机器学习基础与实践

梯度下降

目录

CONTENTS

01 梯度下降在机器学习中的作用

数据

训练

模型

优化问题

找到一组参数 θ^* ，使用这组参数表示的模型 $f(\theta^*)$ 是更好的。

什么样的模型是好的模型?

用损失函数去描述模型与数据的差距。

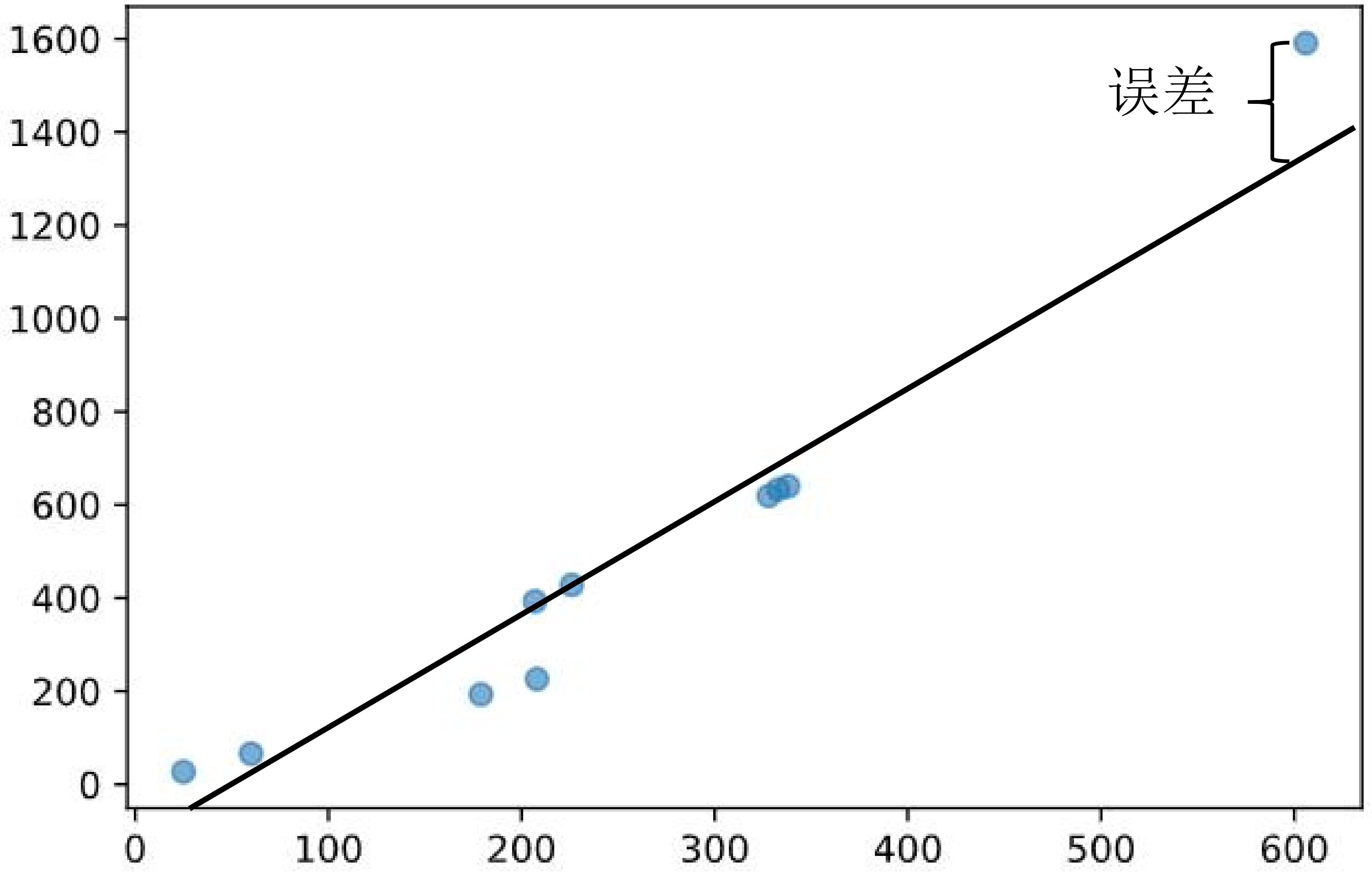
如何找到这个比较好的参数 θ^* ？

梯度下降。

02 梯度下降

假设有一组数据，我们需要建立一个模型描述 x 和 y 之间的关系。

```
x = [338., 333., 328., 207., 226., 25., 179., 60., 208., 606.]
y = [640., 633., 619., 393., 428., 27., 193., 66., 226., 1591]
```



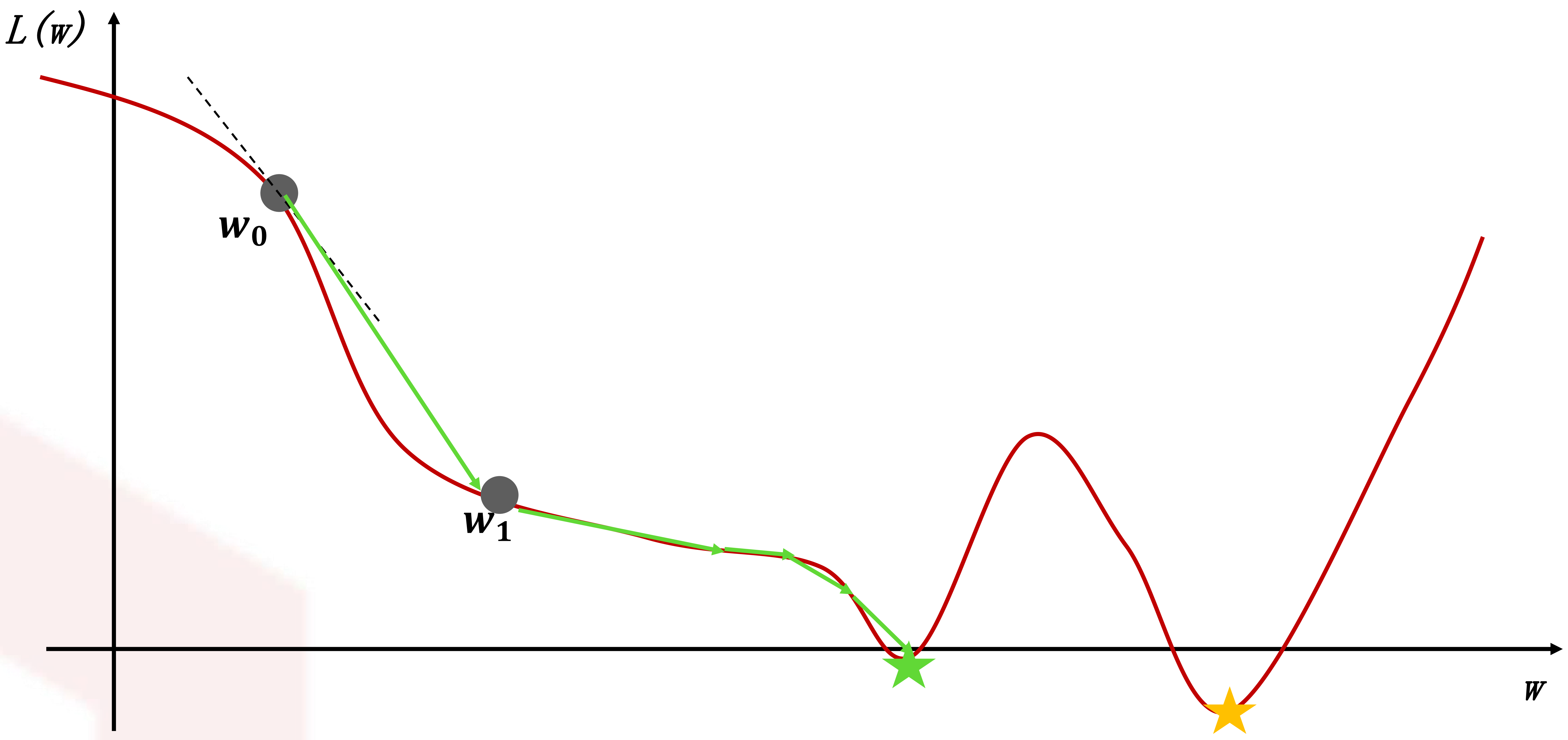
1. 首先我们把这批数据可视化。
2. 通过图像可以大胆假设我们的模型：$f(x) = w * x + b$
3. 用损失函数描述模型与数据的误差：$L(w,b) = \frac{1}{10}\sum_{n=1}^{10}\left(\hat{y}^n - f(x)\right)^2$
4. **目标：找到让损失函数 L 最小的参数 w^*, b^***：$w^*, b^* = \underset{w,b}{argmin}\, L(w,b)$

$$\boxed{w^*, b^* = \underset{w,b}{argmin}\, L(w,b)}$$

梯度下降

02 梯度下降

假设我们要优化的函数 L(w) 只有一个参数 w ，目标是找到使 L(w) 最小的 w。



梯度下降的过程

1. 随机找一个参数 w_0；

2. 计算在 w_0 处 $L(w)$的梯度：$\frac{dL}{dw}|_{w=w_0}$

3. 在原w_0 的位置沿负梯度方向移动；

$\eta\frac{dL}{dw}|_{w=w_0}$ 的距离到w_1. η为学习率，控制移动步长；

4. 重复2-3步骤，直到梯度为0或小于一定数值。

可以拓展到多参数的情况，对所有参数执行上述过程即可。

存在问题：①梯度下降不一定能达到全局最优解。 ② 越靠近最优，梯度可能更小，收敛速度会减慢。

03 实例推导

假设有一组数据，我们需要建立一个模型描述 x 和 y 之间的关系。

```
x = [338., 333., 328., 207., 226., 25., 179., 60., 208., 606.]
y = [640., 633., 619., 393., 428., 27., 193., 66., 226., 1591]
```

假设模型： $f(x) = w * x + b$

损失函数： $L(w,b) = \frac{1}{10}\sum_{n=1}^{10}\left(\hat{y}^n - f(x^n)\right)^2 = \frac{1}{10}\sum_{1}^{10}\left(\hat{y}^n - (w * x^n + b)\right)^2$

目标： $w^*,\ b^* = \underset{w,b}{argmin}\, L(w,b)$

参数迭代：

$$\boldsymbol{w}^{t+1} \leftarrow \boldsymbol{w}^t - \boldsymbol{\eta}\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{w}}$$

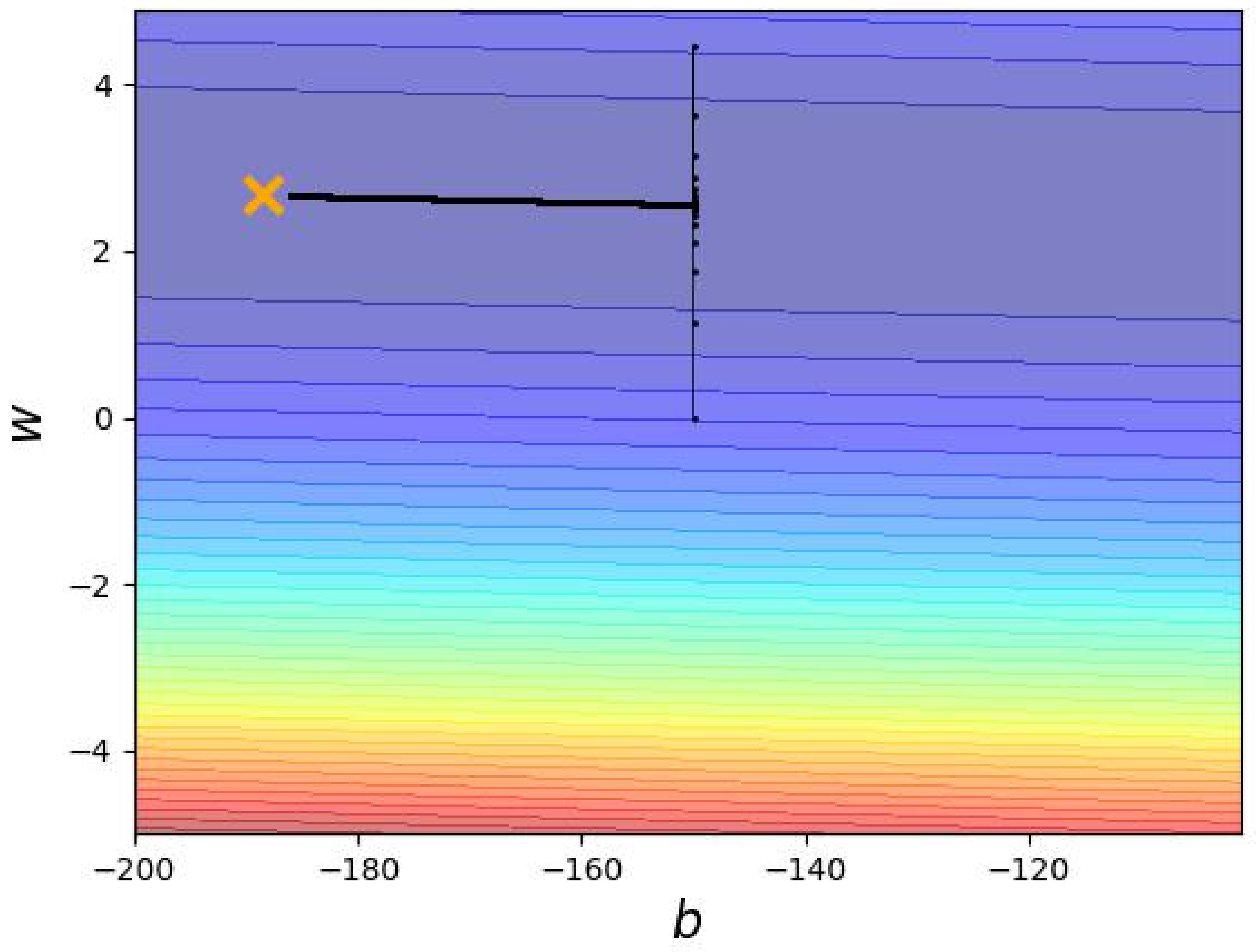
$$\boldsymbol{b}^{t+1} \leftarrow \boldsymbol{b}^t - \boldsymbol{\eta}\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{b}}$$

梯度：

03 实例推导

读入数据

梯度下降

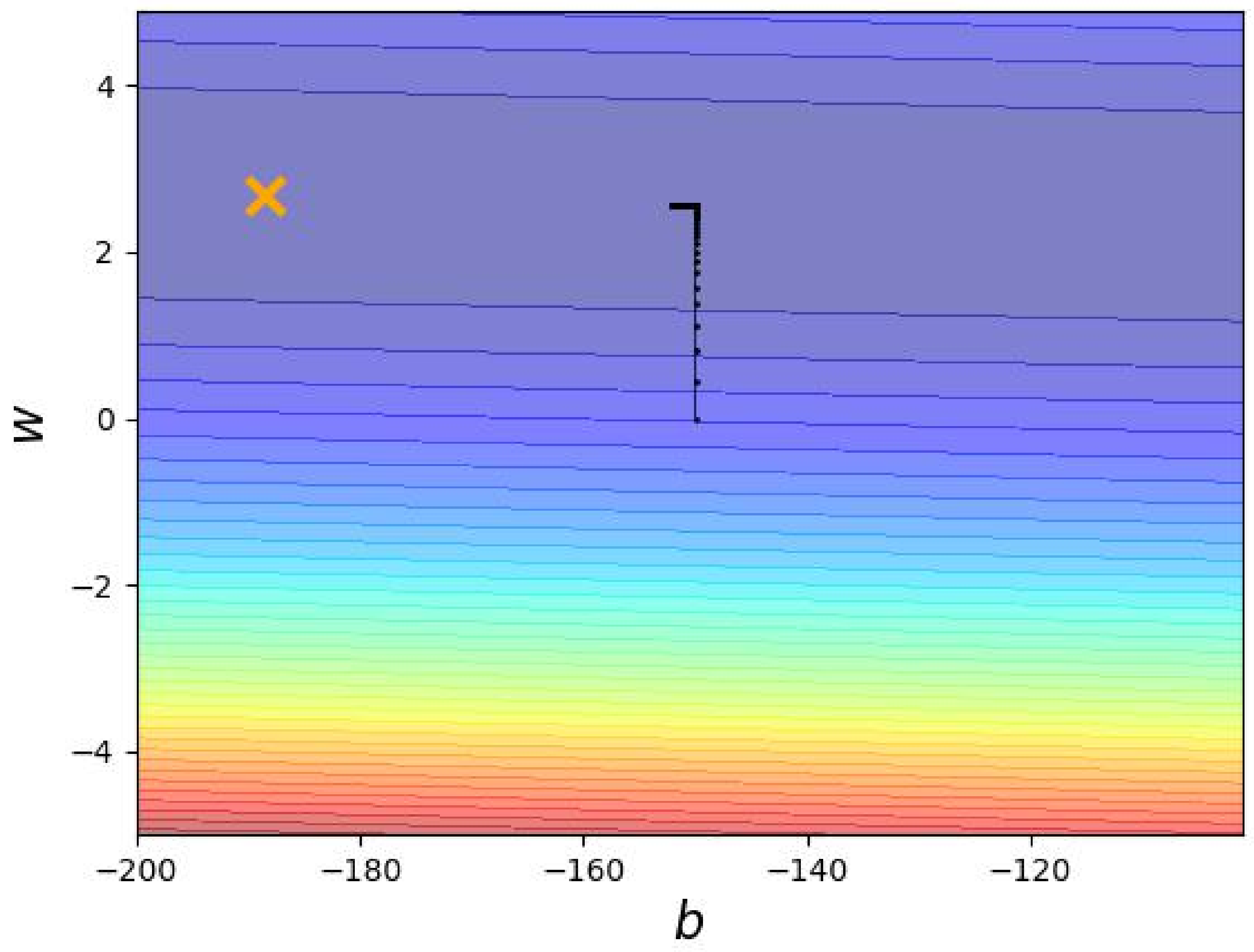


学习率：0.000001
迭代次数：500000

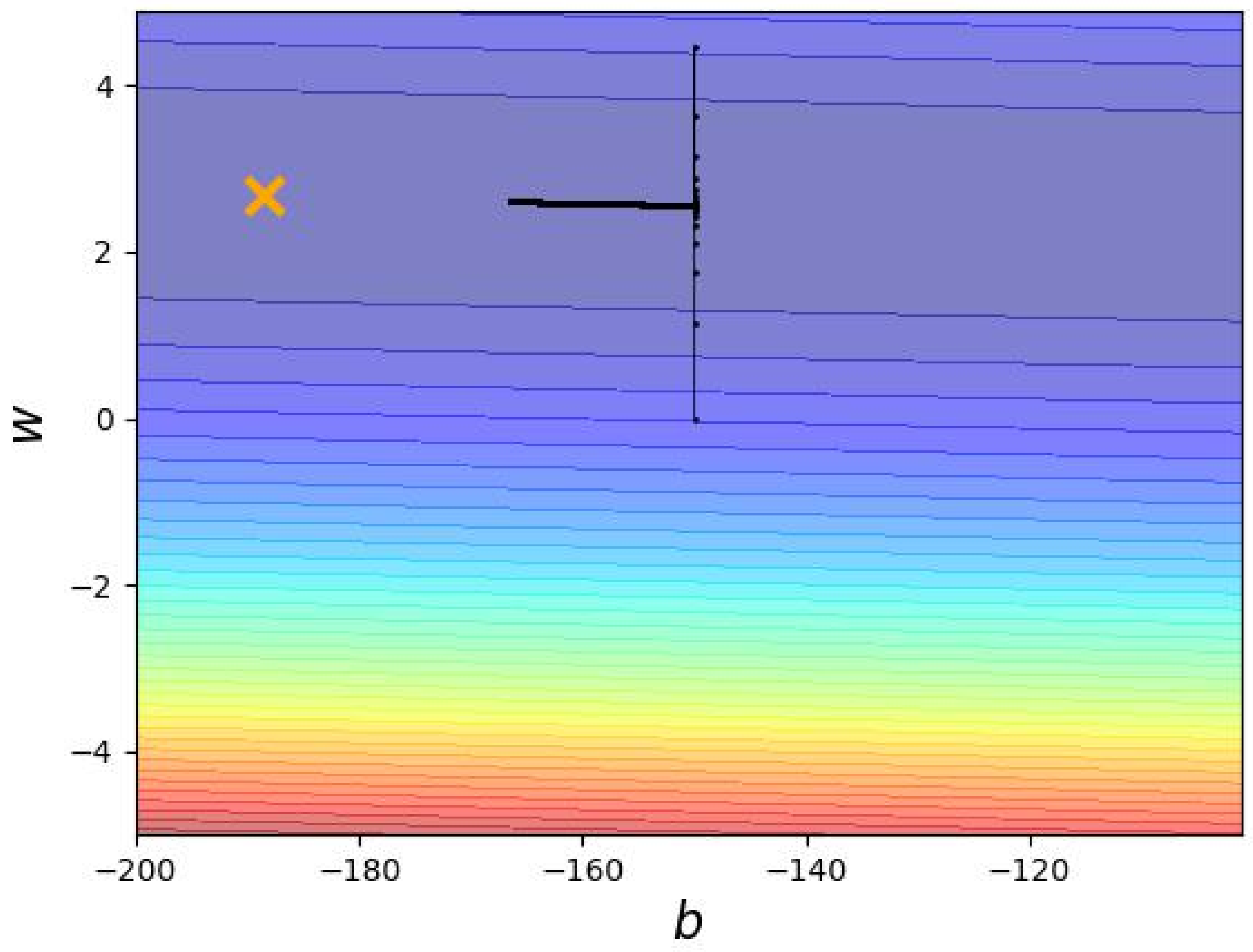
黑线表示迭代过程中参数 w 和 b 的变化。在学习率为0.000001，迭代500000次后模型收敛到最优参数附近。

可以试着调一调学习率，迭代次数看看有什么不同的结果，掌握根据结果调整学习率的方法。

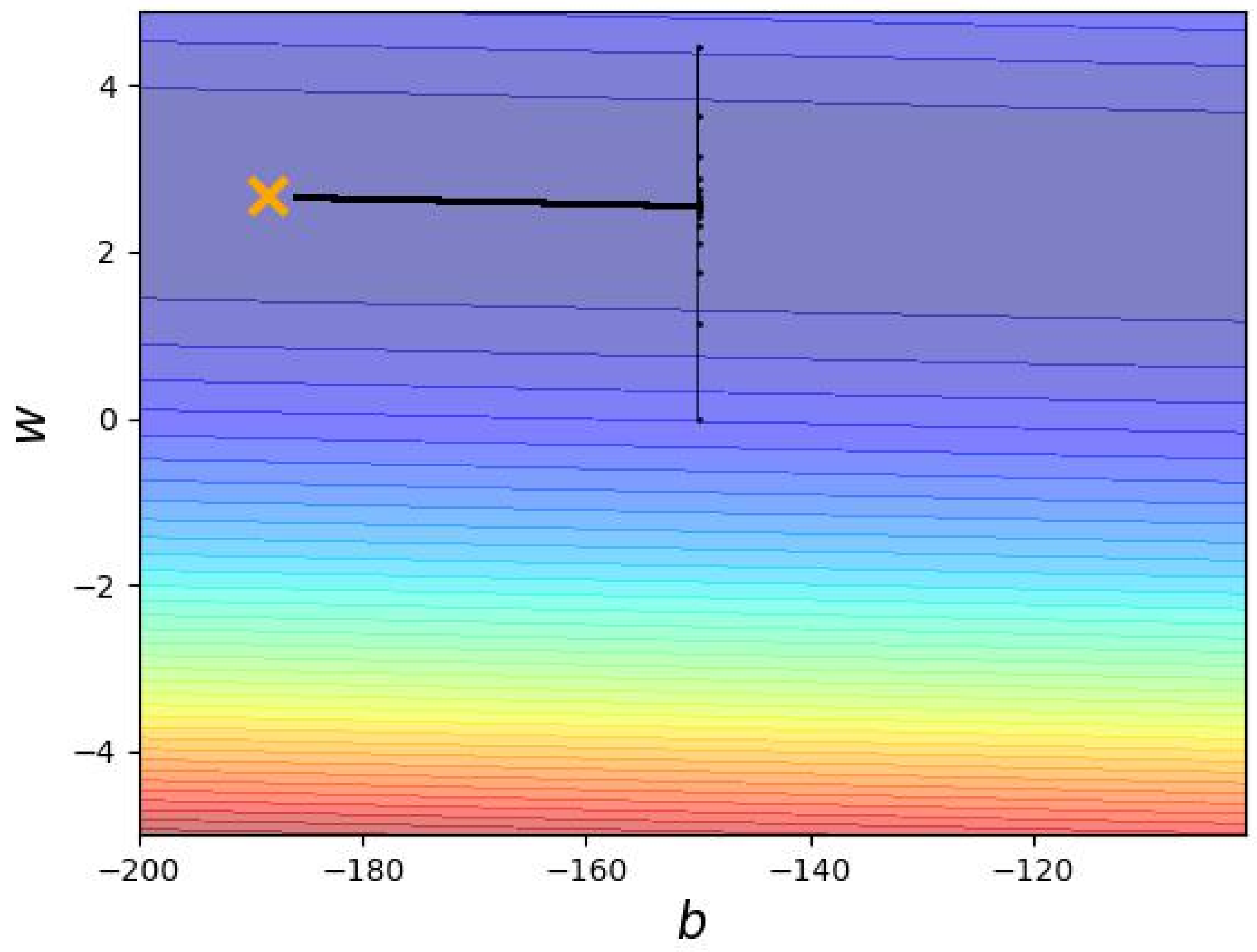
03 实例推导



学习率：0. 0000001
迭代次数： 100000



学习率：0. 000001
迭代次数： 100000



学习率：0. 000001
迭代次数：500000

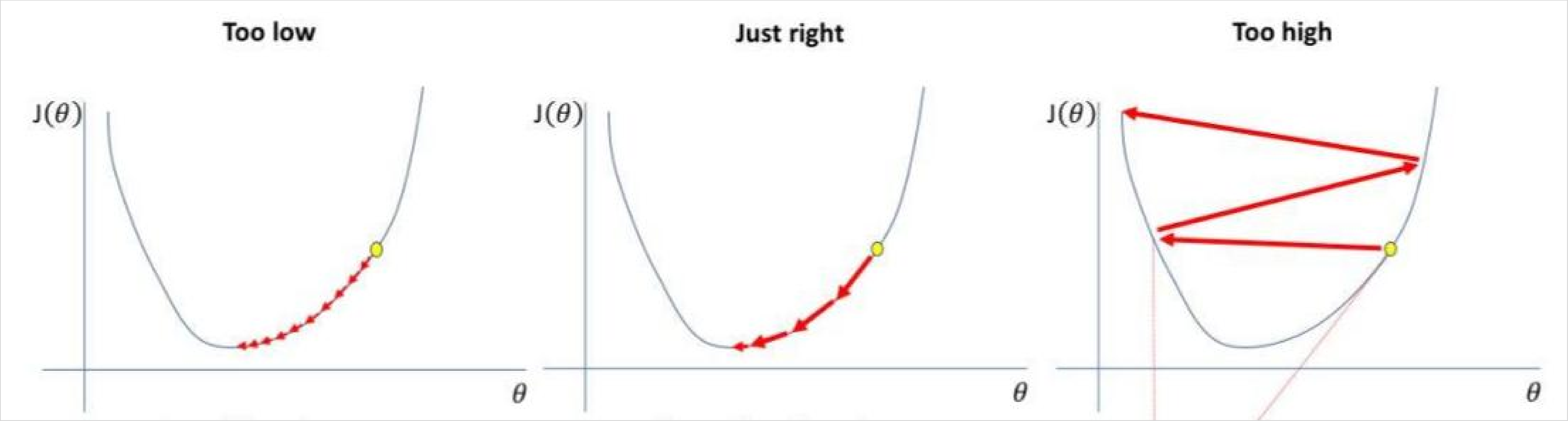
学习率太小 —— 学习率扩大10倍 → 学习率有点大，出现震荡，但在朝目标移动 —— 增加迭代次数 → 增加迭代次数后达到目标

有没有调整学习率的指导性方法？

04 学习率

$$w^{t+1} \leftarrow w^{t} - \eta \frac{\partial L}{\partial w}$$

学习率对于梯度下降的影响



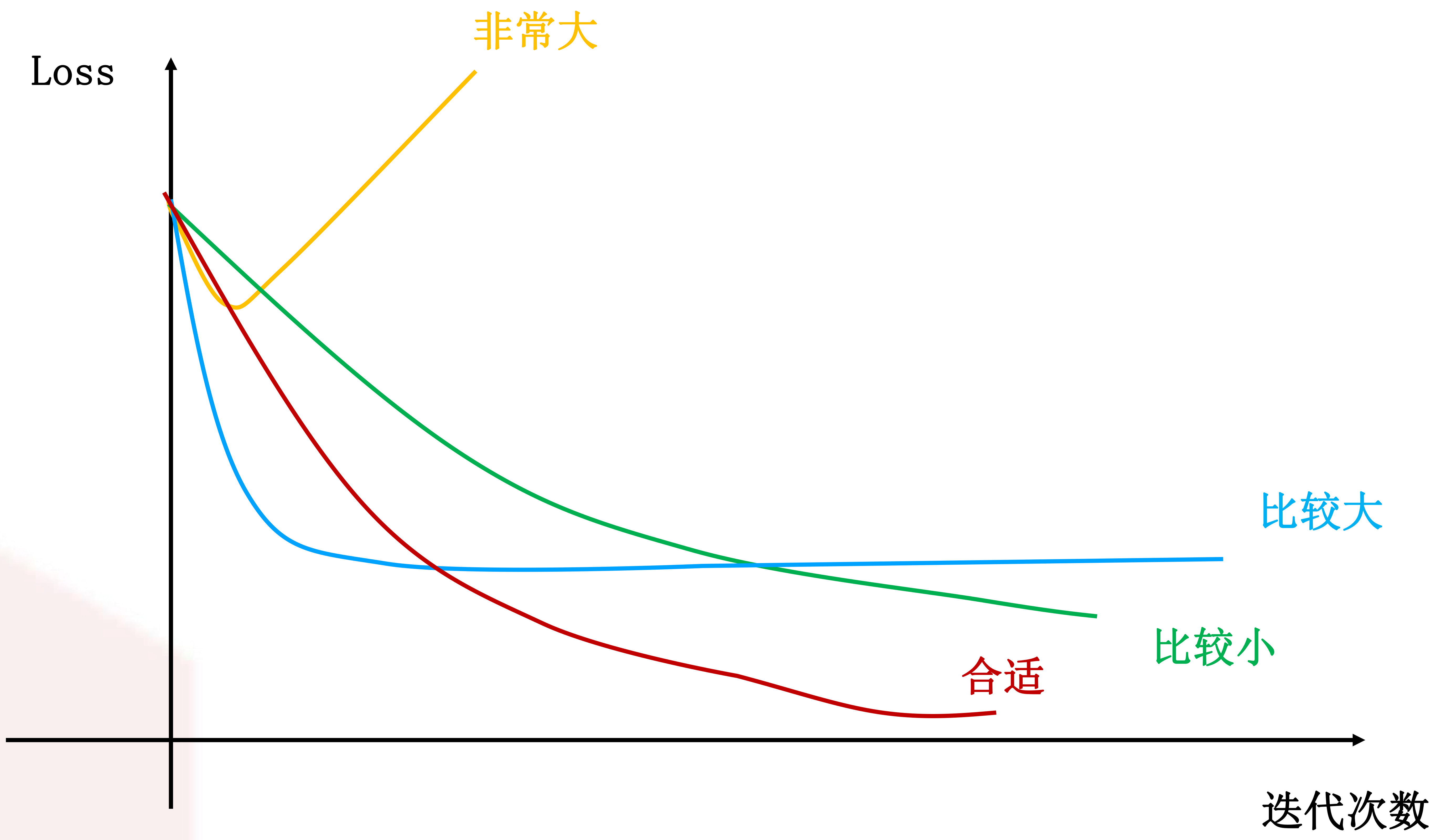
学习率太小：收敛较慢，达到最小值需要更多的迭代次数。

学习率适合：可以很快地到达最小值。

学习率太大：无法达到最小值，并不会收敛，结果是发散的。

04 学习率

如何调整学习率？



通常在训练过程中我们可以绘制随着迭代次数的增加 Loss 的变化曲线，根据曲线情况来调整学习率。

05 随机梯度下降（SGD）

$$\frac{\partial L}{\partial w} = \sum_{n=1}^{10} 2(\hat{y}^n - (w * x^n + b))(-x^n)$$

$$\frac{\partial L}{\partial b} = \sum_{n=1}^{10} 2(\hat{y}^n - (w * x^n + b))(-1)$$

批量梯度下降

$$\frac{\partial L}{\partial w} = 2(\hat{y}^n - (w * x^n + b))(-x^n)$$

$$\frac{\partial L}{\partial b} = 2(\hat{y}^n - (w * x^n + b))(-1)$$

随机梯度下降

区别：

批量梯度下降：在每轮迭代需要对整批数据求梯度然后求平均，再进行参数更新。$O(n)$

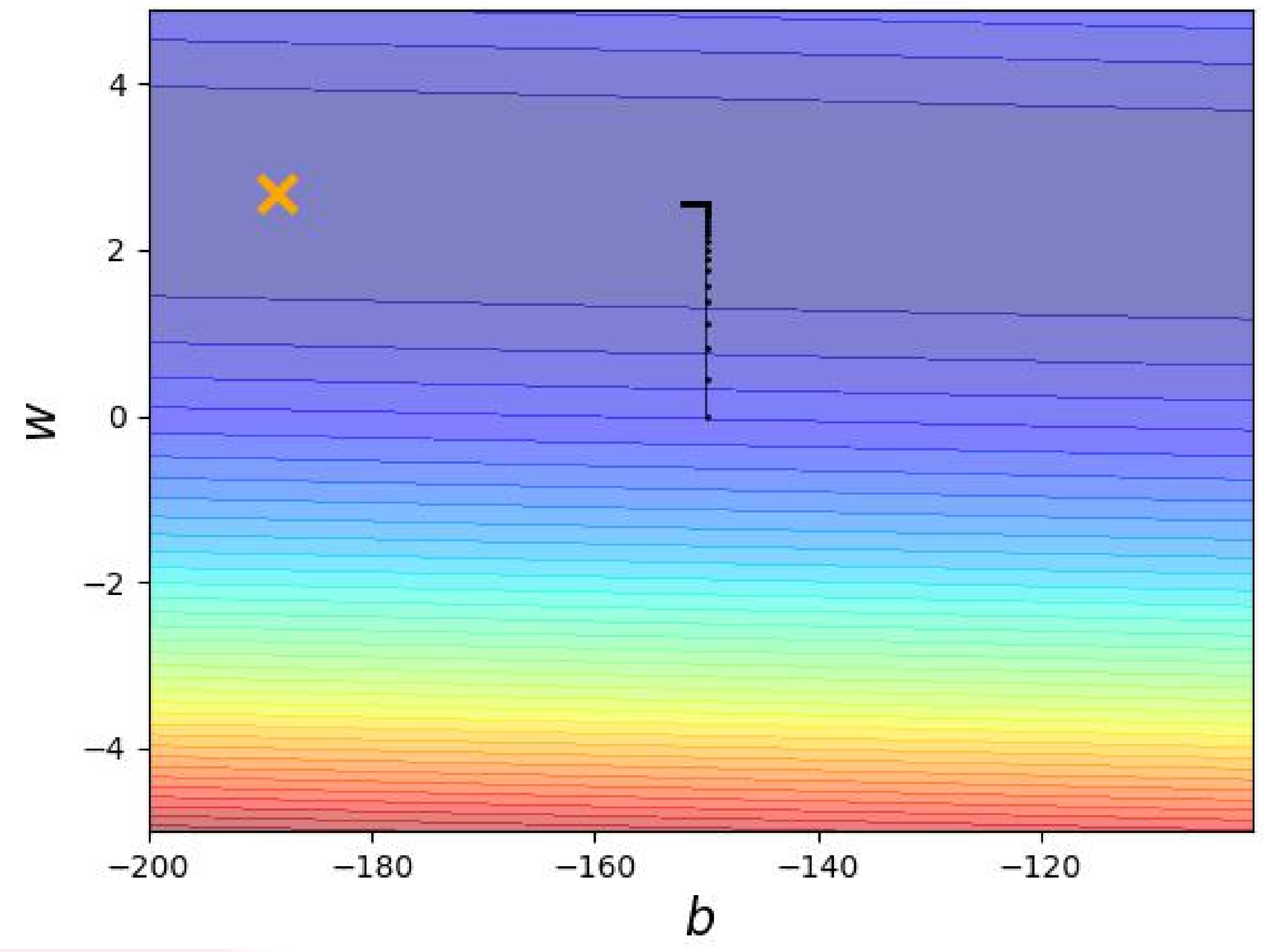
随机梯度下降：在每轮迭代只随机选取一个样本求梯度，再进行参数更新。$O(1)$

随机梯度下降是通过牺牲每次参数更新的精度来降低计算量，同时用更多的迭代次数来补上，所以最终也能实现目标。

优势：在数据量比较大的时候，可以极大地降低计算量。机器学习通常都会用SGD。

06 Adagrad

有没有更好的梯度下降方法？



学习率：0.0000001
迭代次数：100000

观察：参数 w 可以很快地到达最优位置，但参数 b 不能。

问题：

- 能不能自动调学习率？
- 能否针对不同参数设置不同的学习率？

方法：

1. 随着迭代次数增加学习率越来越小

$$\eta^t = \eta / \sqrt{t+1}$$

2. 根据不同参数设置不同的学习率

$$\eta^t = \eta / \sqrt{t+1}$$

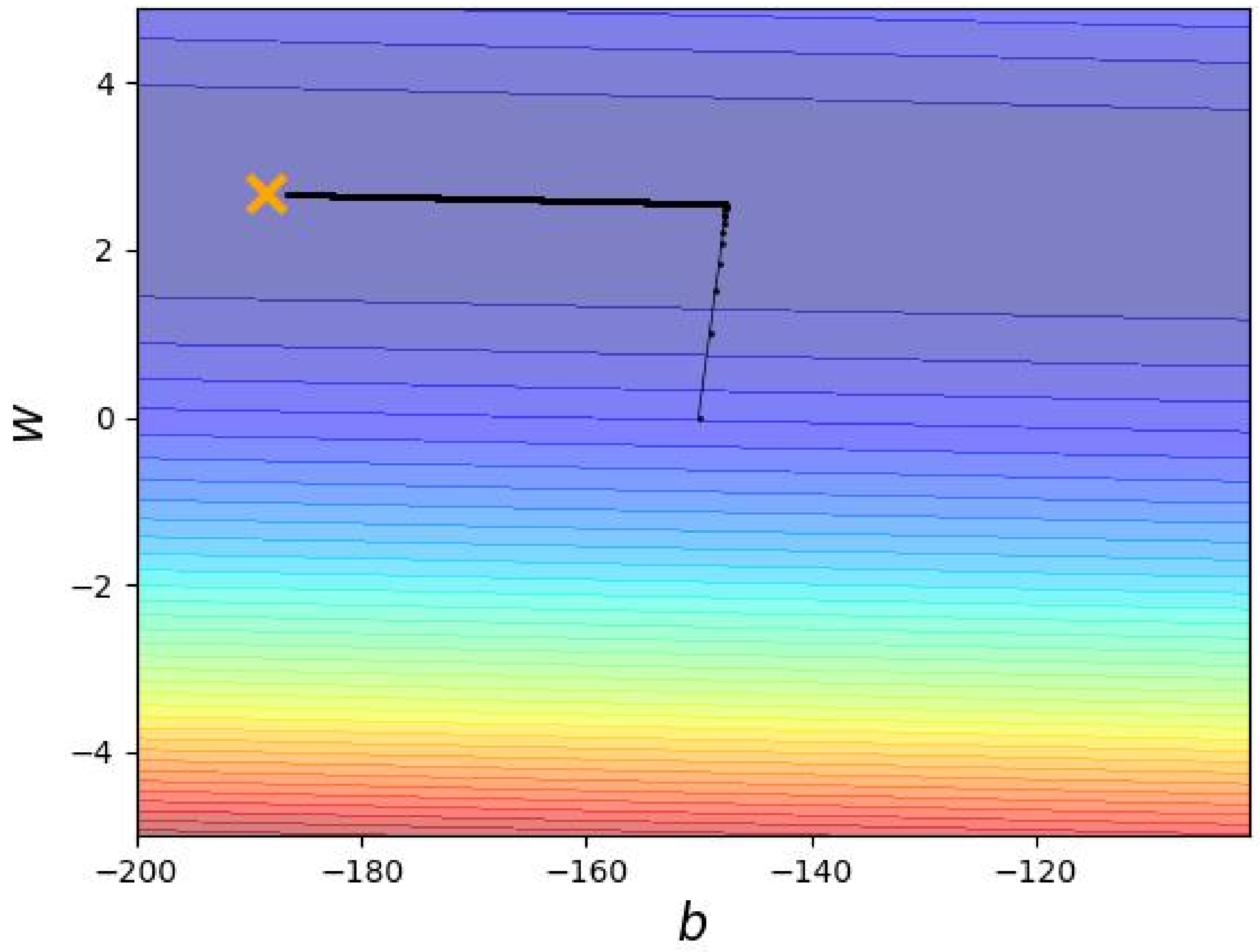
$$w^{t+1} \leftarrow w^t - \frac{\eta^t}{\sigma^t} g^t \longleftarrow \quad \sigma^t = \sqrt{\frac{1}{t+1} \sum_{i=0}^{t} (g^t)^2}$$

$$\boxed{w^{t+1} \leftarrow w^t - \frac{\eta}{\sqrt{\sum_{i=0}^{t} (g^t)^2}} g^t}$$

Adagrad 参数迭代公式

06 Adagrad

实例演示



学习率：1
迭代次数：10000

和之前的批量梯度下降的结果相比，Adagrad 仅用10000次迭代就达到了目标点，并且迭代过程比较稳定。

07 梯度下降方法比较

	SGD	AdaGrad
缺点	• 选择合适的学习率比较困难 • 容易收敛到局部最优	• 依赖人工设置一个全局学习率 • 中后期会使梯度逐渐趋于0，导致训练提前结束

本节所介绍的两种梯度下降算法，是最基本的方法，主要目的在于理解梯度下降的原理，目前实践中会使用更优秀的方法，如Adam，RMSprop。

09

本节小结

本节要点

1. 掌握梯度下降在机器学习中的作用。
2. 理解梯度下降的基本原理和整体框架。
3. 可以自己实现SGD，AdaGrad 算法。