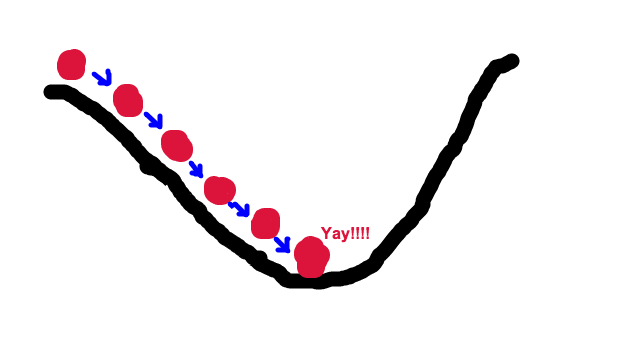


梯度下降 - 徒步下山问题

优达学城

你是否爬过山？我相信在某个位置你必须徒步下山？徒步下山是一种不错的练习，它能够帮助你理解梯度下降。

**当你徒步下山的时候，你的目标是什么？**- 享受快乐，并且到达山底。现在让我们专注于如何到达山底。



徒步下山的过程中，红色的点在做什么？它一直向下走，直至到达底部。让我们称我们的朋友为微积分，看看她对此有何看法。

**导数**

在我们深入之前，我会提醒你一些关于导数的事。可以从三个不同的方面观察导数，最常见的两个方面是：

* 函数曲线的切线斜率
* 函数变化的比率

以下是一些常见的导数：

• d(x2)/dx =2x

• d(−2y5)/dy = −10y4

• d(5−θ)2/dθ = −2(5 − θ)（负号来自−θ ）

听上去很不错！如果我们的函数中不止一个变量该如何？这样，我们将讨论偏导数！让我们来看一些例子：

•∂ (x2y2) /∂x = 2xy2

•∂ (−2y5 + z2)/ ∂y = −10y4

•∂ (5θ1 +2θ2 −12θ3)/ ∂θ2=2

•∂ (0.55−(5θ1 + 2θ2 − 12θ3)) /∂θ2= −2 (你能确定负号-来自哪里吗？)

现在我们已经熟悉了导数。让我们来看一下梯度下降算法。等等，我们还没到这个阶段！更多的微积分！

**什么是梯度？**

梯度是多个变量下的全局求导。设θ’s为以下函数J(θ1, θ2, θ3)的变量

J(Θ) = 0.55 − (5θ1 + 2θ2 − 12θ3)

∂J ∂J ∂J ∇J(Θ)= ∂θ ,∂θ ,∂θ 123 = ⟨−5, −2, 12⟩

这里，∇是一种象征性方式说明我们正在使用梯度函数，并且在< and >中的梯度代表梯度是一个向量。

让我们看一个稍微复杂点的例子。为了确保你真的理解，使用梯度下降来解线性回归问题时，我们将使用此种表达式。

J(Θ) = 1 (0.55 − (5θ1 + 2θ2 − 12θ3))2 2

∂J ∂J ∂J ∇J(Θ)= ∂θ ,∂θ ,∂θ

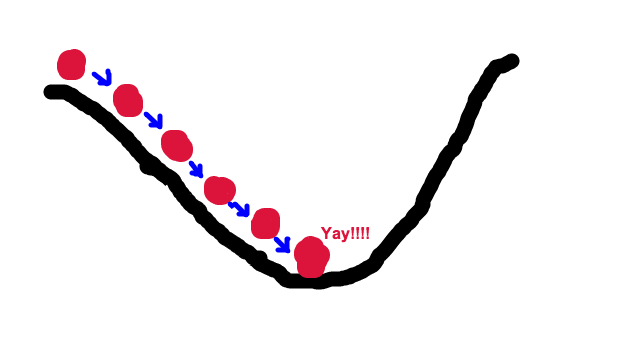
123 =⟨−5(0.55−(5θ1 +2θ2 −12θ3)),−2(0.55−(5θ1 +2θ2 −12θ3)),12(0.55−(5θ1 +2θ2 −12θ3))⟩

**为什么我们要关心梯度？**

梯度是微积分中十分强有力的工具。记住，在一个变量的情况下，梯度代表切线斜率。在多个变量情况下，梯度指向函数上升最快的方向。这在梯度下降算法中广泛应用。让我看看如何它是如何应用的。

**梯度下降算法的作用是什么？**

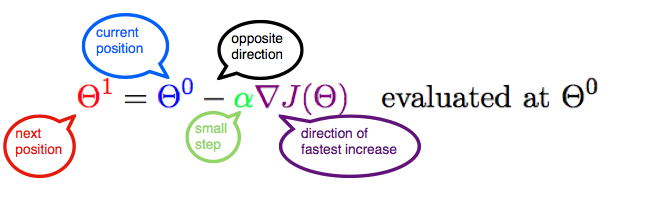
梯度下降算是一个迭代的过程，帮助我们得出函数的最小值（通常这个不会发生，有一些注意事项！）让我们再看一下红点的例子：



如果你希望到达底部，你会朝那个方向走？沿下坡方向，对吗？我们如何找到下坡方向？这和上升最快的方向相反。这代表，如果我们在Θ0点，想要移动到附近最低点（这就是为什么我们叫“局部最小值”），下一步应该是：

Θ1 = Θ0 − α∇J(Θ) 从Θ0计值

这需要进一步说明。



图片文字

（next position：下一个位置

Current position：当前位置

Opposite direction：反方向

Small step：最小步伐

Direction of fastest increase：上升最快的方向

Evaluated at：计值）

**α是怎么回事？**

α称为学习率或步伐大小。这意味着我们要慢慢来，不至于越过底部。当我们接近最小值的时候，这就变得十分重要。选取合理的α至关重要。如果α太小，算法永远无法到达最低点，如果α太大，我们可能会越过或者错失最底部。

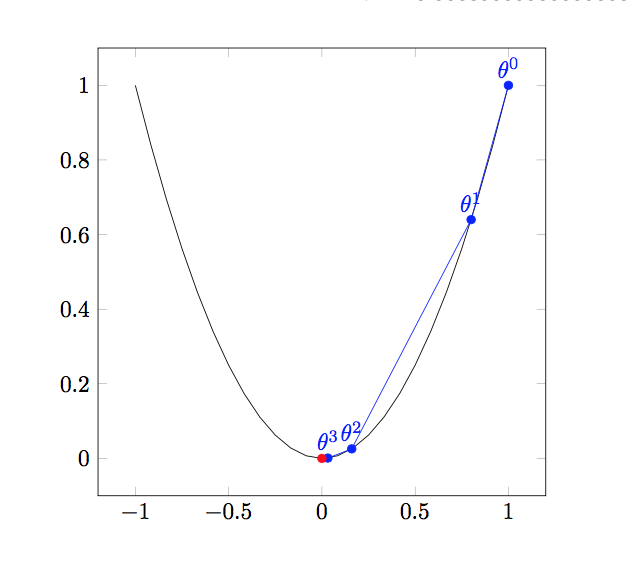
**为何取负-？**

− 负号代表我们的方向与∇J相反，例如，我们沿着与上升最快的反方向走。

**一个变量的示例**

让我们看看一个变量会是什么样。假设J(θ) = θ2，然后求出导数J′(θ) = 2θ。我们设置初始值θ0 = 1，α = 0.4，然后，

* θ0  = 1
* θ1  = θ0 − α ∗ J′(θ0)  = 1−0.4∗2  = 0.8
* θ2  = θ1 − α ∗ J′(θ1)  = 0.15999999999999992
* θ3  = 0.03199999999999997
* θ4  = 0.0063999999999999925

****

**两个变量的示例**

假设J(Θ) = θ12 + θ2。很简单，通过观察我们可以看到取(0, 0)时，J取最小值。让我们看看，如果这是我们的梯度下降方法所得出的。

选择Θ0 = (1, 3) ， α = 0.1。

∇J(Θ) = ⟨2θ1, 2θ2⟩

从 (1, 3)计值，梯度向量为< 2, 6 >

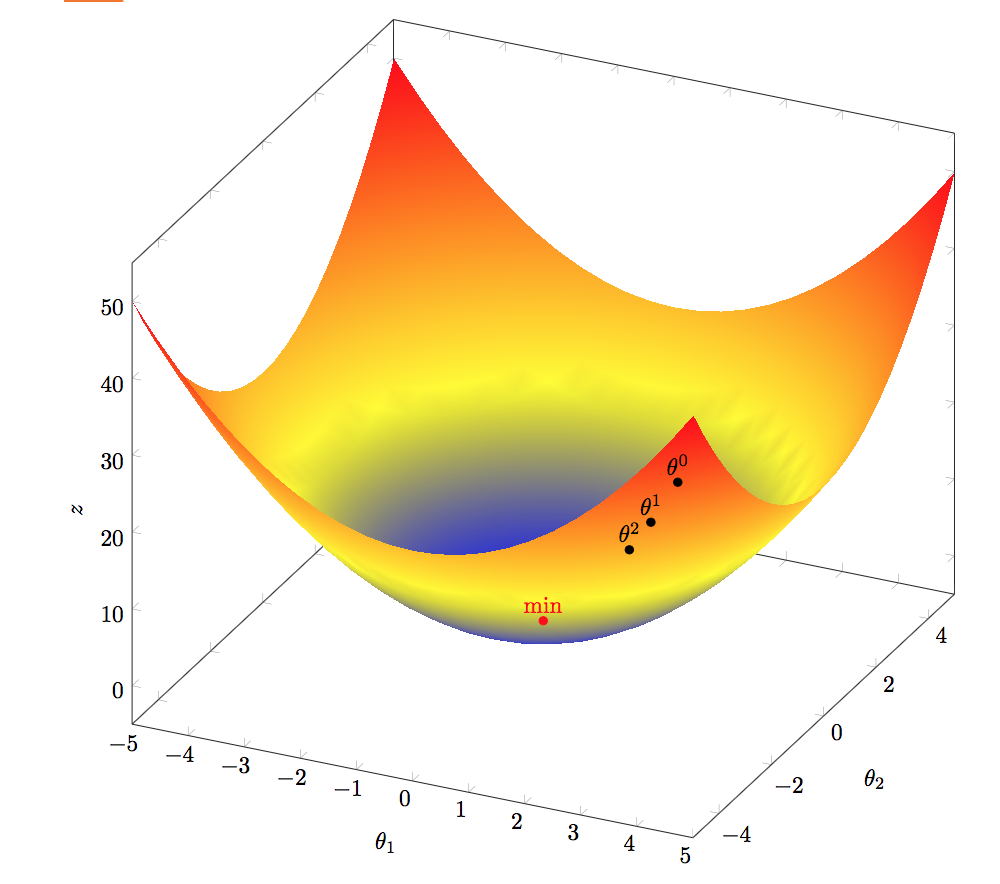
* Θ0  = (1, 3)
* Θ1  = Θ0 − α∇J(Θ)  = (1, 3) − 0.1(2, 6)  = (0.8, 2.4)
* Θ2  = (0.8, 2.4) − 0.1(1.6, 4.8)  = (0.64, 1.92)
* Θ3  = (0.512, 1.536)
* Θ4  = (0.4096, 1.2288000000000001)

.

Θ10 = (0.10737418240000003, 0.32212254720000005) .

Θ50 = (1.1417981541647683e−05, 3.425394462494306e−05) .

Θ100 = (1.6296287810675902e−10, 4.888886343202771e−10)



好极了！我们看到我们确实接近了我们所知道的最小值(0, 0)。

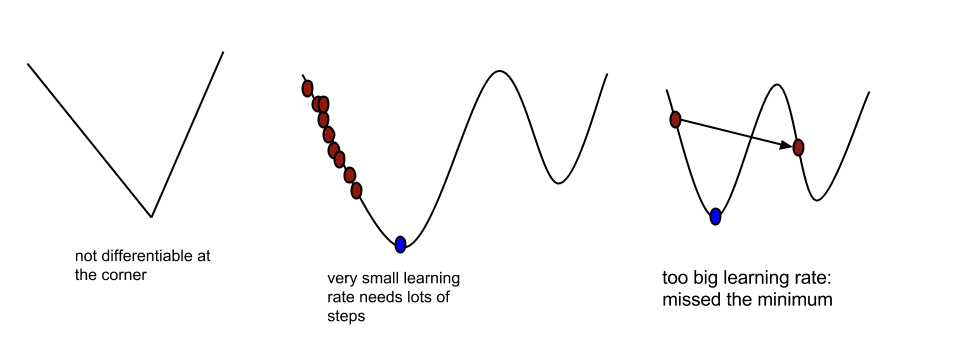
       

**注意事项**

以下几个最重要的事情要记住：

•函数必须是可微的。

•学习率不应太小或太大。



图片文字

（not differentiable at: 转角不可微

very small learning rate needs lots of steps: 很小的学习率需要很多步伐

too big learning rate missed the minimum: 太大的学习率会错失最小值）

**线性回归的应用**

梯度下降算法是一种用于线性回归以找到误差最小或者代价函数最小的方法。记住，你的误差或代价函数必须是可微的，才能使用梯度下降。这是取平方差的原因之一。使用绝对值计算误差会导致上图所示的“角”。