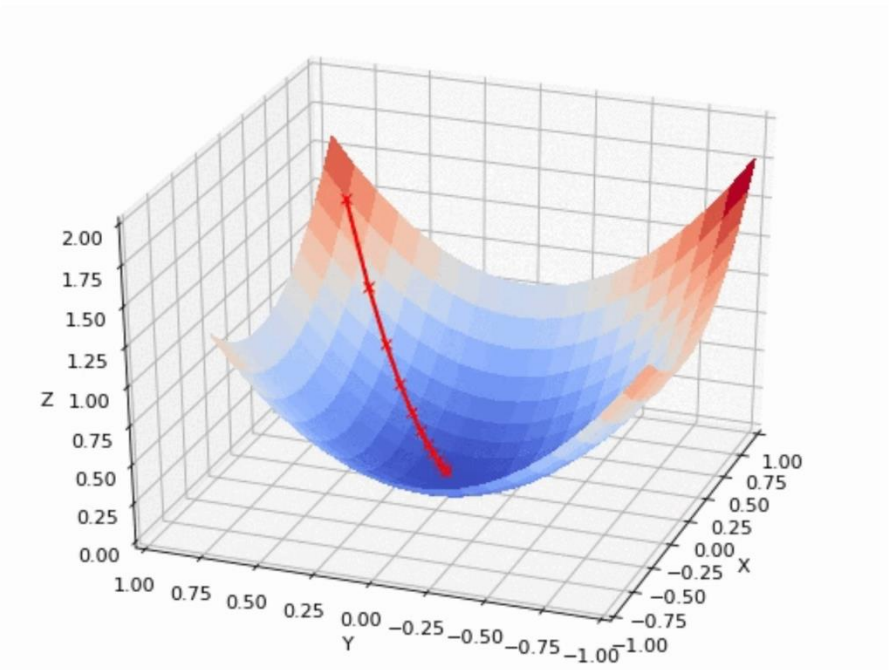
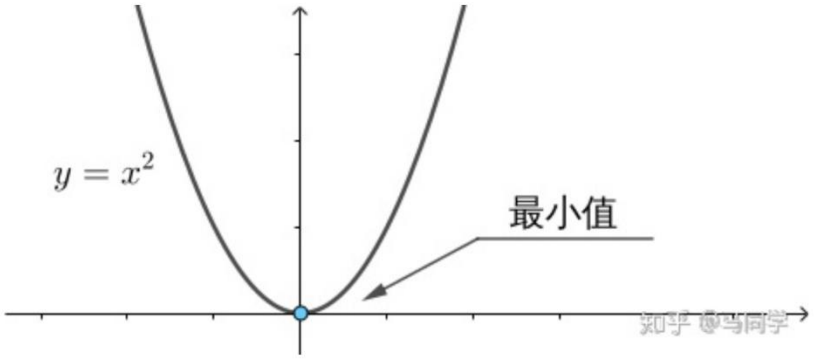
梯度下降法是用来计算函数最小值的。它的思路很简单，想象在山顶放了一个球，一松手它就会顺着山坡最陡峭的地方滚落到谷底：



凸函数图像看上去就像上面的山谷，如果运用梯度下降法的话，就可以通过一步步的滚动最终来到谷底，也就是找到了函数的最小值。

1 动机

先解释下为什么要有梯度下降法？其实最简单的二维凸函数是抛物线 $f(x) = x^2$ ，很容易通过解方程 $f'(x) = 0$ 求出最小值在 $x = 0$ 处：



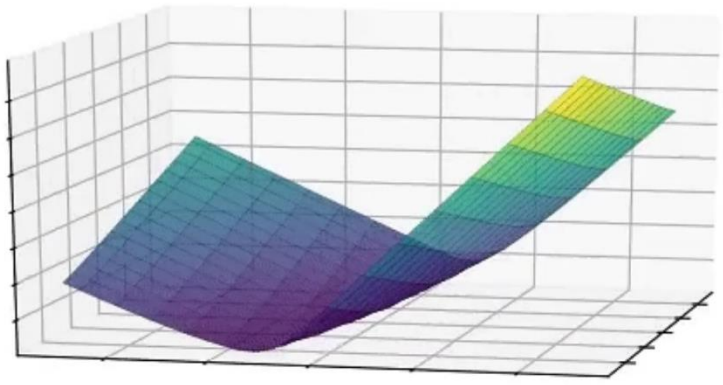
只是有一些凸函数，比如下面这个二元函数[Q]（该函数实际上是逻辑回归的经验误差函数，在监督式学习中确实需要求它的最小值）：

$$
\begin{aligned}
f(w_0, w_1) = \frac{1}{6}\Big[&\ln\left(1+e^{w_0+2w_1}\right)+\ln\left(1+e^{-w_0-7w_1}\right)\\
&+\ln\left(1+e^{-w_0-4w_1}\right)+\ln\left(1+e^{w_0+w_1}\right)\\
&+\ln\left(1+e^{-w_0-5w_1}\right)+\ln\left(1+e^{w_0+4.5w_1}\right)\Big]
\end{aligned}
$$

要求它的最小值点就需要解如下方程组：

$$
\begin{cases}
\dfrac{\partial f}{\partial w_0} = \dfrac{1}{6}\Big[\dfrac{e^{w_0+2w_1}}{1+e^{w_0+2w_1}} - \dfrac{e^{-w_0-7w_1}}{1+e^{-w_0-7w_1}} \\
\qquad\qquad - \dfrac{e^{-w_0-4w_1}}{1+e^{-w_0-4w_1}} + \dfrac{e^{w_0+w_1}}{1+e^{w_0+w_1}} \\
\qquad\qquad - \dfrac{e^{-w_0-5w_1}}{1+e^{-w_0-5w_1}} + \dfrac{e^{w_0+4.5w_1}}{1+e^{w_0+4.5w_1}}\Big] = 0 \\
\dfrac{\partial f}{\partial w_1} = \dfrac{1}{6}\Big[\dfrac{2e^{w_0+2w_1}}{1+e^{w_0+2w_1}} - \dfrac{7e^{-w_0-7w_1}}{1+e^{-w_0-7w_1}} \\
\qquad\qquad - \dfrac{4e^{-w_0-4w_1}}{1+e^{-w_0-4w_1}} + \dfrac{e^{w_0+w_1}}{1+e^{w_0+w_1}} \\
\qquad\qquad - \dfrac{5e^{-w_0-5w_1}}{1+e^{-w_0-5w_1}} + \dfrac{4.5e^{w_0+4.5w_1}}{1+e^{w_0+4.5w_1}}\Big] = 0
\end{cases}
$$

这个方程组实在太复杂了，直接求解难度太高，好在 $f(w_0, w_1)$ 的图像就像一座山谷：



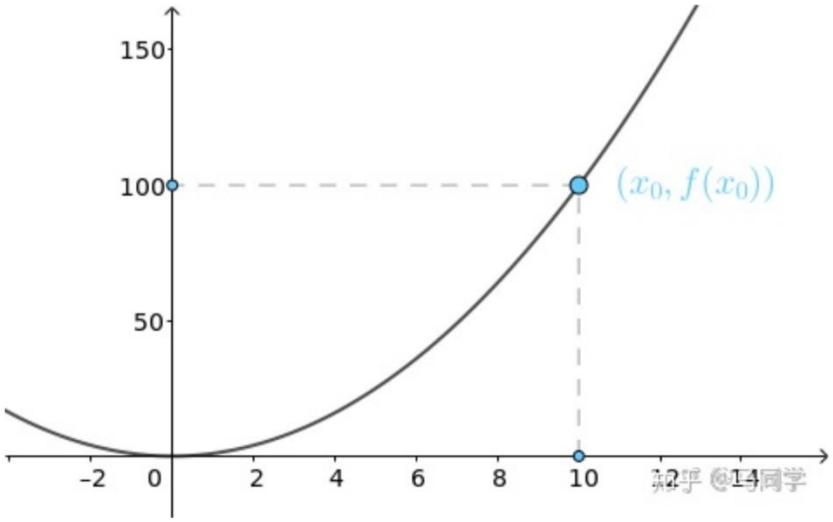
所以可以用梯度下降法来找到 $f(w_0, w_1)$ 的谷底，也就是最小值。

2 最简单的例子

梯度下降法在本文不打算进行严格地证明和讲解，主要通过一些例子来讲解，先从最简单的凸函数 $f(x) = x^2$ 开始讲起。

2.1 梯度向量

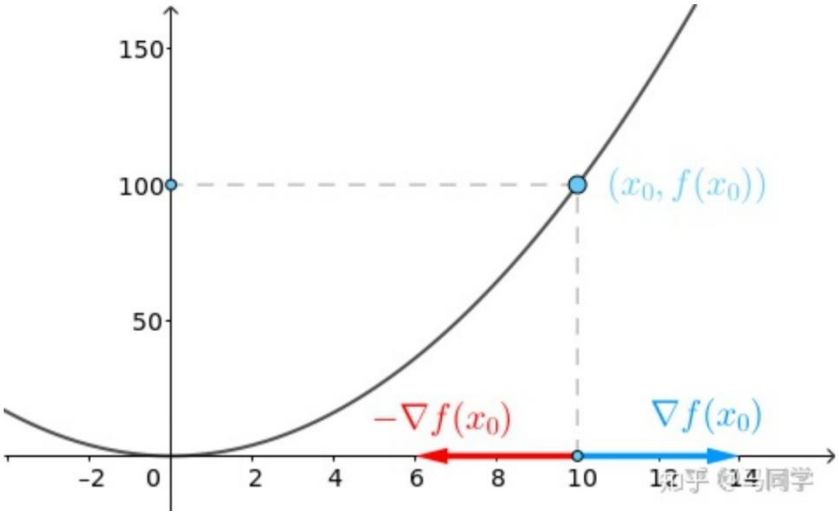
假设起点在 $x_0 = 10$ 处，也就是将球放在 $x_0 = 10$ ：



它的梯度为 1 维向量：

$$\nabla f(x_0) = f'(x_0)\boldsymbol{i} = \Big(f'(x_0)\Big) = \Big(2x|_{x_0=10}\Big) = (20)$$

这是在 x 轴上的向量，它指向函数值增长最快的方向，而 $-\nabla f(x_0)$ 就指向减少最快的方向：



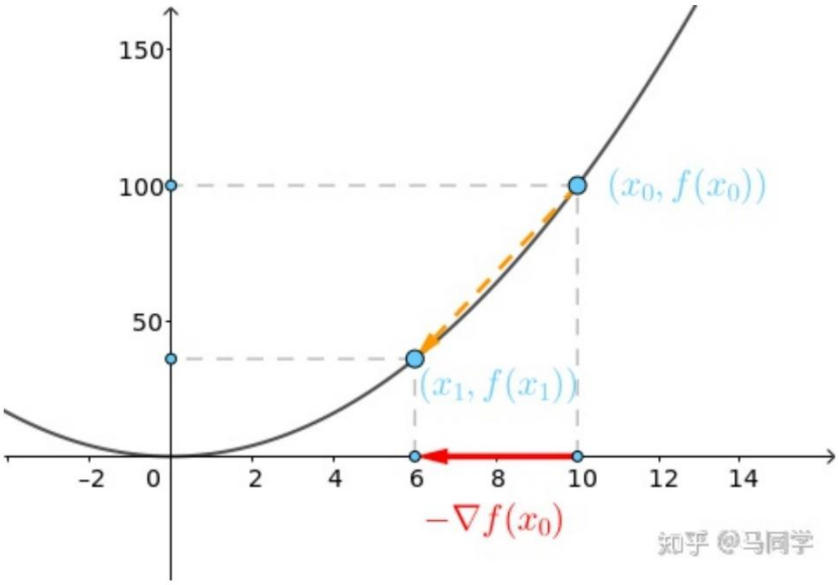
将 x_0 也看作 1 维向量 (x_0) ，通过和 $-\nabla f(x_0)$ 相加，可以将之向 $-\nabla f(x_0)$ 移动一段距离得到新的向量 (x_1) ：

$$(x_1) = (x_0) - \eta \nabla f(x_0)$$

其中 η 称为步长，通过它可以控制移的动距离，本节设 $\eta = 0.2$ ，那么：

$$(x_1) = (x_0) - \eta \nabla f(x_0) = (10) - 0.2 \times (20) = (6)$$

此时小球（也就是起点）下降到了 $x_1 = 6$ 这个位置：



2.2 迭代

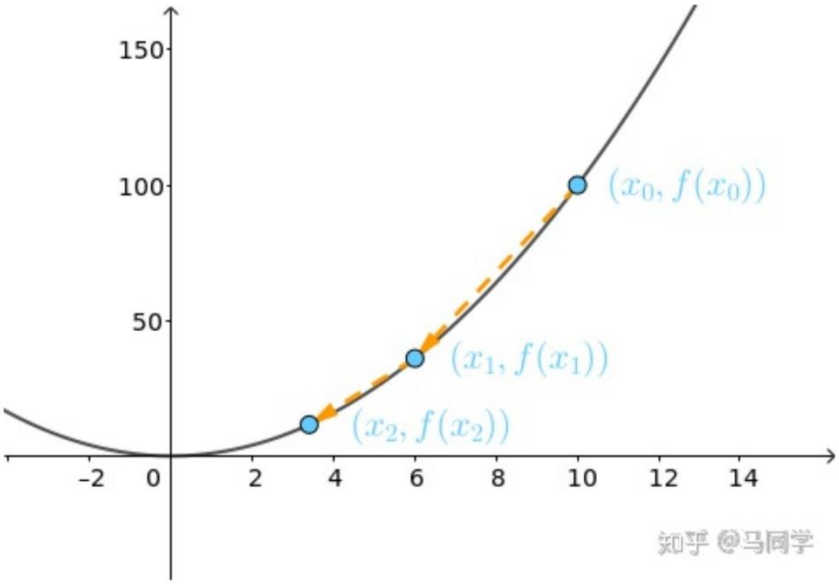
x_1 的梯度为：

$$\nabla f(x_1) = f'(x_1)\boldsymbol{i} = \Big(f'(x_1)\Big) = \Big(2x|_{x_1=6}\Big) = (12)$$

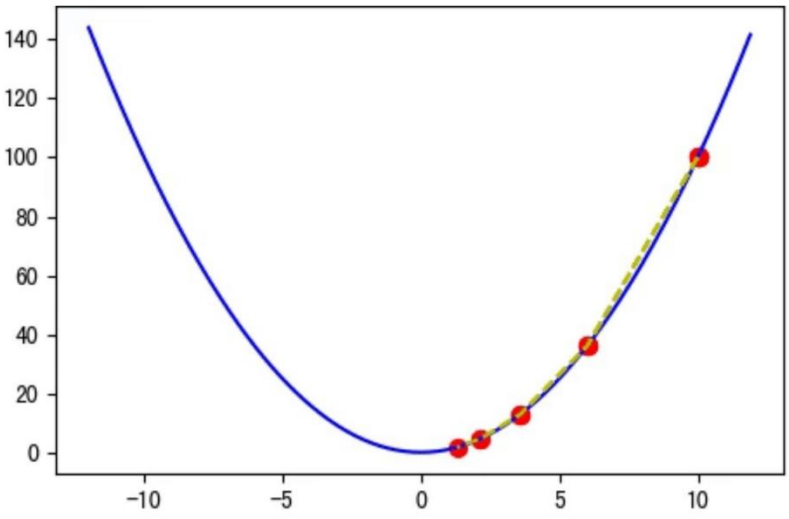
继续沿着梯度的反方向走：

$$(x_2) = (x_1) - \eta \nabla f(x_1) = (6) - 0.2 \times (12) = (3.6)$$

小球就滚到了更低的位置：



重复上述过程到第 10 次，小球基本上就到了最低点，即有 $x_{10} \approx 0$ ：



2.3 梯度下降法

把每一次的梯度向量 ∇f 的模长 $||\nabla f||$ 列出来，可以看到是在不断减小的，因此这种方法称为梯度下降法：

把每一次的梯度向量 ∇f 的模长 $||\nabla f||$ 列出来，可以看到是在不断减小的，因此这种方法称为梯度下降法：

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
$||\nabla f||$	20	12	7.2	4.32	2.59	1.56	0.93	0.56	0.34	0.2	0.12

这也比较好理解，当最终趋向于 0 时有：

$$||\nabla f|| = 0 \implies \nabla f = 0 \implies f'(x) = 0$$

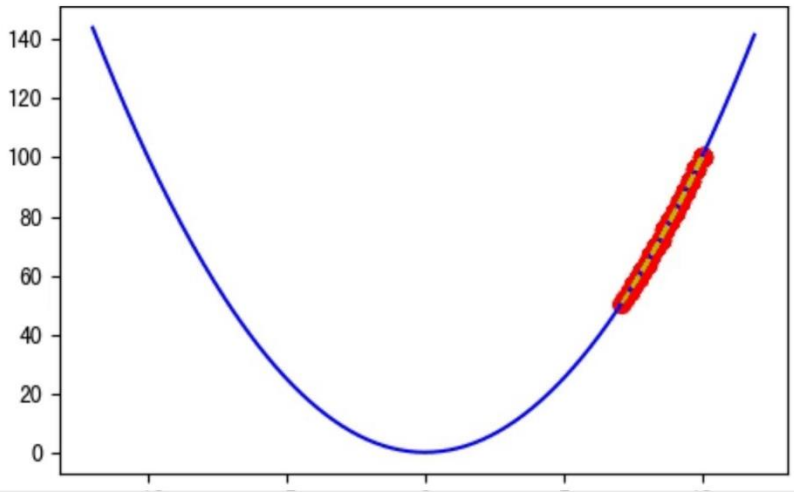
所以梯度下降法求出来的就是最小值（或者在附近）。

3 步长

上面谈到了可以通过步长 η 来控制每次移动的距离，下面来看看不同步长对最终结果的影响。

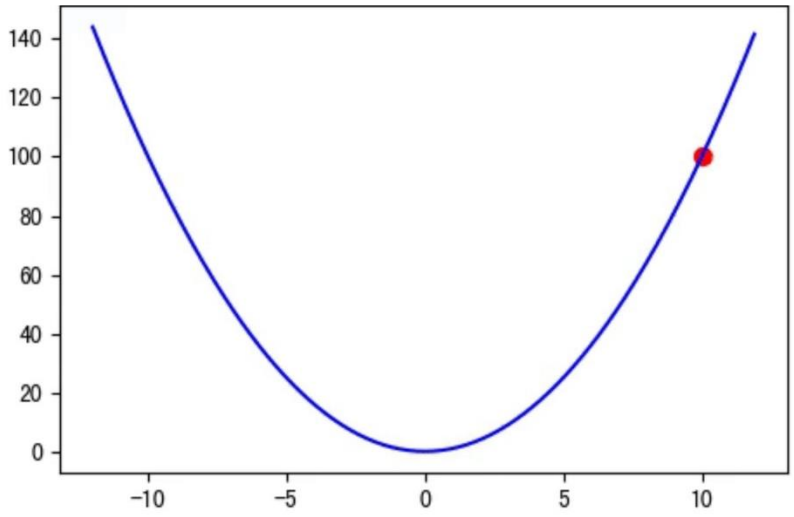
3.1 过小

如果设 $\eta = 0.01$ 就过于小了，迭代 20 次后离谷底还很远，实际上 100 次后都无法到达谷底：



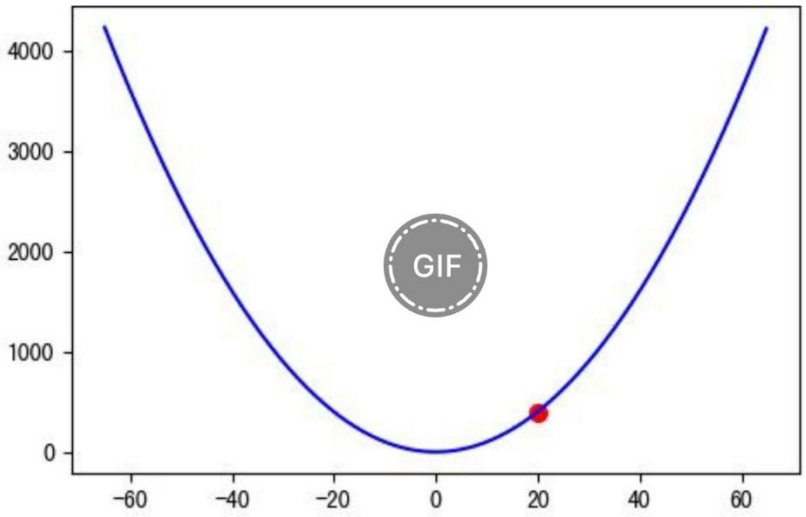
3.2 合适

上面例子中用的 $\eta = 0.2$ 是较为合适的步长，10 次就差不多找到了最小值：



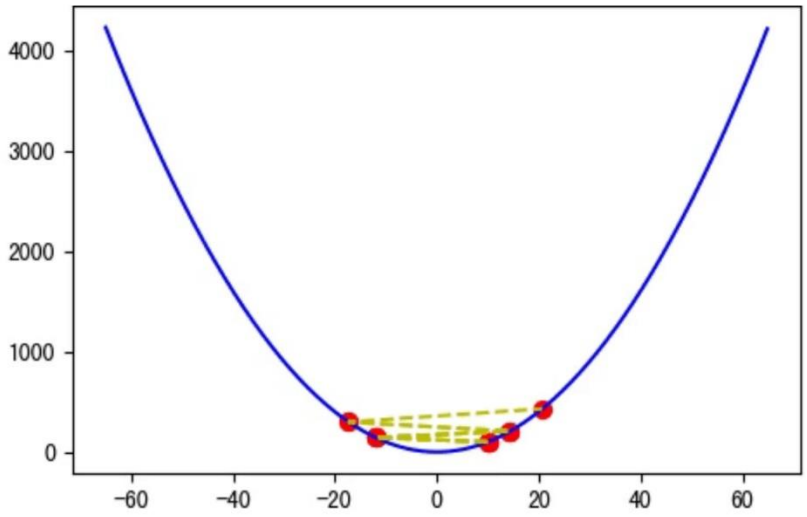
3.3 较大

如果令 $\eta = 1$ ，这个时候会来回震荡（下图看上去只有两个点，实际上在这两个点之间来来回回）：



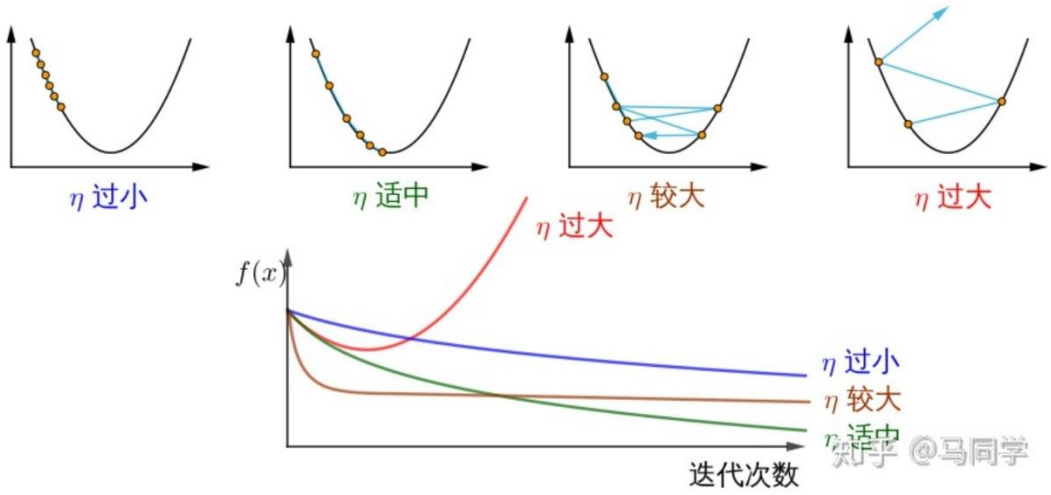
3.4 过大

继续加大步长，比如令 $\eta = 1.1$ ，反而会越过谷底，不断上升：



3.5 总结

总结下，不同的步长 η ，随着迭代次数的增加，会导致被优化函数 $f(x)$ 的值有不同的变化：



寻找合适的步长 η 是个手艺活，在工程中可以将上图画出来，根据图像来手动调整：

- $f(x)$ 往上走（红线），自然是 η 过大，需要调低
- $f(x)$ 一开始下降特别急，然后就几乎没有变化（棕线），可能是 η 较大，需要调

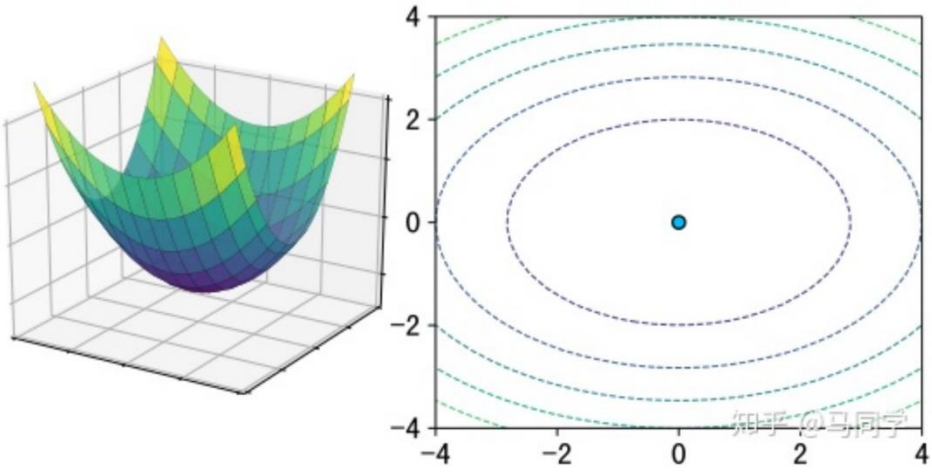
- $f(x)$ 几乎是线性变化（蓝线），可能是 η 过小，需要调高

4 三维的例子

原理都介绍完了，下面再通过一个三维的例子来加强对梯度下降法的理解。假设函数为：

$$f(\boldsymbol{x}) = x_1^2 + 2x_2^2$$

其图像及等高线如下（等高线中心的蓝点表示最小值）：



下面用梯度下降法来寻找最小值。

4.1 前进一步

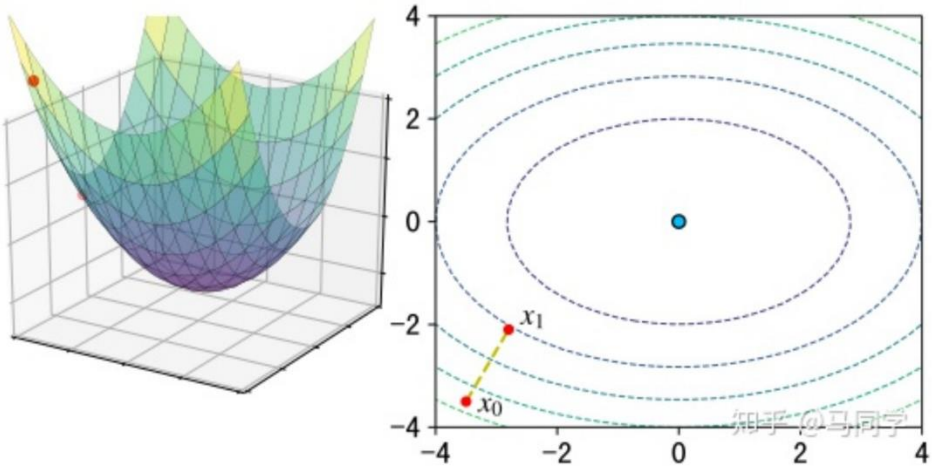
设初始点为 $\boldsymbol{x}_0 = (-3.5, -3.5)$，此时梯度为：

$$\nabla f(\boldsymbol{x}_0) = (\frac{\partial f(\boldsymbol{x}_0)}{\partial x_1}, \frac{\partial f(\boldsymbol{x}_0)}{\partial x_2}) = (2x_1, 4x_2)\Big|_{x_1=-3.5, x_2=-3.5} = (-7, -14)$$

令步长 $\eta = 0.1$，那么下一个点为：

$$\begin{aligned} \boldsymbol{x}_1 &= \boldsymbol{x}_0 - \eta \nabla f(\boldsymbol{x}_0) \\ &= (-3.5, -3.5) - 0.1 \times (-7, -14) = (-2.8, -2.1) \end{aligned}$$

可以看到向最小值方向前进了一步：

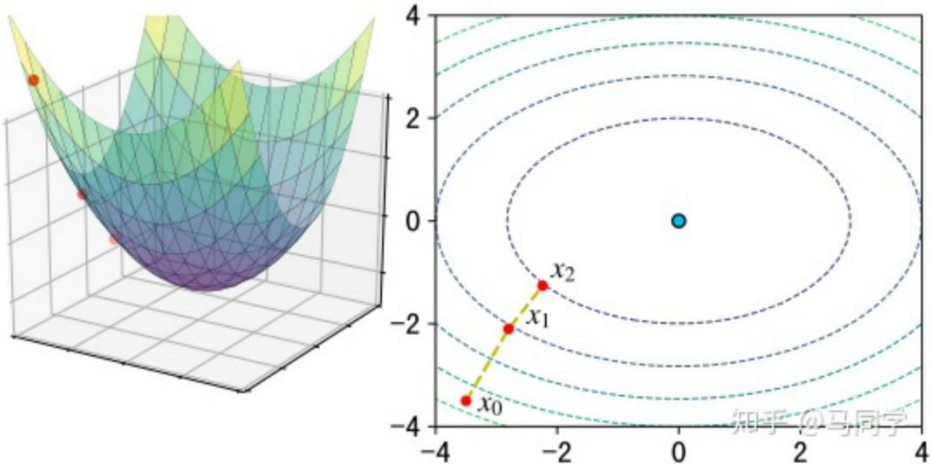


4.2 迭代

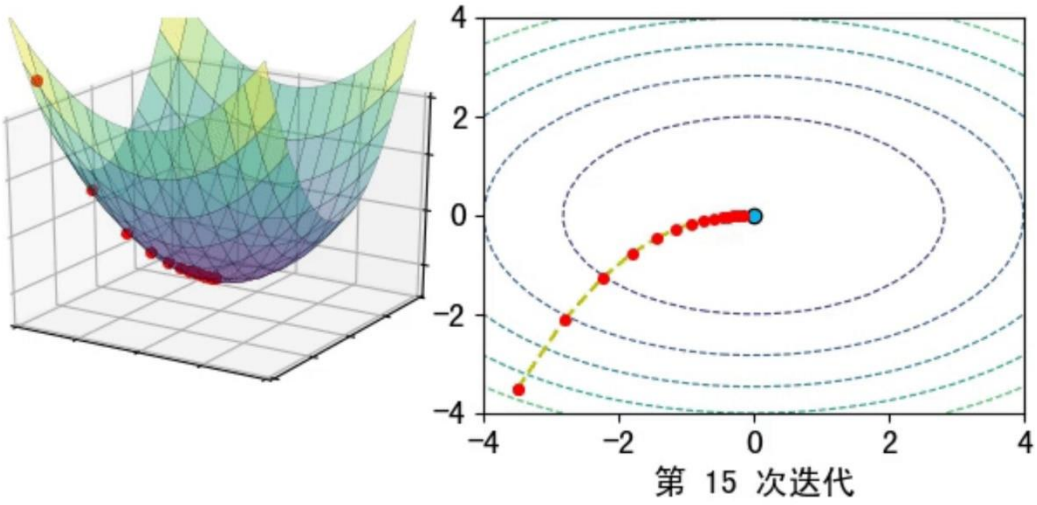
同样的方法找到下一个点：

$$\begin{aligned}\boldsymbol{x}_2 &= \boldsymbol{x}_1 - \eta\nabla f(\boldsymbol{x}_1)\\ &= (-2.8,-2.1) - 0.1\times(-5.6,-8.4) = (-2.24,-1.26)\end{aligned}$$

此时又向最小值靠近了：



如此迭代20次后，差不多找到了最小值：


4
2
0
−2
−4
−4
−2
0
2
4
第 15 次迭代