现在通过一个简单的算法来聊一下模型的具体训练流程

从简单的线性回归算法开始聊

简单的线性回归算法公式：



还可以写成这样：

备注：



损失函数、目标函数：



现在要不断的更改，参数的值，让的值最小，如果的值小于用户设置的阈值，那么代表模型已经训练出来（此时，就是最佳参数）

训练模型的流程大概已经推理出来了：

1. 随机产生，参数值
2. 带入到损失函数中，求得 误差值
3. 判断是否小于用户指定的阈值
4. 若是大于，则调整，参数，继续循环（2），（3），（4）步骤
5. 若小于，代表模型训练完成

总结：不断的调整参数，找到损失函数的最小值，小于阈值就行

问题：

1. 这不是没有用到简单的线性回归公式？ 藏在损失函数中呢
2. 关于第（4）步中的，调整，参数，电脑随机调整？不管三七二十一的去调整？

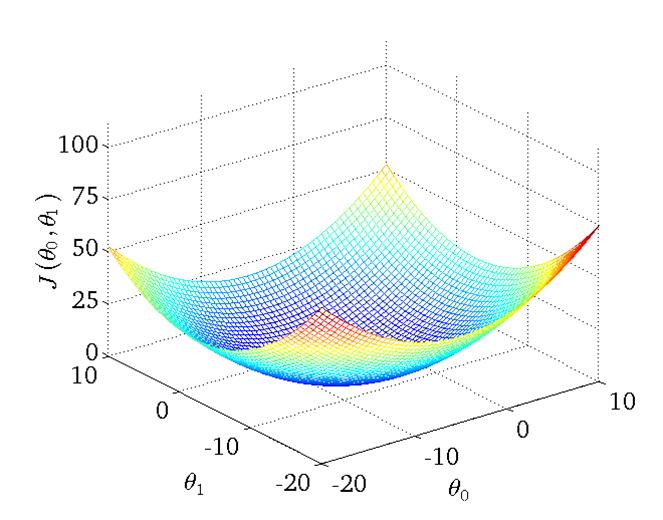
若这样调整，那训练一个模型的时间会很长，并且效率很低，那应该怎么调整呢？

假设：，与关系呈凹函数形状的样子







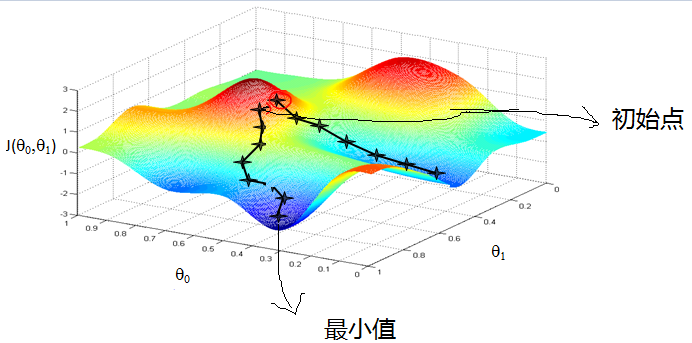


怎么调整参数? 肯定是分别沿着梯度的反方向调整

这种思想就是梯度下降算法的思想

整体梳理一下梯度下降法的思想：

将损失函数看成是一座山



怎么才能以最快的速度到达山底呢？

在山上环顾四周，找到坡最抖的路，走几步，然后再环顾四周，再次寻找最抖的路，以此下山，这样的速度是最快的

这一切都是我们在假设成立的条件下的幻想，证明一下假设是否成立？

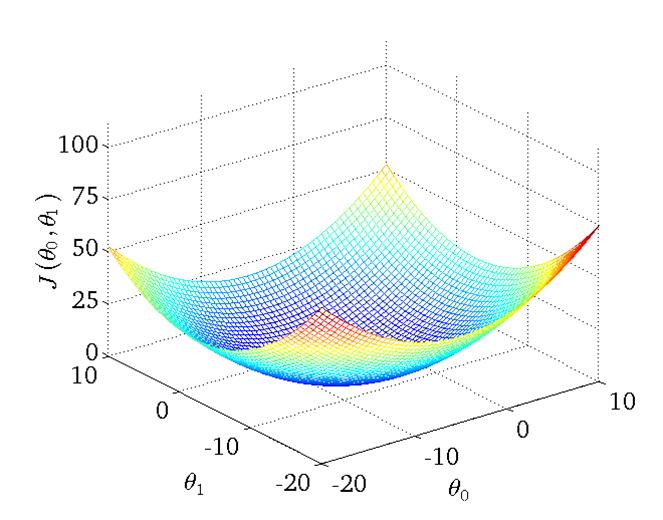


确定函数形状：

1. 找点描线
2. 二次求导
3. 经验推导

简化上面公式：





找梯度 说白了就是找斜率，找斜率说白了就是求各个维度上的偏导数

小案例操作一下Spark MLlib 什么是特征

手写训练线性回归算法模型的代码

偏导数：





偏导数：



1. 随机产生w0 w1参数
2. 带入到损失函数中，是否小于阈值
3. 使用梯度下降算法来调整w0参数



1. 使用梯度下降算法来调整w1参数



5、带入到误差函数中 继续求解

误差函数区别