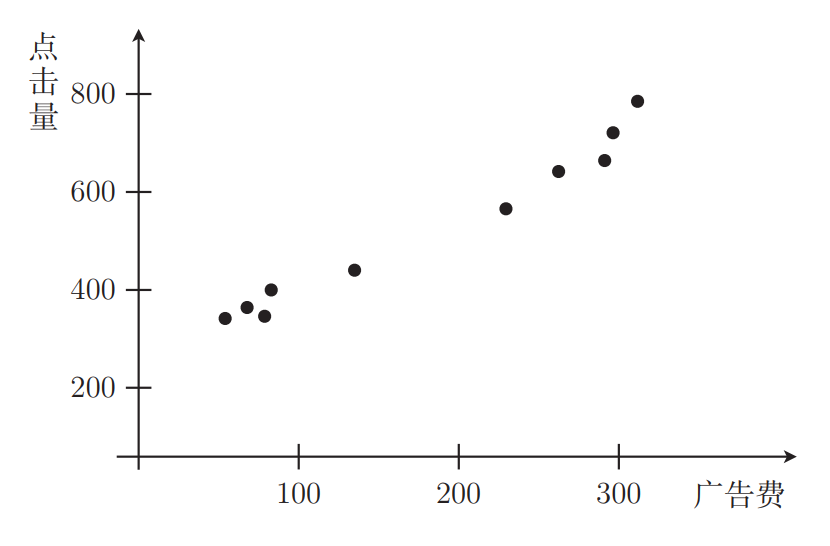
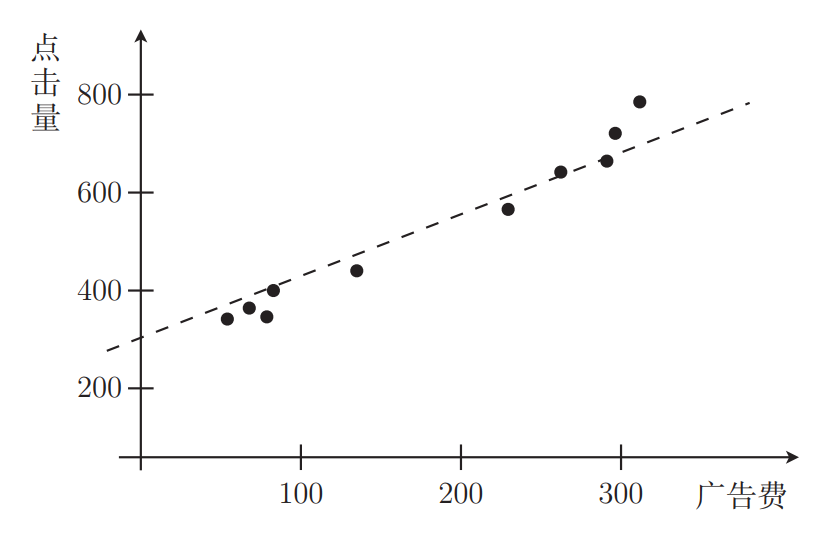
1. 设置问题

以 Web 广告和点击量的关系为例来学习回归，投入的广告费越多，广告的点击量就越高。﻿根据广告费和实际点击量的对应关系数据，可以将两个变量用下面的图展示出来。图中的值是随便选的。



1. 定义模型

﻿把图想象为函数。只要知道通过图中各点的函数的形式，就能根据广告费得知点击量了。﻿点击量经常变化，这叫作“点击量中含有噪声”，所以函数并不能完美地通过所有的点。



没错，这个模型就是个一次函数，﻿我们得像下面这样定义一次函数的表达式。

﻿我们就要使用机器学习来求出正确的和的值。

1. 最小二乘法

﻿图上画的那些点，﻿是我们手头上已有的广告费及其相应点击量的数据。﻿这些数据称为训练数据。虚线就是我们的模型。﻿我们将训练数据中的广告费代入，把得到的点击量与训练数据中的点击量相比较，然后找出使二者误差平方的总和最小的。

﻿我们来把刚才说的内容用表达式展现出来。假设有n个训练数据，那么它们的误差平方之和可以用这样的表达式表示。这个表达式称为目标函数。

解释：和﻿中的不是次幂的意思，而是指第个训练数据。﻿我们对每个训练数据的误差取平方之后，全部相加，然后乘以。这么做是为了找到使 的值最小的。这样的问题称为**最优化问题**。

﻿为什么要计算误差的平方呢？﻿如果只是简单地计算差值，我们就得考虑误差为负值的情况。负数误差和正数误差相加会使误差变的比实际小。﻿我们一般也不用绝对值，而用平方。因为之后要对目标函数进行微分，比起绝对值，平方的微分更加简单。

﻿那为什么整个表达式还要乘以呢？﻿这也和之后的微分有关系，是为了让作为结果的表达式变得简单而随便加的常数。

﻿我们实际来计算一下表达式的值吧。设= 1、= 2，然后我们用前4 个点作为训练数据代入表达式。求误差。结果有点大……

﻿= × ( +++ )

= × (66 049 + 59 536 + 44 944 + 53 824)

= 112 176.5

﻿112 176.5 这个值本身没有什么意义，我们要修改参数，使这个值变得越来越小。让这个值变小，也就是让误差变小。这种做法称为**最小二乘法**。

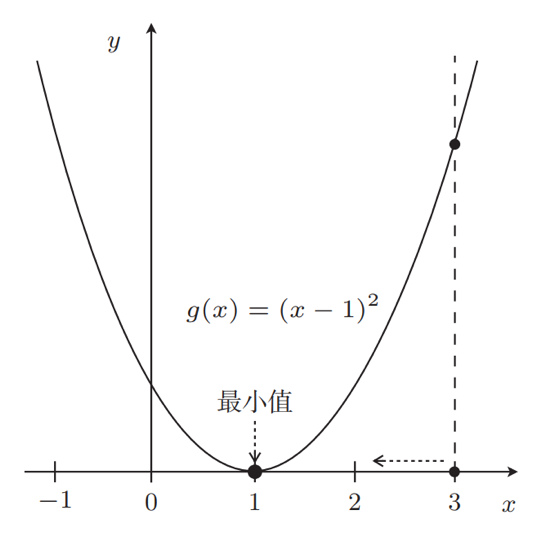
回顾下刚才的内容，我们的目的是找出使误差平方总和最小的。拥有和﻿两个参数。如何得到的值呢？我们先来看一个拥有一个参数的二次函数﻿：

它的微分是这样的：

他的增减表和导数的符号是这样的：

|  |  |  |
| --- | --- | --- |
| ﻿的范围 | 的符号 | 的增减 |
| ﻿﹤ 1 | - | ﻿↘︎ |
| ﻿ = 1 | 0 |  |
| ﻿ ﹥ 1 | + | ↗︎ |

﻿根据这张增减表我们可以知道，在 < 1 时，的图形向右下方延伸，反之当 > 1 时，的图形向右上方延伸，换句话说就是从左下方开始延伸的。



﻿比如在 = 3 这一点，为了使的值变小，我们需要向左移动，也就是必须减小如果是在另一侧的 = −1 这一点，为了使的值变小，我们需要向右移动，也就是必须增加。﻿只要向与导数的符号相反的方向移动，就会自然而然地沿着最小值的方向前进了。

﻿我们把刚才说的内容用表达式展示出来，就是这样的。这也被称为**最速下降法**或**梯度下降法**。

解释：﻿是称为学习率的正的常数(“调参侠”调的参数之一)，读作“伊塔”。根据学习率的大小，到达最小值的更新次数也会发生变化。换种说法就是收敛速度会不同。有时候甚至会出现完全无法收敛，一直发散的情况。

拥有和﻿两个参数，也就是说这个目标函数是拥有和﻿的双变量函数，所以不能用普通的微分，而要用偏微分。如此一来, 的更新表达式就是这样的。

此处省略求导过程，最终的更新表达式就是这样的。

﻿嗯，就是这样。只要根据这个表达式来更新和﻿，就可以找到正确的一次函数了。这就是机器学习训练的过程。