



神经网络中的激活函数是一个非线性函数，用于在神经网络中引入非线性。常用的激活函数包括：

1. Sigmoid函数: $f(x) = \frac{1}{1+e^{-x}}$

优点：在输入值较小时，具有较大的导数，容易进行梯度下降；能够将输出映射到0到1之间。

缺点：容易出现梯度消失问题；不是0中心化的，因此可能会导致梯度方向偏向某个方向。

2. Tanh函数: $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

优点：在输入值较小时，具有较大的导数，容易进行梯度下降；将输出映射到-1到1之间，比sigmoid函数具有更广的范围。

缺点：也容易出现梯度消失问题；同样不是0中心化的，可能会导致梯度方向偏向某个方向。

3. ReLU函数: $f(x) = \max(0, x)$

优点：计算速度快；在输入值大于0时，具有较大的导数，能够有效避免梯度消失问题；可以将输出稀疏化，减少神经元之间的冗余计算。

缺点：在输入值小于0时，导数为0，容易出现“死神经元”，无法更新梯度。

4. Leaky ReLU函数: $f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \alpha x, & \text{otherwise} \end{cases}$

优点：与ReLU函数类似，但在输入值小于0时，导数不为0，可以避免出现“死神经元”。

缺点：在实践中，参数 α 需要手动调整，较难确定其最佳值。

5. ELU函数: $f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \alpha(e^x - 1), & \text{otherwise} \end{cases}$

优点：具有ReLU函数的优点，并且在输入值小于0时，导数不为0，可以避免出现“死神经元”。

缺点：计算比ReLU函数略慢；同样需要手动调整参数 α 。

6. Softmax函数: $f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^k e^{x_j}}$

优点：将输入值转换为概率分布，非常适合多分类任务。

缺点：容易受到数值不稳定性的影响，需要进行数



Send a message.

