



缺点：容易受到数值不稳定性的影响，需要进行数



如何理解梯度下降



梯度下降是一种用于优化函数的迭代算法，目的是找到一个局部最小值或全局最小值。

在机器学习中，我们通常需要最小化一个损失函数，该损失函数衡量了模型预测值与真实值之间的差异。因此，我们需要找到使损失函数最小化的模型参数。

梯度下降算法通过不断迭代模型参数，并计算损失函数相对于这些参数的梯度，来更新模型参数的值。梯度是损失函数在某一点处的斜率，表示损失函数在该点处的变化率和方向。因此，梯度的方向指向了损失函数变化最快的方向，梯度的反方向指向了损失函数变化最慢的方向。

梯度下降算法的基本思想是，从任意初始点开始，沿着梯度的反方向移动一定的步长，直到达到局部最小值或全局最小值。

具体来说，梯度下降算法的迭代公式为：

$$\theta = \theta - \alpha \nabla J(\theta)$$

其中， θ 表示模型参数， $J(\theta)$ 表示损失函数， $\nabla J(\theta)$ 表示损失函数在当前参数点处的梯度， α 表示学习率，表示每次迭代的步长大小。

在每次迭代中，梯度下降算法会根据当前参数点处的梯度和学习率来更新模型参数的值，使得损失函数不断减小，直到达到局部最小值或全局最小值。学习率的选择很重要，如果学习率过大，每次更新的步长可能会过大，导致无法收敛或者收敛速度慢；如果学习率过小，则每次更新的步长太小，可能会导致收敛速度太慢。

梯度下降算法是一种基础且常用的优化算法，被广泛应用于各种机器学习任务中。



Send a message.

