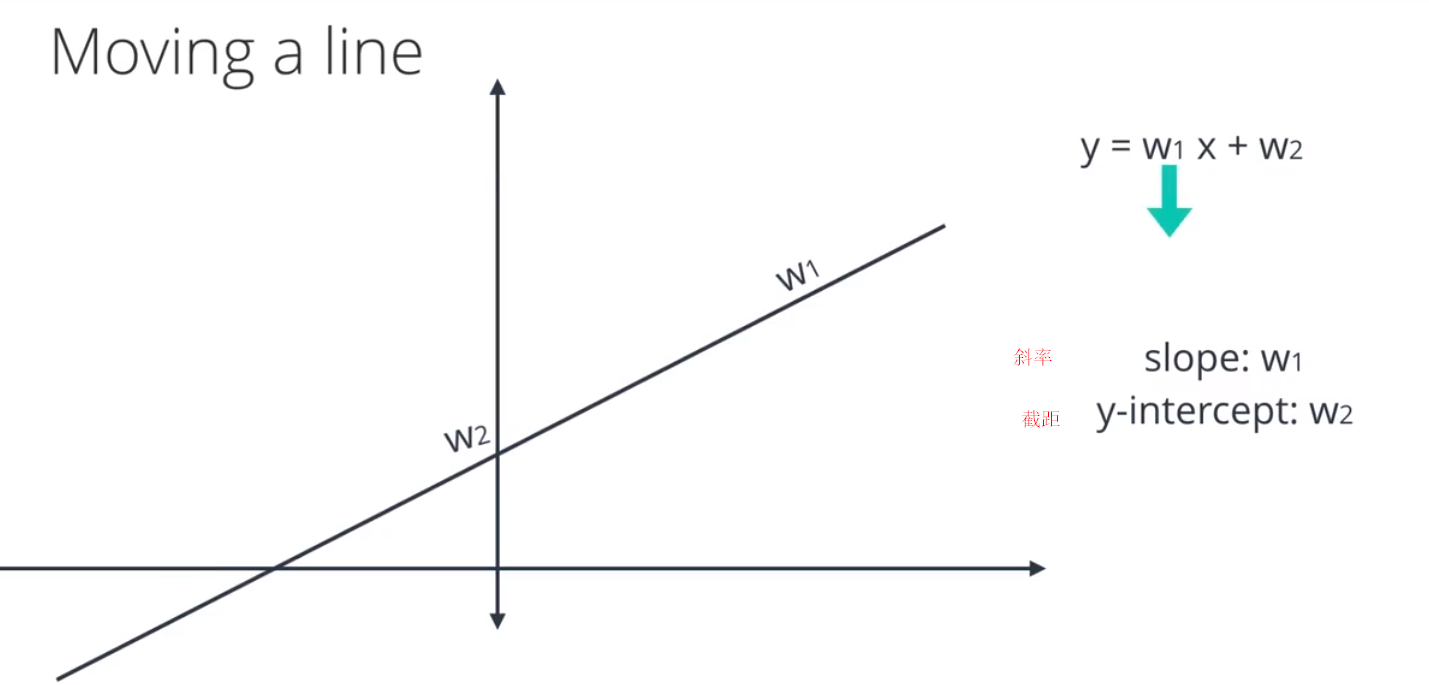
# 02．线性回归定义

线性回归是从数据中，找到一条拟合的线，这条线将举例所有点最近，然后进行预测.

回归模型算法

# 05．调整一条直线

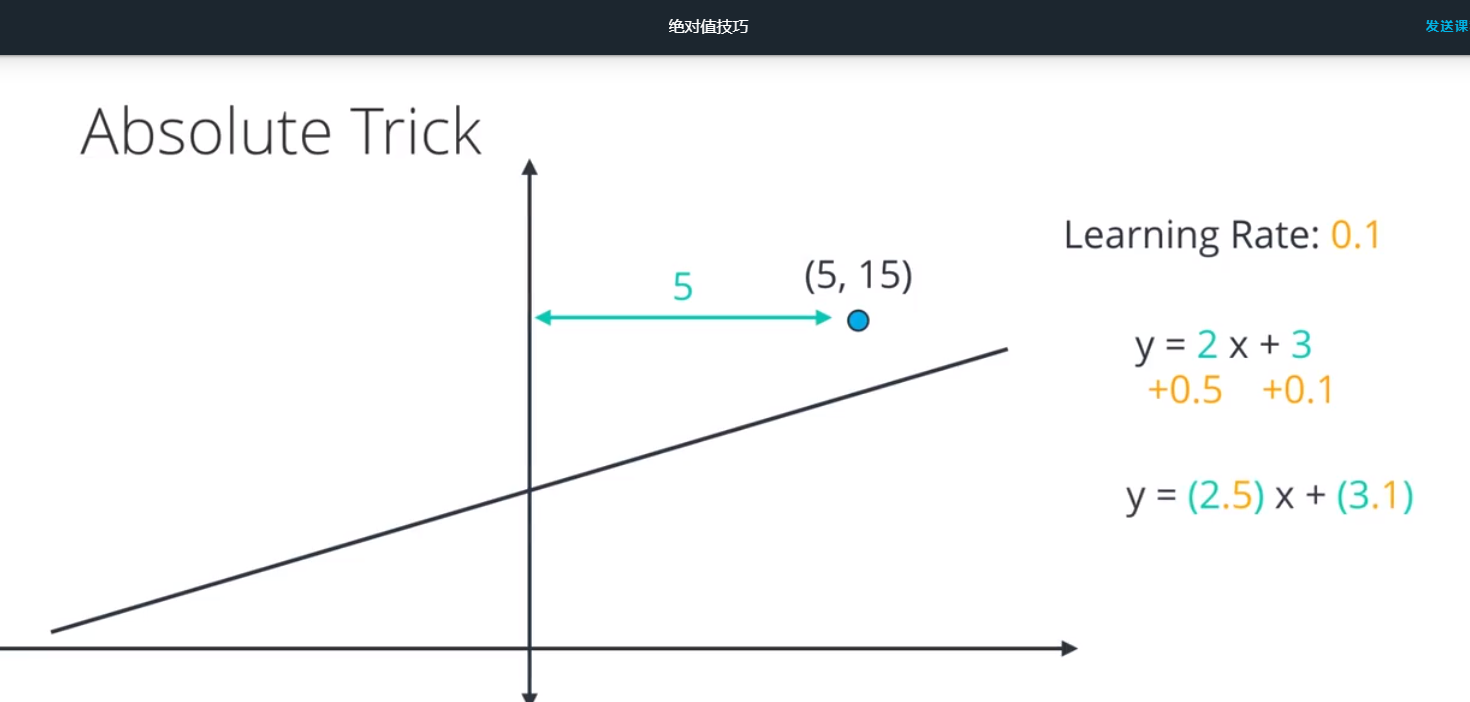
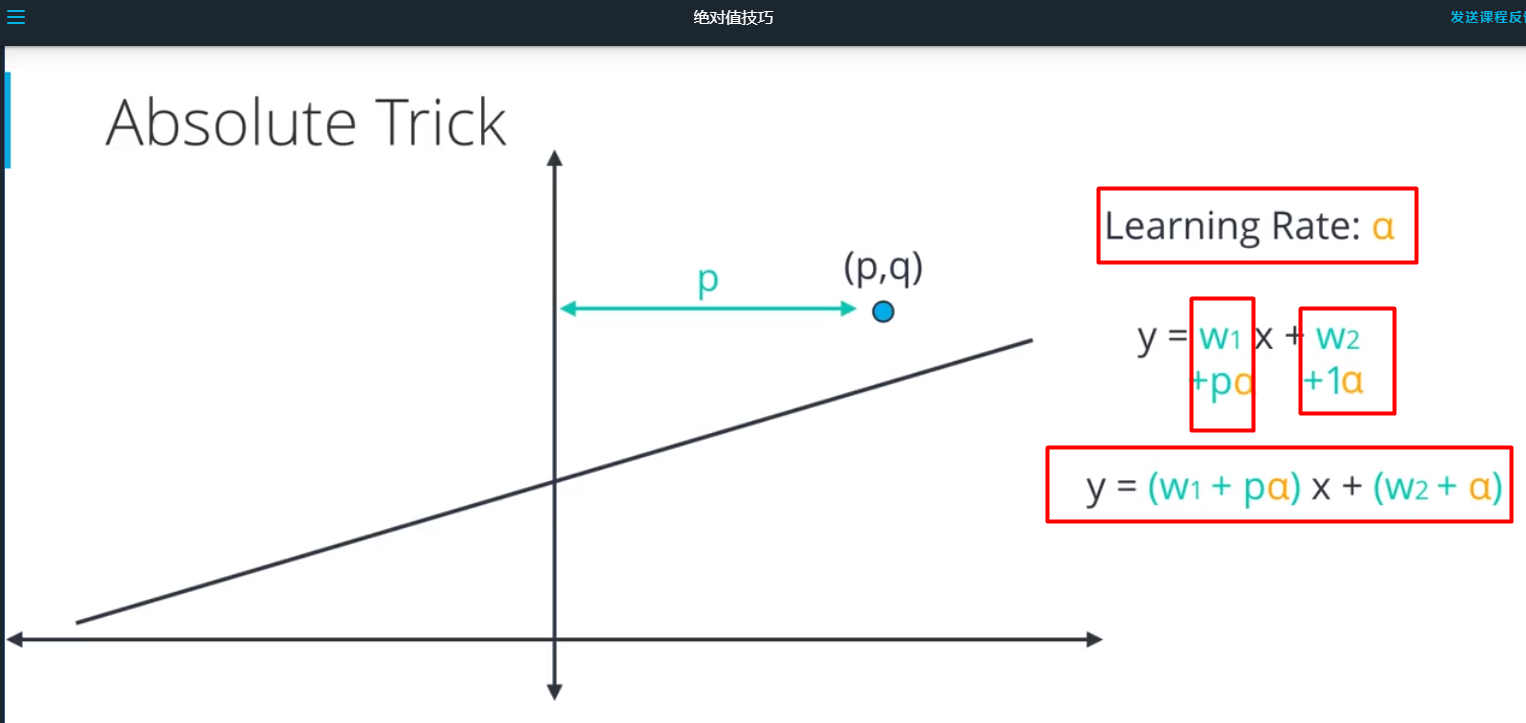


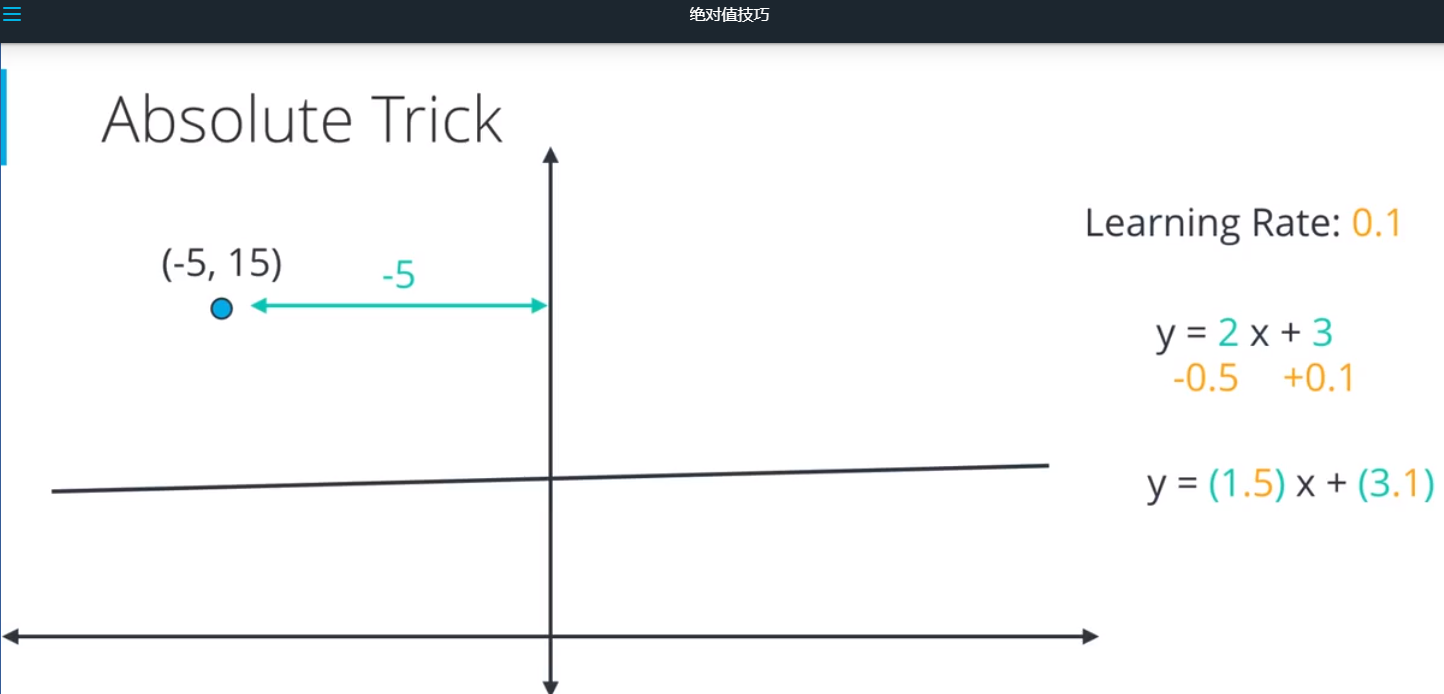
w2：调整截距，y进行平行移动

增加w1斜率升高，向逆时针方向变化，反正依然

# 06．绝对值技巧

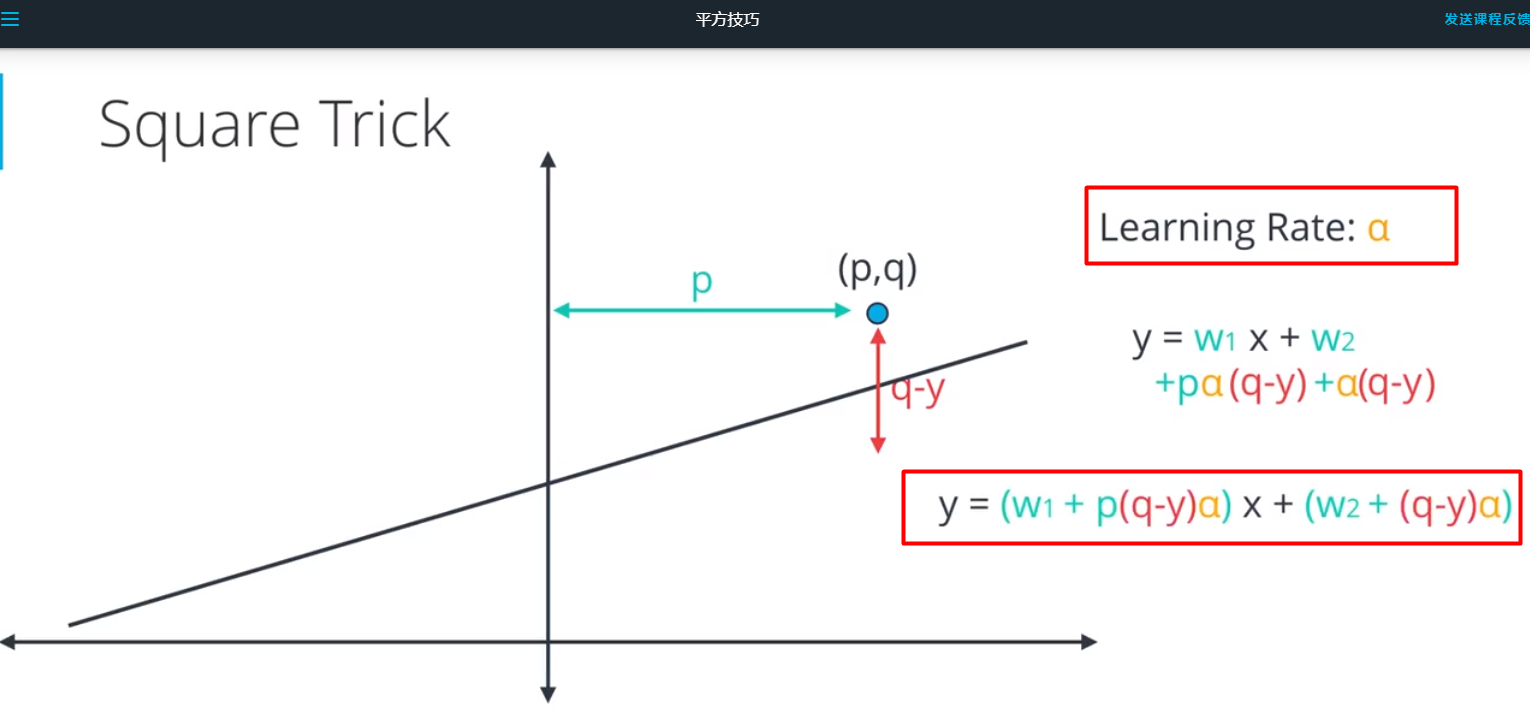
为了让直线靠近 某点，需要同时调整 w1斜率，w2截距。通过学习率，控制直线的调整幅度。

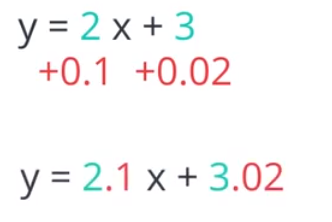
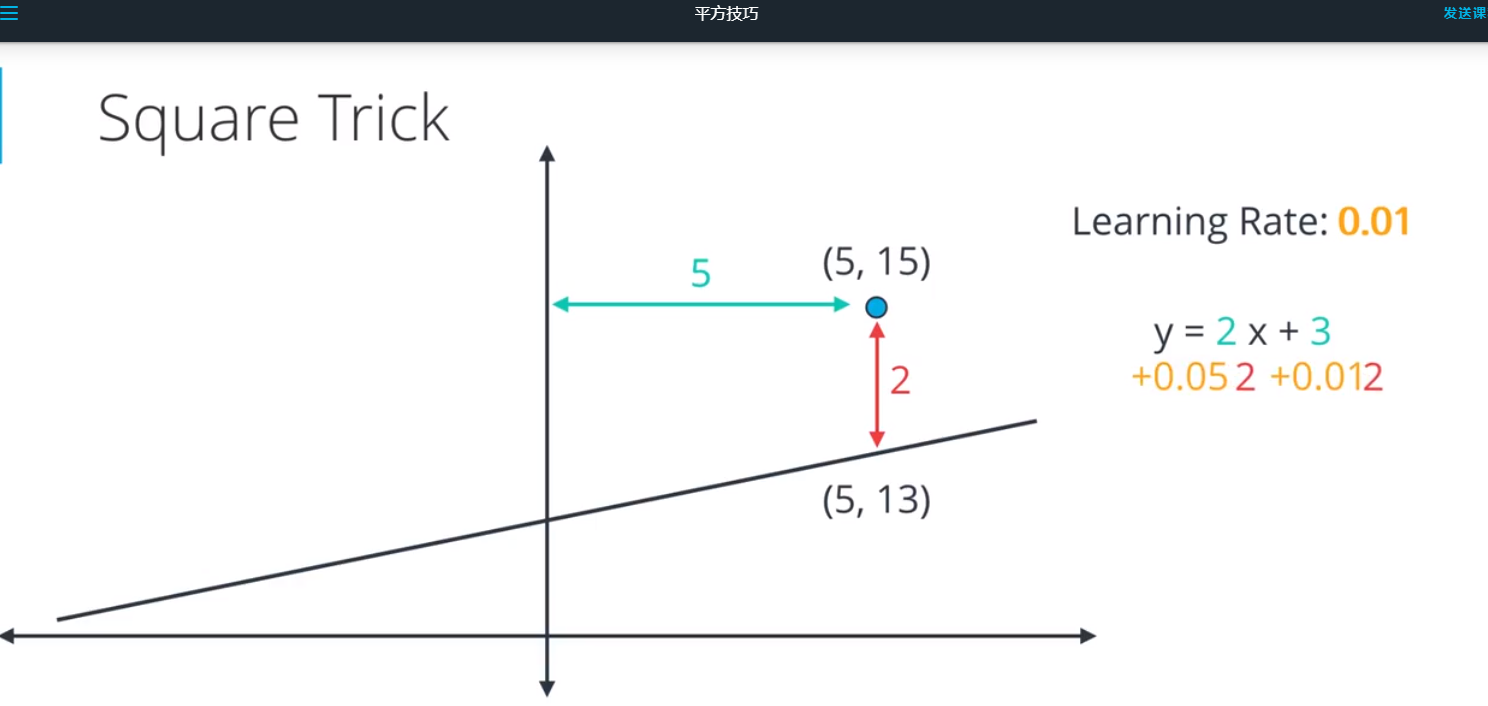




# 07．平方技巧

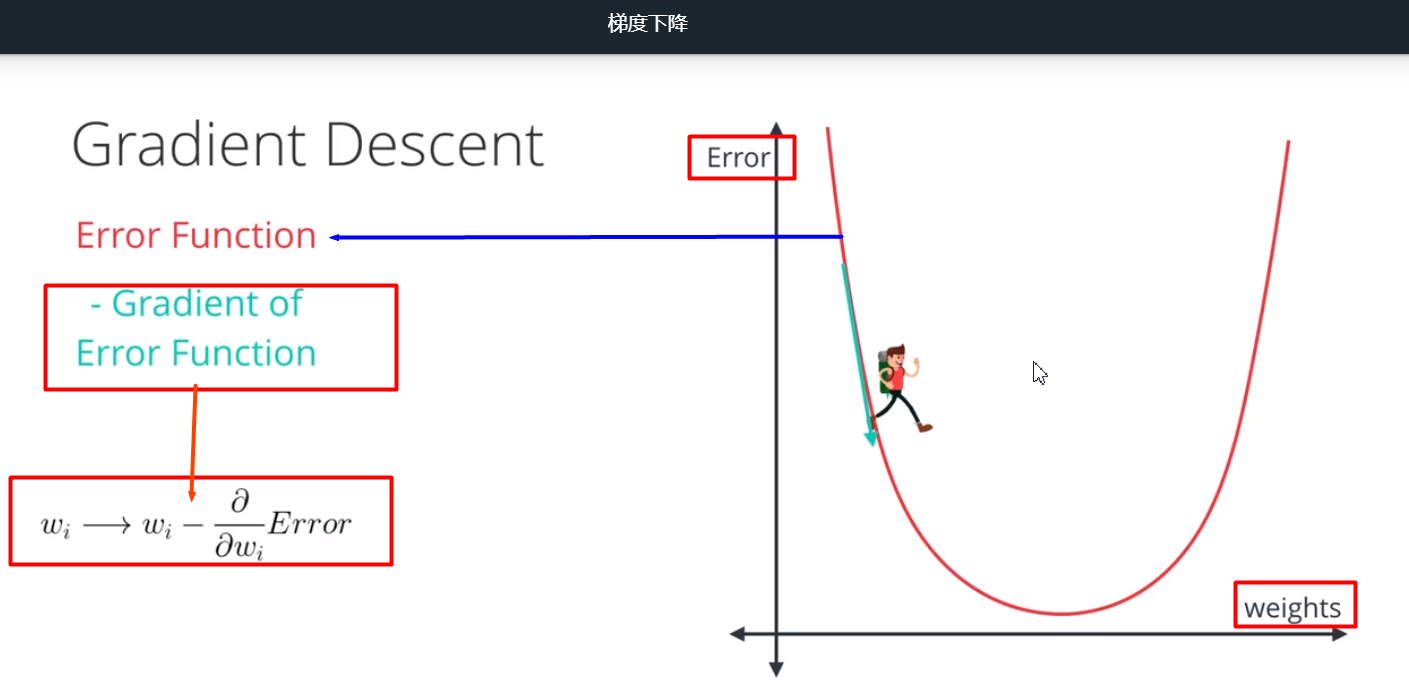
直线更加靠近点





梯度下降算法及优化

08． 梯度下降



说明：

这个图有两个维度,虽然在实际过程中 绘图可能会包含多个维度。

权重（weights）位于 x 轴

误差(Error)位于 y 轴

有一个这样的误差函数:红色曲线

必要解释

y= wx + b

y’ 是数据的标签，正确的值。

Error function = y – y’

Error 是weight 对应Error function函数的值

w = weights

梯度

梯度指出了函数增加至最大时的方向

导数为正，表示 x轴增大的方向有最大值

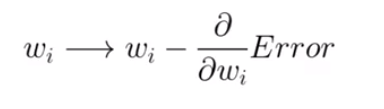
导数为负，表示 x轴减小的方向有最大值

梯度下降法

梯度下降方法实际上 针对 权重的导数 或 误差梯度函数,

由于梯度为负值 表明方向朝下时 函数可以减少到最小。所以我们要向梯度负值方向迈出一步，就能找到误差的最小值。

梯度下降法的目的是更新 线性方程的权重，当通过求导，找到最小值的方向后，更新权重，既 W I =W i -W i 的误差导数



结论：

由上可知，由于梯度是负数，那么wi的权重是在增加的，而不是在减小。

在实际生活中 我们会用这个导数 乘以学习率 因为我们只想迈出一小步

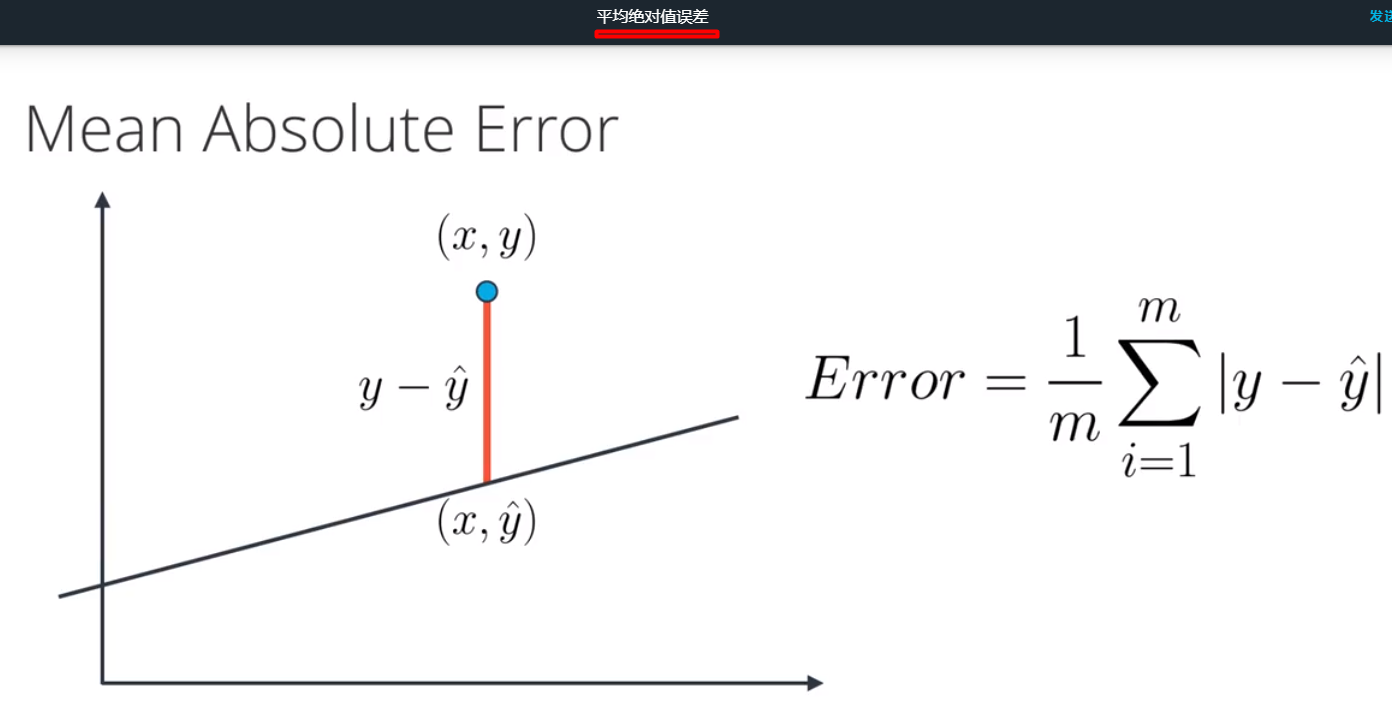
这表明误差函数正在下降 并且我们更加接近最小值

如果反复进行几次,我们将会得到最小值 或者误差极小的较优值

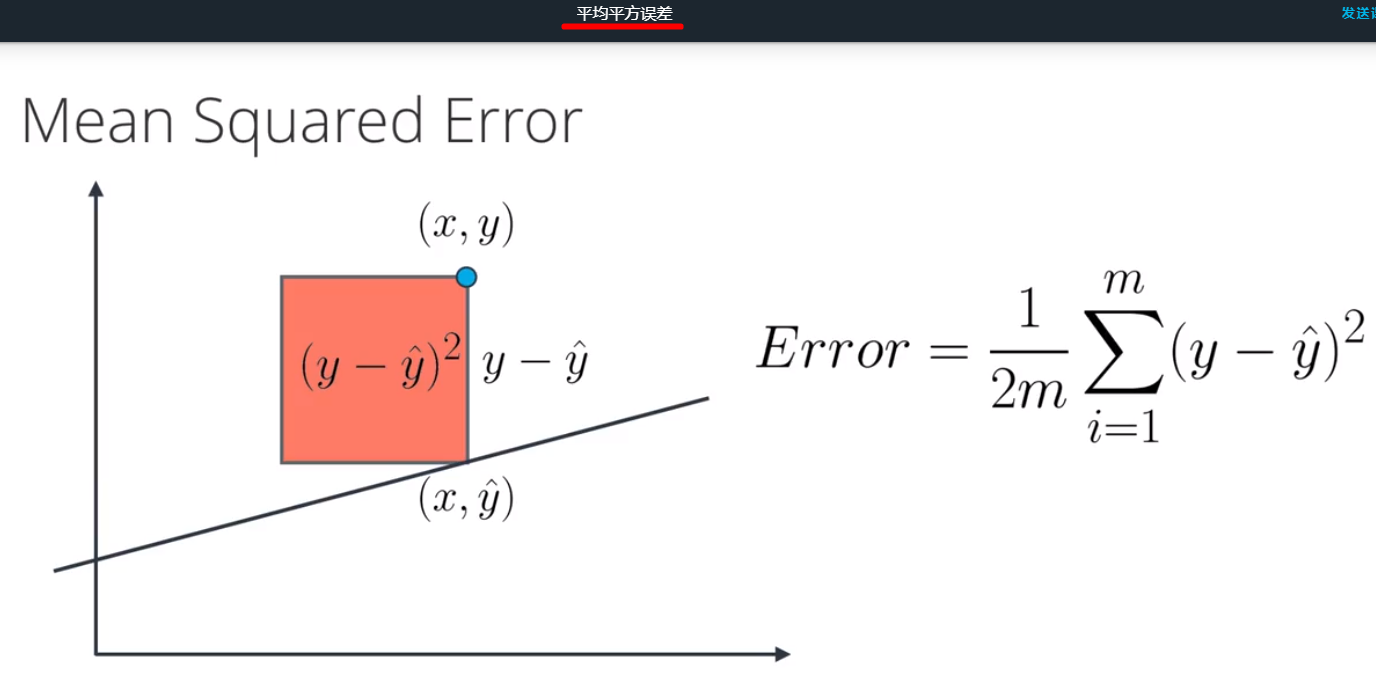
我们一旦到达这个点,就能得到线性回归问题的较优解,这就是梯度下降法的内容

误差函数：平均绝对误差 和 均方误差

# 09．平均绝对值误差



# 10．平均平方误差



# 11．最小化误差函数

以上学习了两种算法 用于让一条直线拟合一些点：

1.一种是使用技巧 ,包括绝对值技巧和平方技巧

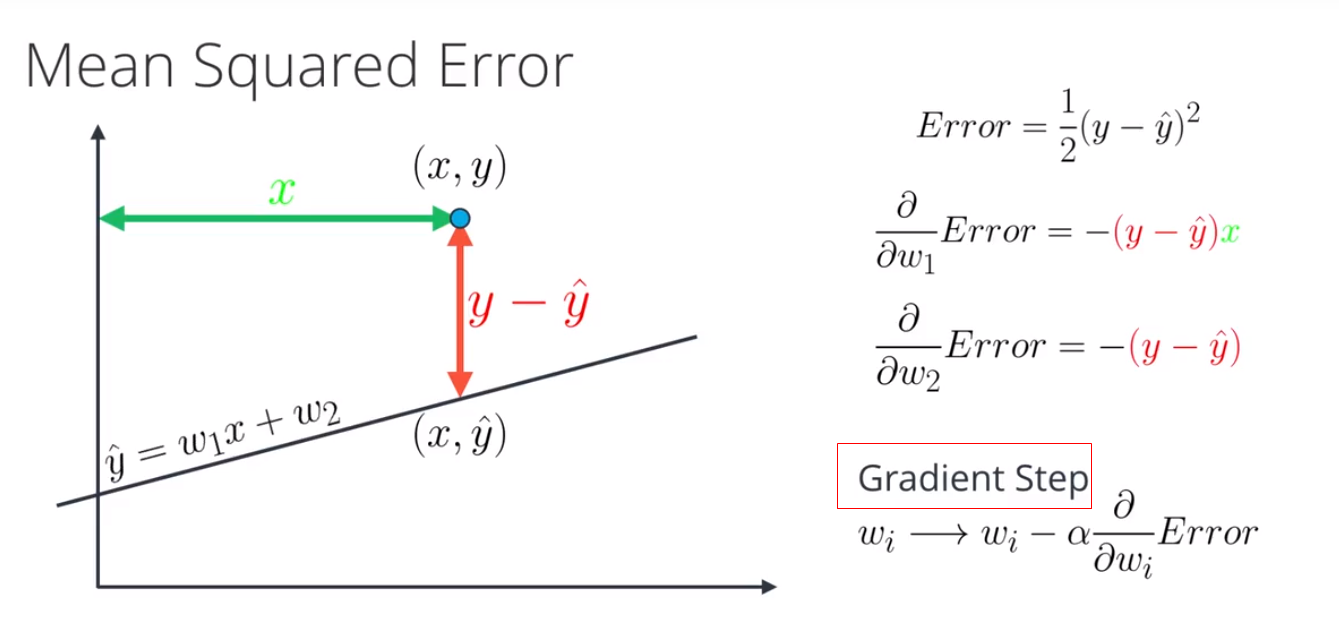
2.一种是最小化任何误差函数,即 平均绝对误差 和 均方误差

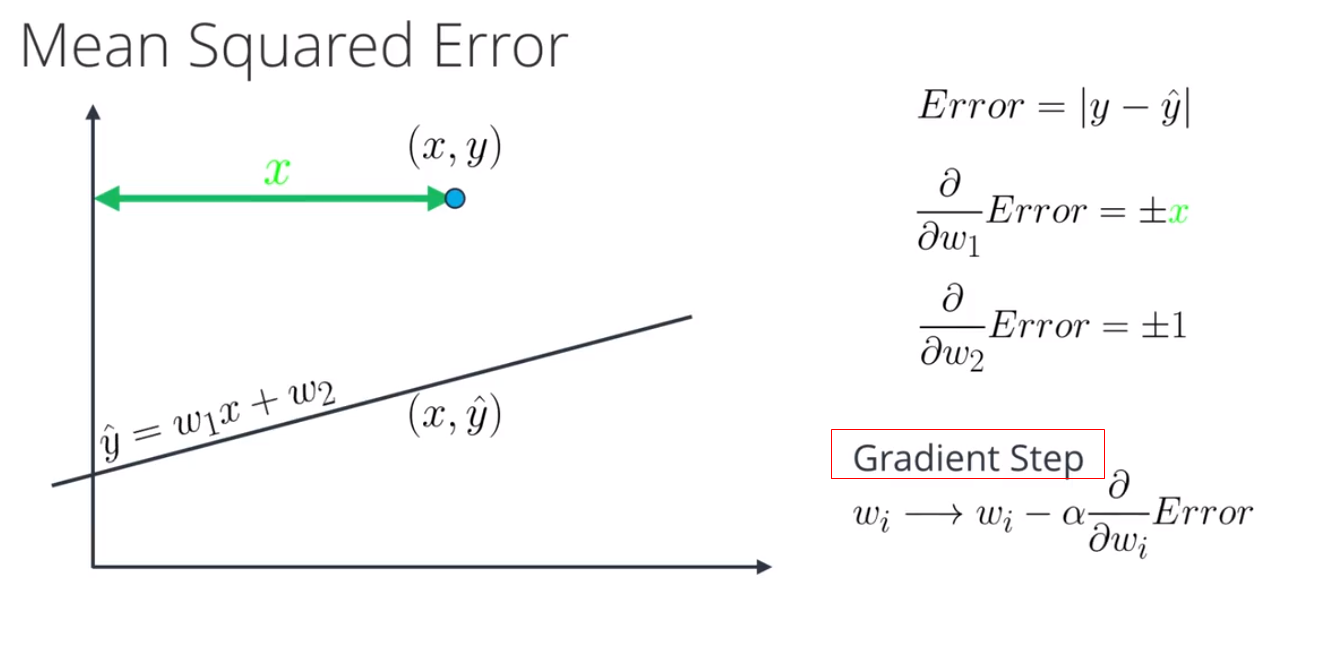
3.其实这两种算法是一样的。

结论：

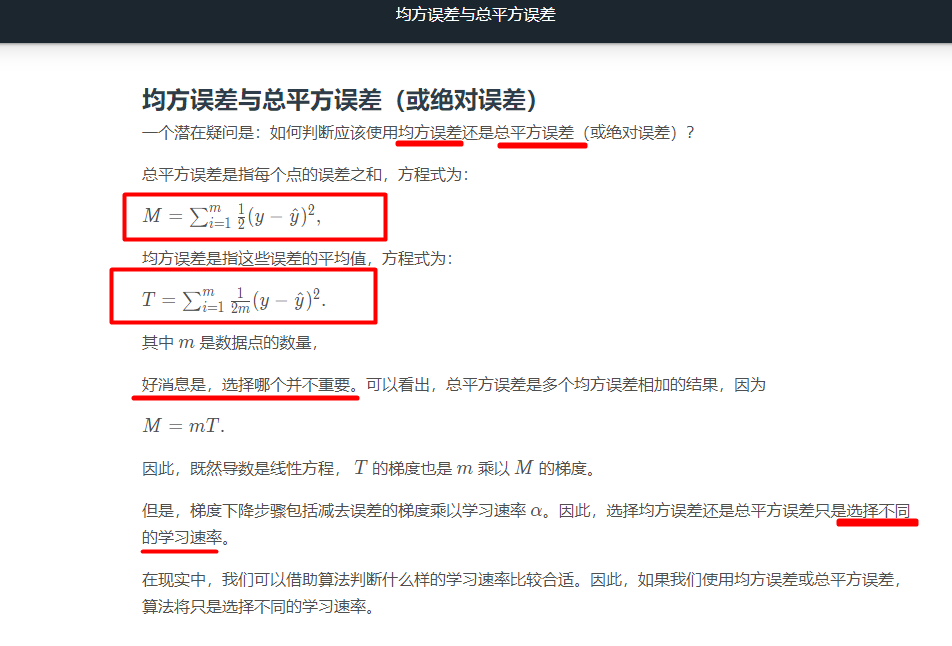
最小化平均绝对误差时，梯度下降法与绝对值技巧完全相同。

最小化平方误差时 梯度下降法与平方技巧完全相同。



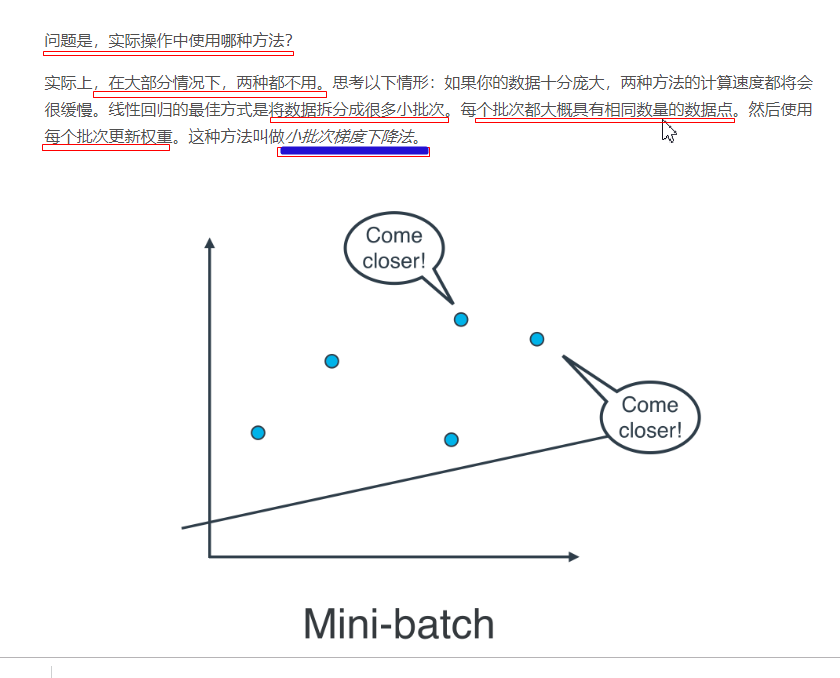


# 12．均方误差与总平方误差

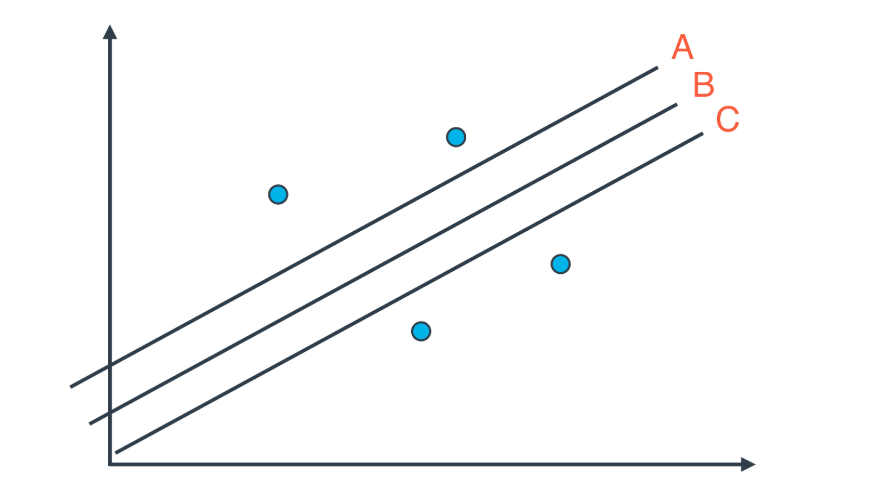


# 13．小批量梯度下降法





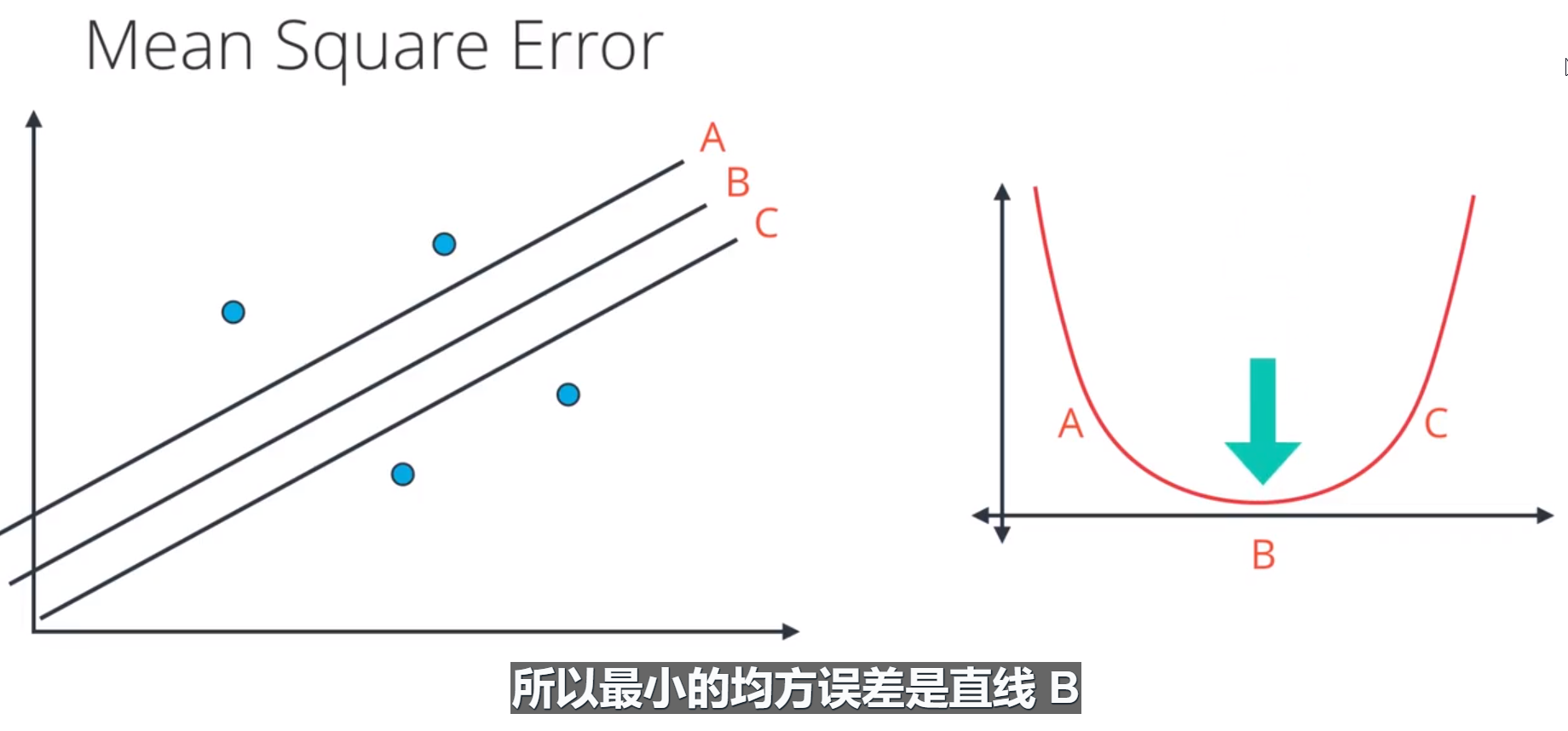
# 14．绝对值误差VS平方误差（Lab）



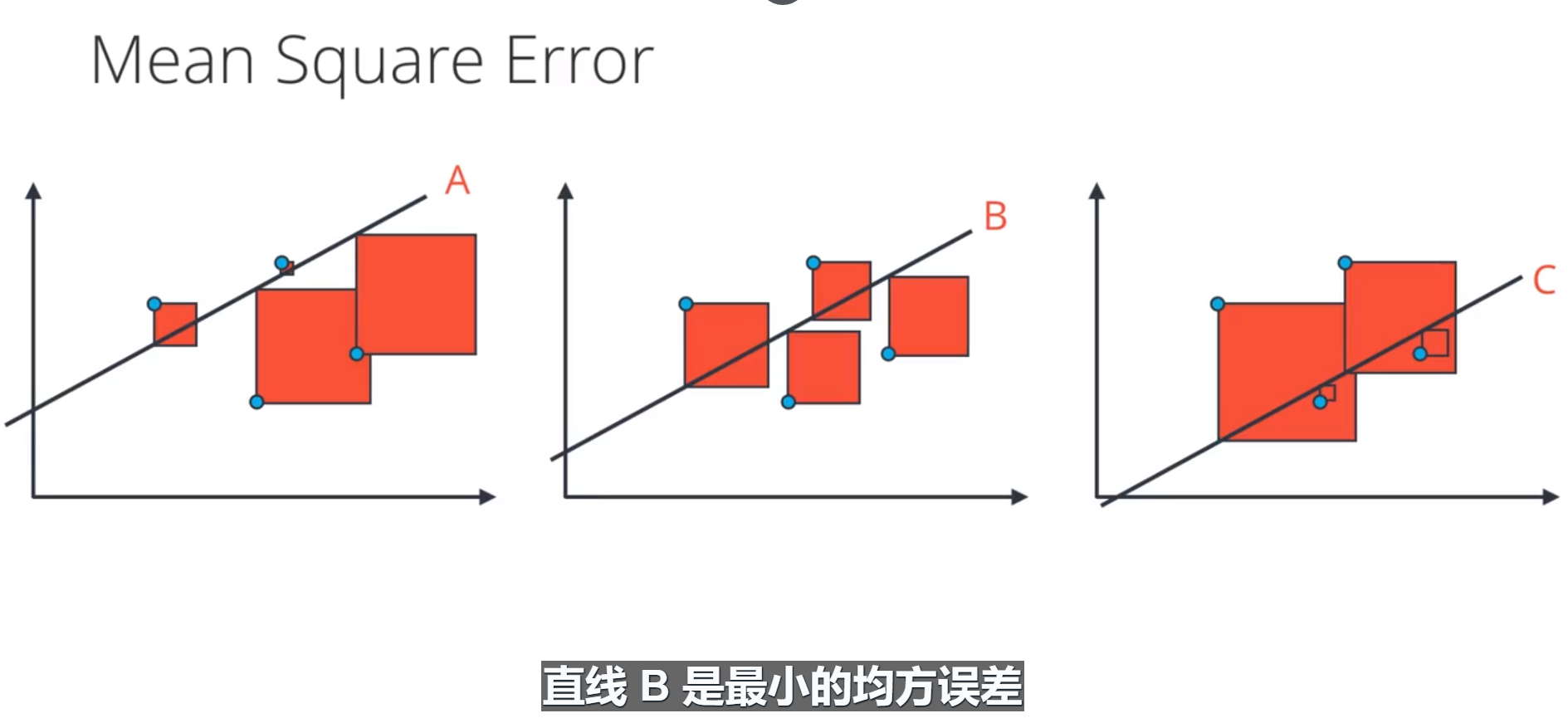




均方误差解法1

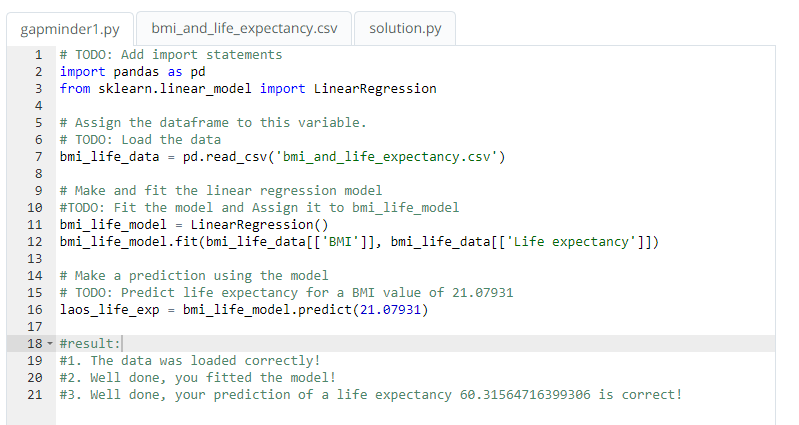


均方误差解法2

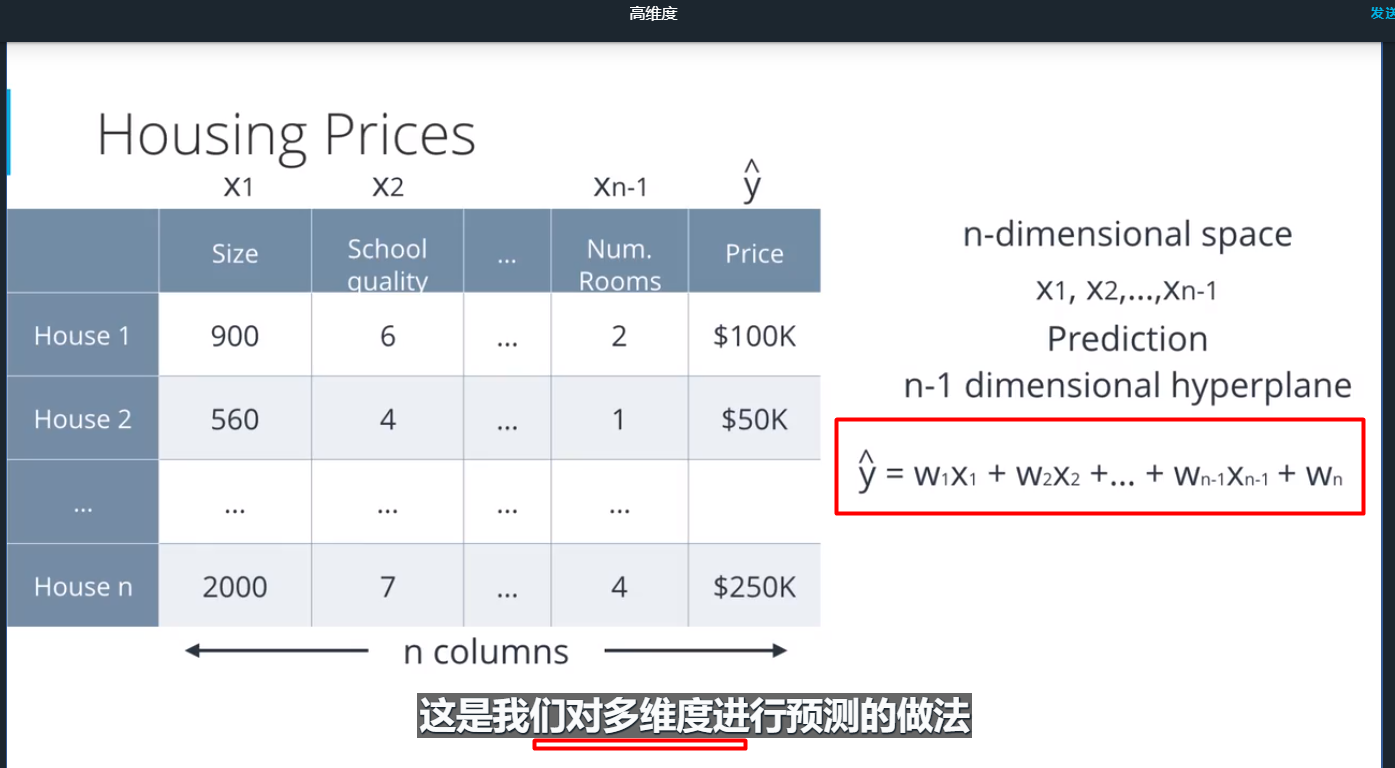


# 15．scikit-learn中的线性回归



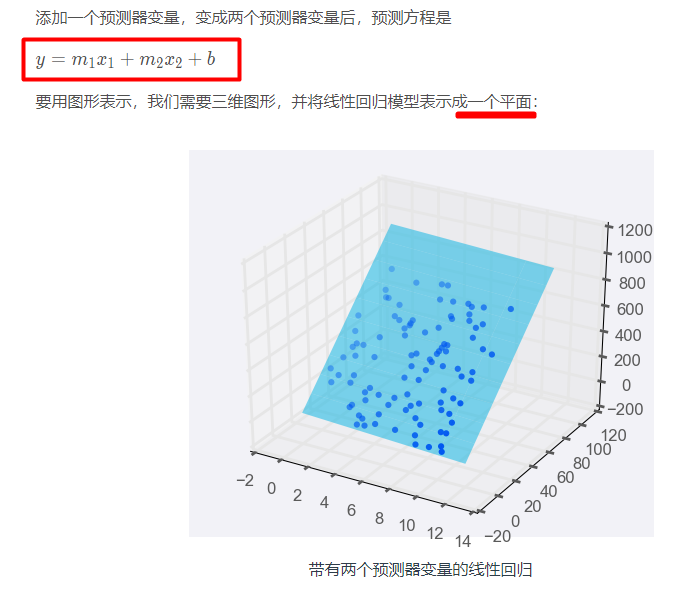


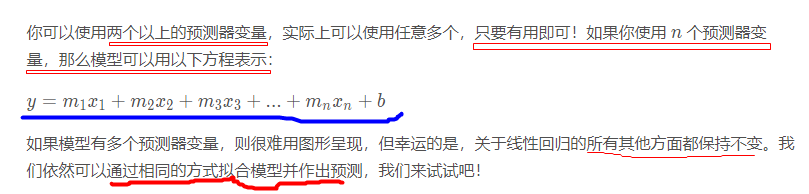
# 16．高维度



# 17．多元线性回归



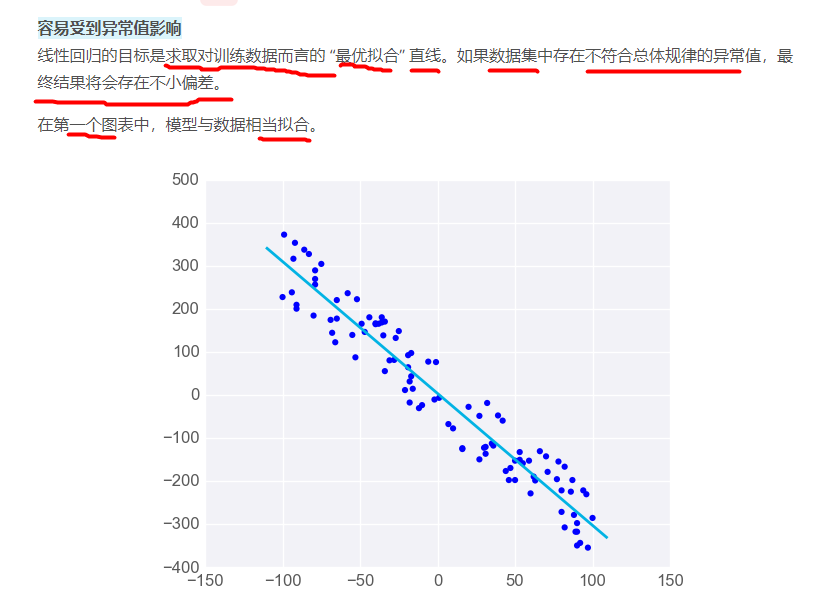


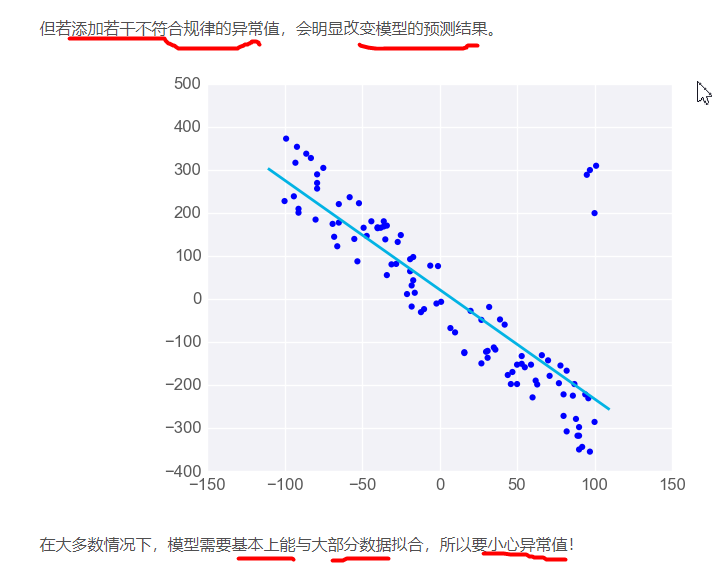


# 18．解数学方程组

# 20．线性回归注意事项※



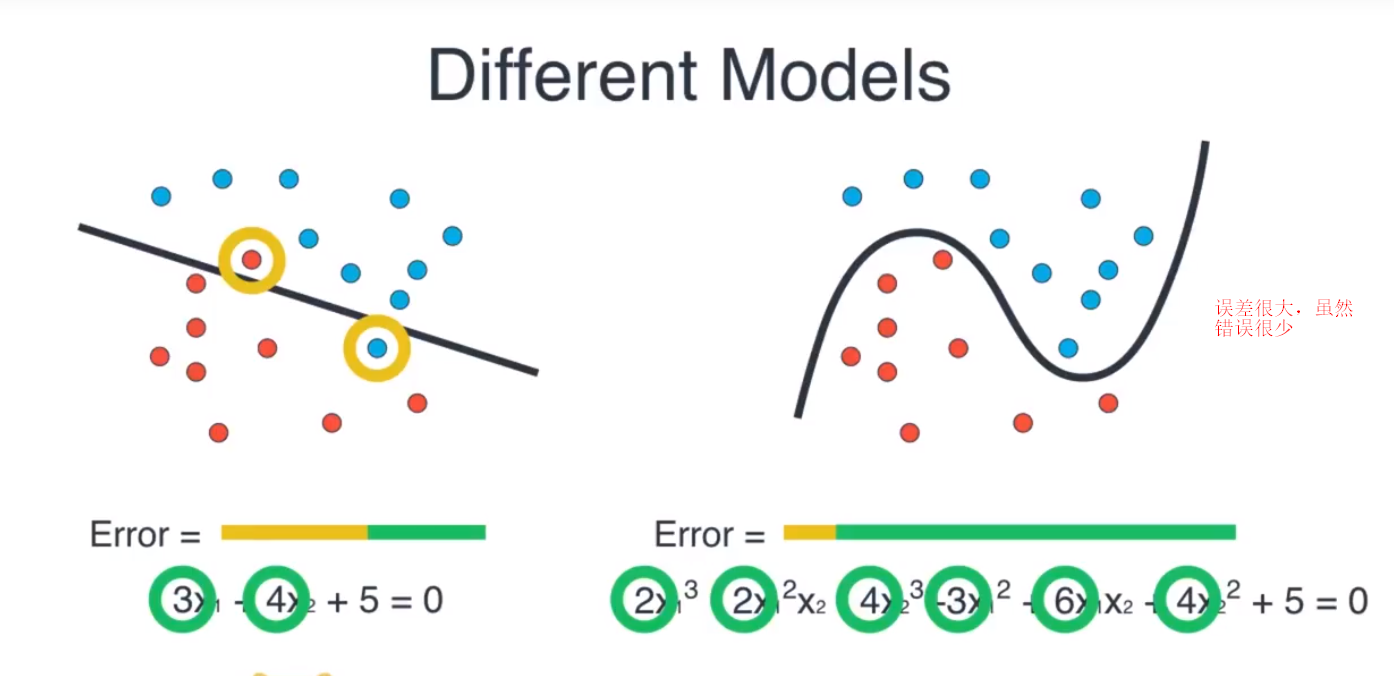




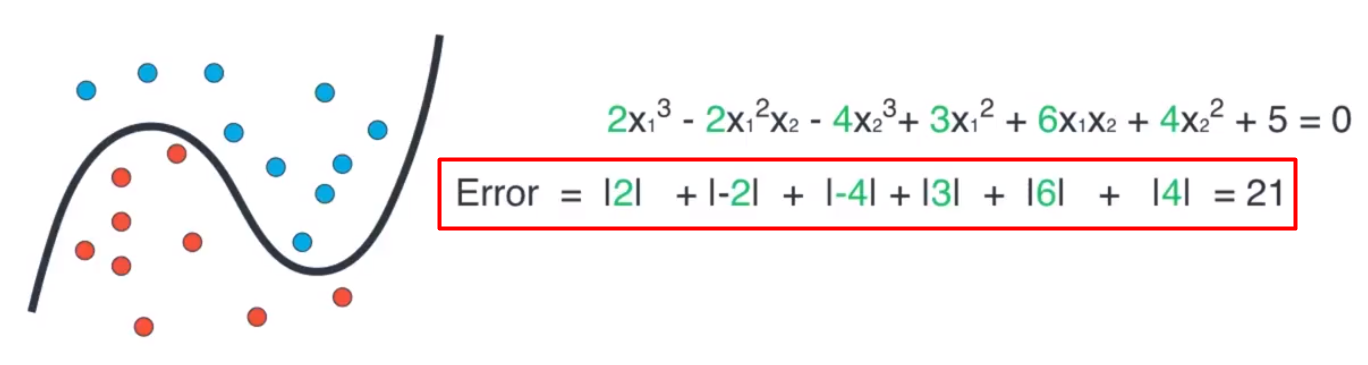
# 22．正则化

正则化主要作用是，确保模型不会过度拟合

正则化方法是在训练数据不够多时，常常会导致过拟合（overfitting）。这时向原始模型引入额外信息，以便防止过拟合和提高模型泛化性能的一类方法的统称。

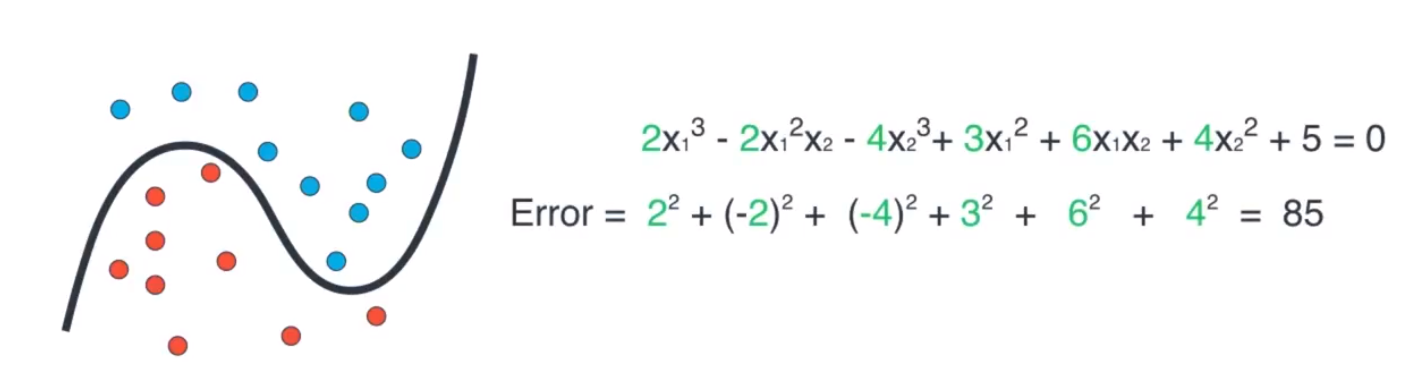


L1正则化（是系数绝对值相加）





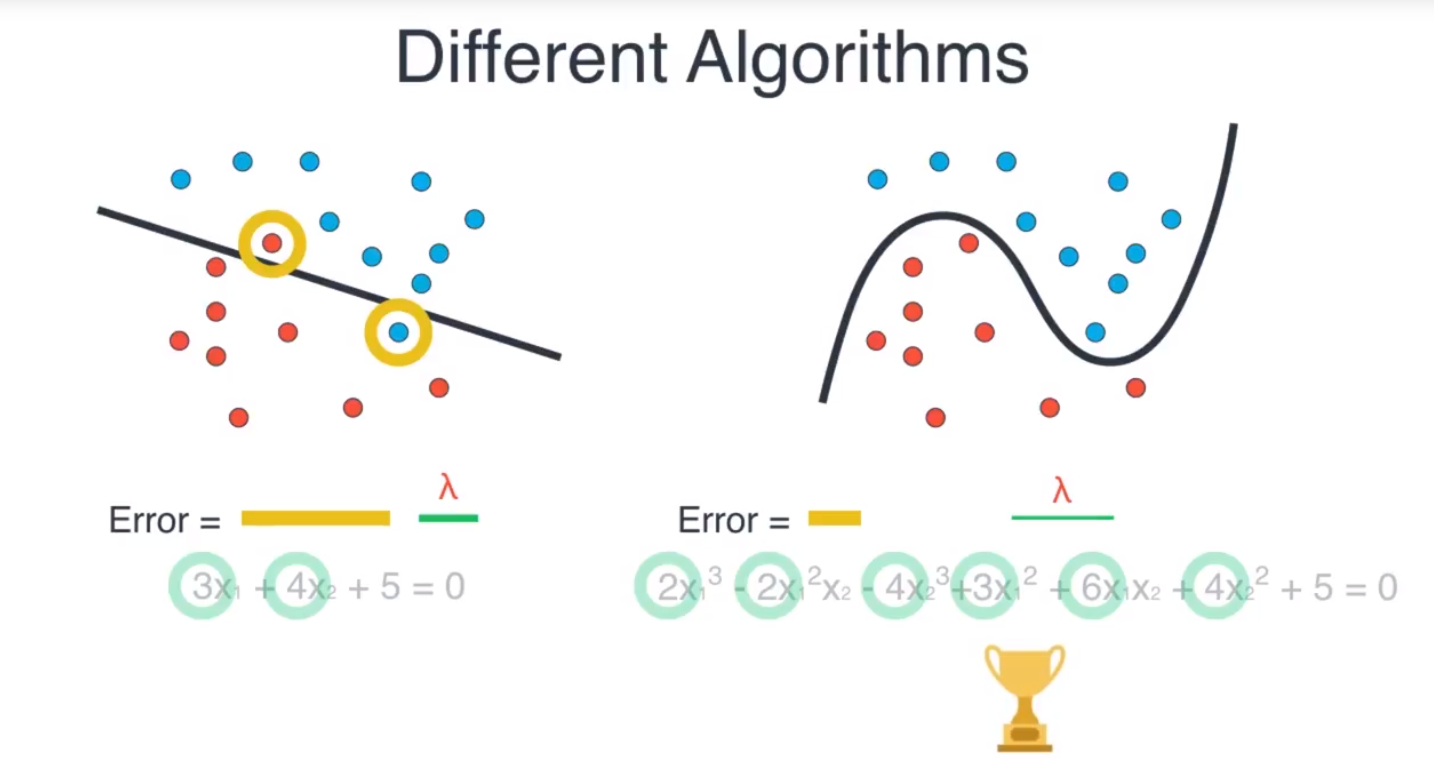
L2正则化（是系数平方和 相加）

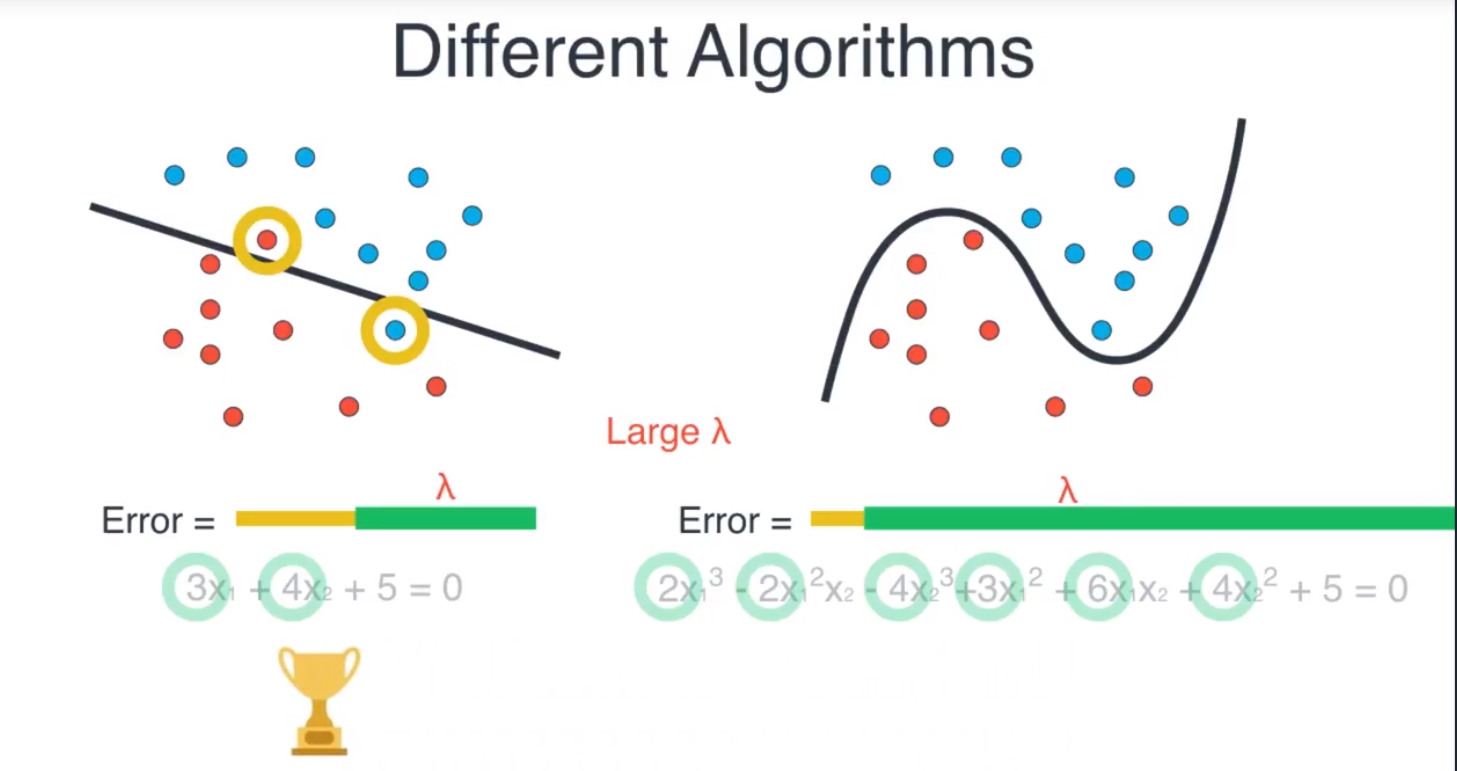




调优参数：**λ**

λ 的用途是 乘以 复杂误差（L1或L2），乘以那个误差可以变大或变小，如果λ增大，会更多地惩罚 复杂误差， λ减小 ，则会减小 复杂误差 的惩罚。





L1、L2的选择

