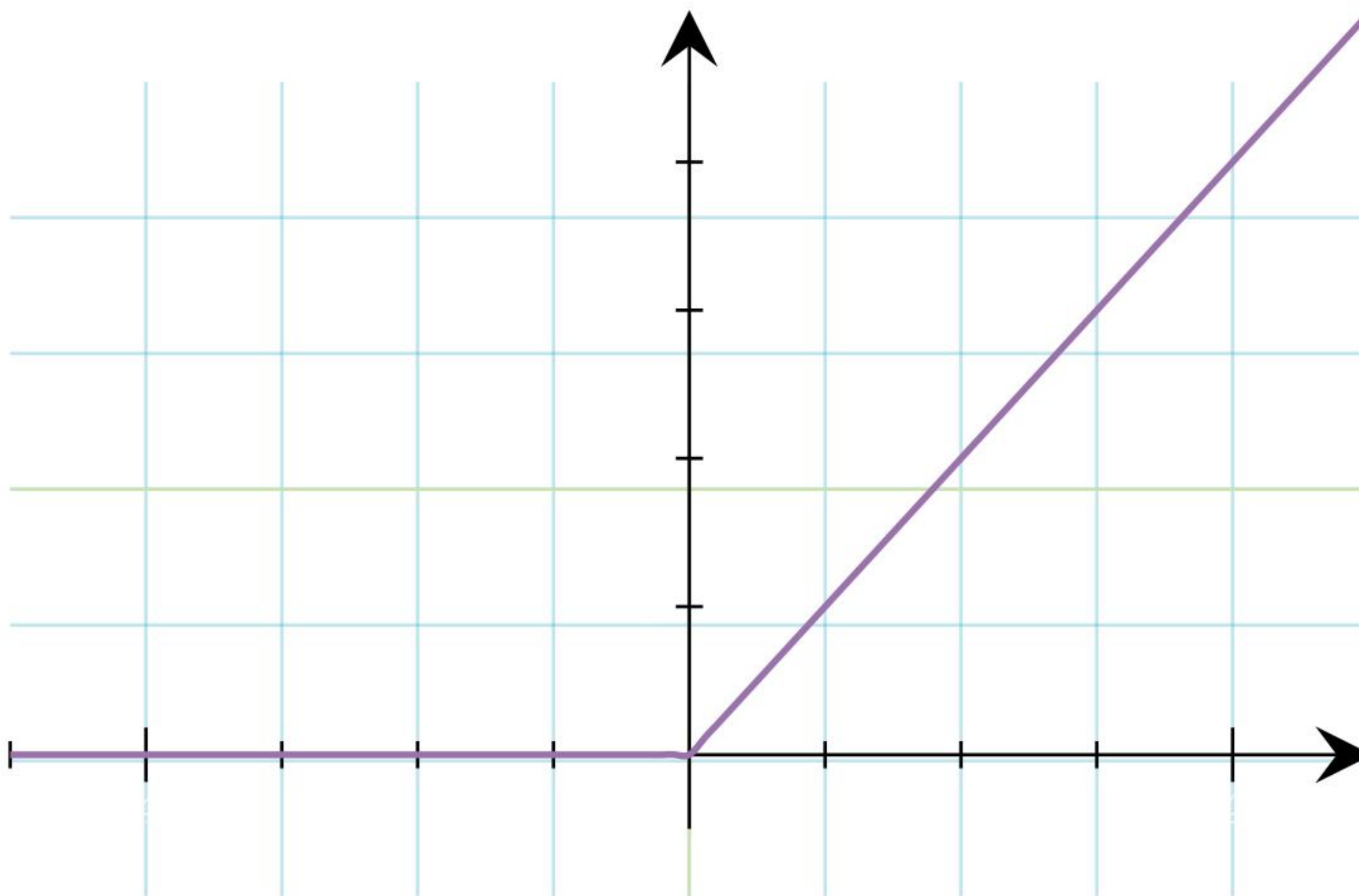
1. 形式 (Rectified Linear Unit)

$$\sigma(z) = \max\{0, z\}$$

- 值域: $[0, \infty)$
- 导数:

$$\sigma'(z) = \begin{cases} 0 & z \leq 0 \\ 1 & z > 0 \end{cases}$$

ReLU激活函数



ReLU激活函数

1. 优点

- 计算效率高
- 减轻了 sigmoid 激活函数等的梯度消失的问题
- 稀疏激活
- 较快的收敛速度

2. 缺点

- Dying ReLU.
- 输出不关于 0 对称，可能导致“协变量”偏移
- 在 0 处不可导

Leaky ReLU激活函数

1. 形式 (Leaky Rectified Linear Unit)

$$\sigma(z) = \begin{cases} \alpha z & z \leq 0 \\ z & z > 0 \end{cases}$$

- α : 较小的正常数。例如, $\alpha = 0.01$
- 值域: $(-\infty, \infty)$
- 导数:

$$\sigma'(z) = \begin{cases} \alpha & z \leq 0 \\ 1 & z > 0 \end{cases}$$

Leaky ReLU激活函数

