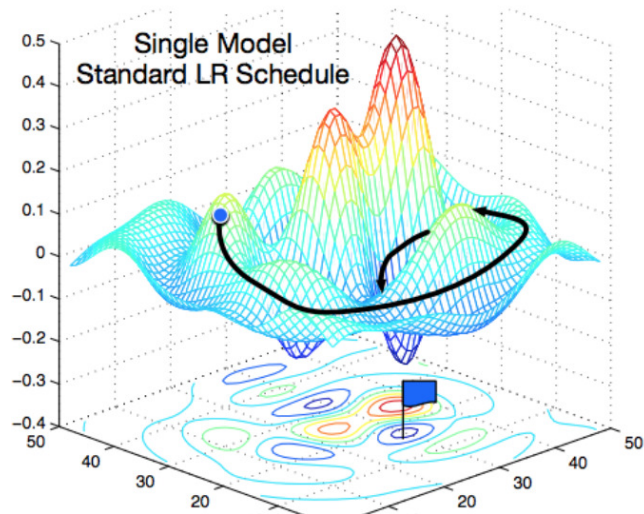




基础优化方法



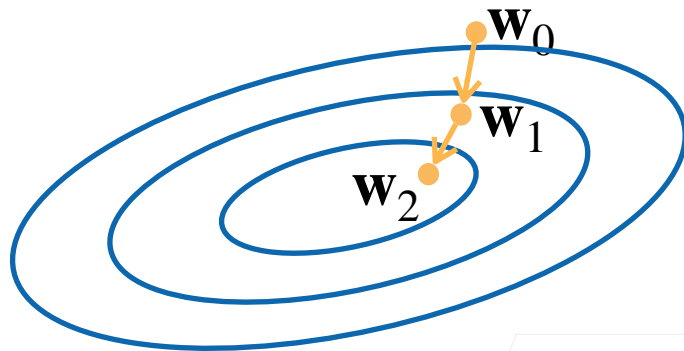
梯度下降



- 挑选一个初始值 \mathbf{w}_0
- 重复迭代参数 $t=1,2,3$

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \eta \frac{\partial \ell}{\partial \mathbf{w}_{t-1}}$$

- 沿梯度方向将增加损失函数值
- 学习率：步长的超参数

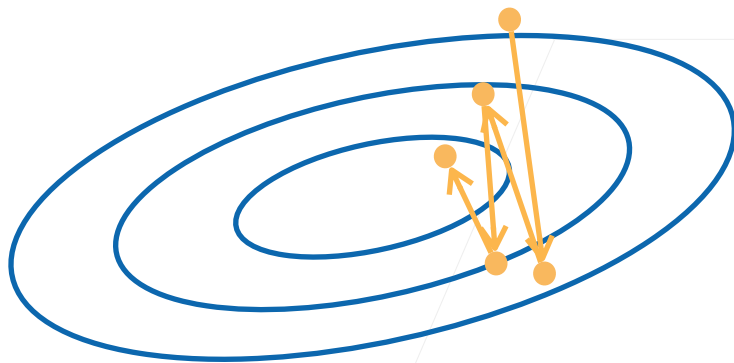
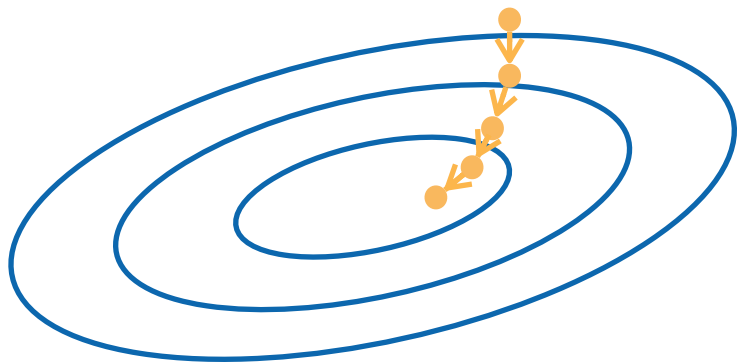


选择学习率

太慢导致速度慢 太快导致振荡
所以找不到最优解，反正就随
机应变吧！

不能太小

也不能太大





小批量随机梯度下降

- 在整个训练集上算梯度太贵
 - 一个深度神经网络模型可能需要数分钟至数小时
- 我们可以随机采样 b 个样本 i_1, i_2, \dots, i_b 来近似损失

$$\frac{1}{b} \sum_{i \in I_b} \ell(\mathbf{x}_i, y_i, \mathbf{w})$$

- b 是批量大小，另一个重要的超参数



选择批量大小

不能太小

每次计算量太小，不适合并行来最大利用计算资源

不能太大

内存消耗增加
浪费计算，例如如果所有样本都是相同的

总结



- 梯度下降通过不断沿着反梯度方向更新参数求解
- 小批量随机梯度下降是深度学习默认的求解算法
- 两个重要的超参数是批量大小和学习率