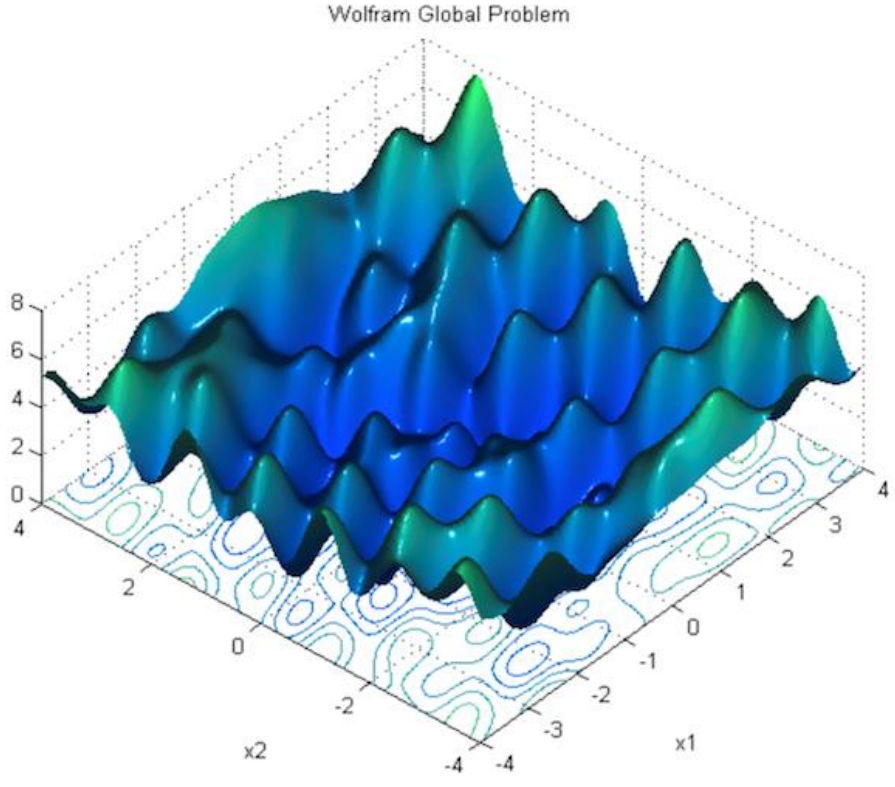
25 优化问题

概要

- 优化问题
  - 局部最小值和全局最小值
  - 凸集和凸函数
  - 凸优化证明
- 梯度下降
  - 学习率
  - 收敛率证明
  - 随机梯度下降
  - 小批量随机梯度下降

优化问题



优化问题

- 一般形式：

  $\text{minimize } f(\mathbf{x}), \text{ subject to } \mathbf{x} \in C$

  - 成本函数

    $$f: \mathbb{R}^n \to \mathbb{R}$$

  - 约束集的例子

    $$C = \{\mathbf{x} \mid h_1(\mathbf{x}) = 0, \ldots, h_m(\mathbf{x}) = 0, g_1(\mathbf{x}) \leq 0, \ldots, g_r(\mathbf{x}) \leq 0\}$$

  - 如果$C = \mathbb{R}^n$，则不受约束

局部最小值和全局最小值

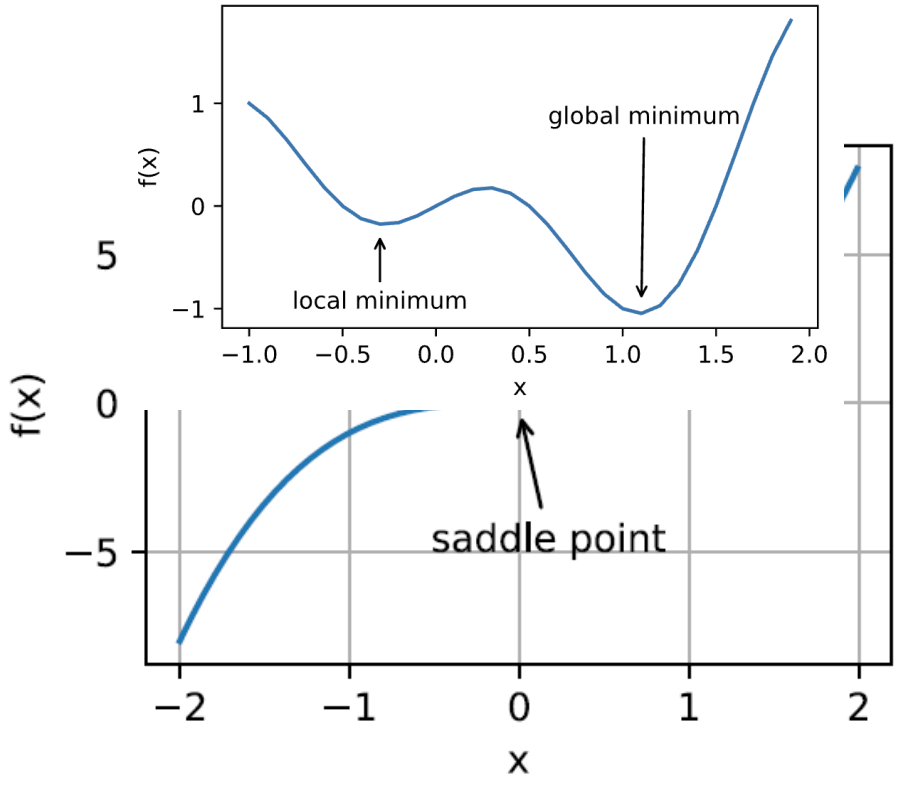
- 大多数优化问题都没有闭式(closed-form)解决方案【没有解析解】
- 我们的目标是通过迭代方法找到最小值
  - 全局最小值 $\mathbf{x}^*$：

    $$f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad \forall \mathbf{x} \in C$$

  - 局部最小值 $\mathbf{x}^*$，存在 ε：

    $$f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad \forall \mathbf{x}: \| \mathbf{x} - \mathbf{x}^* \| \leq \varepsilon$$

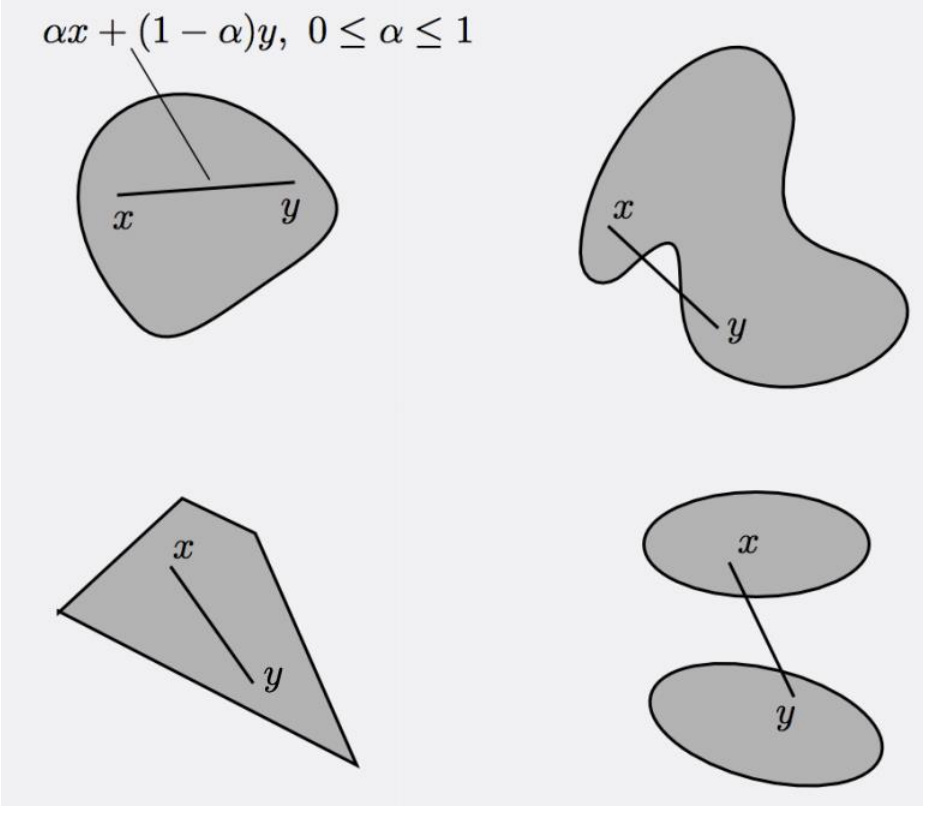
  - 鞍点(saddle point)是指函数的所有梯度都消失但既不是全局最小值也不是局部最小值的任何位置。



凸集

- 凸性(convexity)在优化算法的设计中起到至关重要的作用
  - 在这种情况下对算法进行分析和测试更容易
- $\mathbb{R}^n$的子集C为凸集(convex set)，如果满足

$$\alpha\mathbf{x} + (1-\alpha)\mathbf{y} \in C$$
$$\forall\alpha \in [0,1]\ \forall\mathbf{x},\mathbf{y} \in C$$

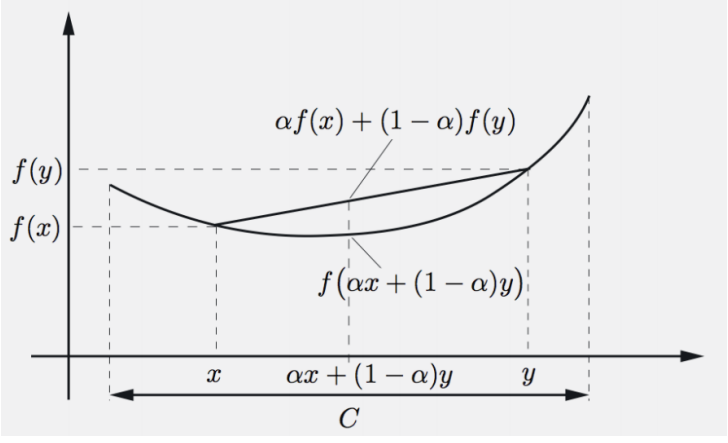


凸函数

- 函数$f: C \to \mathbb{R}$为凸函数(convex)，如果满足
  $f(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \leq \alpha f(\mathbf{x}) + (1-\alpha)f(\mathbf{y})$
  $\forall \alpha \in [0,1] \ \forall \mathbf{x}, \mathbf{y} \in C$
- 如果不等式在 $\alpha \in (0,1)$ 和 $\mathbf{x} \neq \mathbf{y}$ 条件下是严格的，那么f被称为严格凸
- 詹森不等式（Jensen‘ sinequality），它是凸性定义的一种推广

$$\sum_i \alpha_i f(x_i) \geq f\left(\sum_i \alpha_i x_i\right) \text{ and } E_X[f(X)] \geq f(E_X[X])$$

- 凸函数的期望不小于期望的凸函数



凸函数性质

- 局部极小值是全局极小值
- 凸函数的下水平集是凸的
- 一阶特征条件
  - f是凸函数，当且仅当

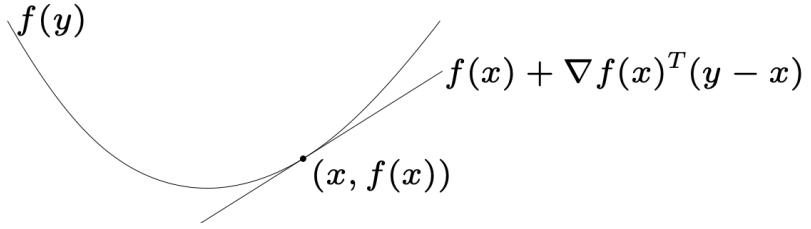
$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) \qquad \forall \mathbf{x}, \mathbf{y} \in C$$

  - 如果等号不成立，那么f是严格凸的
- 二阶特征条件
  - f是凸函数，当且仅当：

$$\nabla^2 f(\mathbf{x}) \geq 0 \qquad \forall \mathbf{x} \in C$$

  - f是严格凸的，当且仅当：

$$\nabla^2 f(\mathbf{x}) > 0 \qquad \forall \mathbf{x} \in C$$

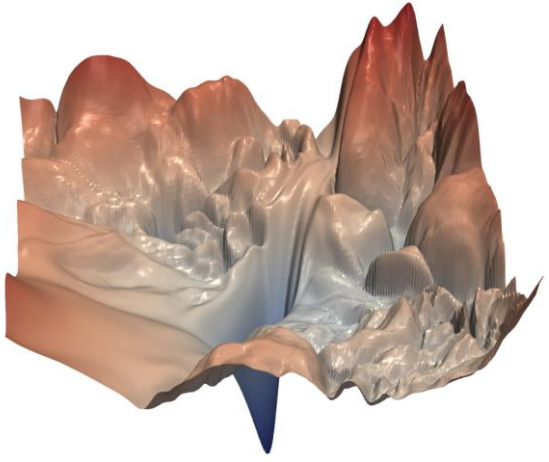


常见凸集和非凸集

- 凸集
  - 线性回归 $f(\mathbf{x}) = \| \mathbf{W}\mathbf{x} - \mathbf{b} \|_2^2$

$$\nabla f(\mathbf{x}) = 2\mathbf{W}^T(\mathbf{W}\mathbf{x} - \mathbf{b}), \nabla^2 f(\mathbf{x}) = 2\mathbf{W}^T$$

  - Softmax 回归
- 非凸集
  - 多层感知器
  - 卷积神经网络
  - 循环神经网络



凸优化

- 如果f是凸函数，而且C是凸集，则该问题称为凸问题：
  - 任何局部最小值都是全局最小值
  - 如果严格凸成立，则为唯一的全局最小值

凸优化证明

- 局部最小值为 $\mathbf{x}$，假设存在全局最小值 $\mathbf{y}$：
  - 选择一个 $\alpha \leq 1 - \varepsilon/|\mathbf{x} + \mathbf{y}|$ 和一个 $\mathbf{z} = \alpha\mathbf{x} + (1 - \alpha)\mathbf{y}$
  - 然后 $\| \mathbf{x} - \mathbf{z} \| = (1 - \alpha) \| \mathbf{x} + \mathbf{y} \| \leq \varepsilon$
  - 由于 $\mathbf{y}$ 是全局最小值，所以 $f(\mathbf{y}) < f(\mathbf{x})$

    $f(\mathbf{z}) \leq \alpha f(\mathbf{x}) + (1 - a) f(\mathbf{z}) < \alpha f(\mathbf{x}) + (1 - a) f(\mathbf{x}) = f(\mathbf{x})$
  - 它与局部最小值 $\mathbf{x}$ 相矛盾

凸函数

- 凸集的交点是凸的，并集不是
- 根据詹森不等式，“一个多变量凸函数的总期望值”大于或等于“用每个变量的期望值计算这个函数的总值“
- 一个二次可微函数是凸函数，当且仅当其Hessian（二阶导数矩阵）是半正定的
- 凸约束可以通过拉格朗日函数来添加。在实践中，只需在目标函数中加上一个惩罚就可以了
- 投影映射到凸集中最接近原始点的点

梯度下降


SOMMET
3 VALLÉ
3 230m
COL DE
ROSAEL
3000 m
COL DE
THORENS
3002 m
GLACIER DU
BOUCHET
La tyrOlienne
white wild
Bouchet
BOUCHET
Lory
ROSAËL
Mauriennaise
Gentianes
La croix d'Antide
PEYRON
Peyron
PLAN BOUCHET
2350 m
Chalet Chinal Donat
TELECORDE
3 VALLEES EXPRE

一维梯度下降

连续可微实值函 数 $f: \mathbb{R} \to \mathbb{R}$, 利用泰勒展开, 可以得到

$$f(x+\epsilon) = f(x) + \epsilon f'(x) + \mathcal{O}(\epsilon^2).$$

假设在负梯度方向上移动ϵ会减少f。简单起见，选择固定步长 $\eta > 0$ ，取 $\epsilon = -\eta f'(x)$ 将其代入泰勒展开式，得

$$f(x - \eta f'(x)) = f(x) - \eta f'^2(x) + \mathcal{O}(\eta^2 f'^2(x)).$$

如果其导数 $f'(x) \neq 0$ 没有消失，我们就能继续展开，这是因为 $\eta f'^2(x) > 0$ 。此外，我们总是可以令η小到足以使高阶项变得不相关。因此,

$$f(x - \eta f'(x)) \lesssim f(x)$$

这意味着，如果我们使用$x \leftarrow x - \eta f'(x)$来迭代 x, 函数 $f(x)$ 的值可能会下降。

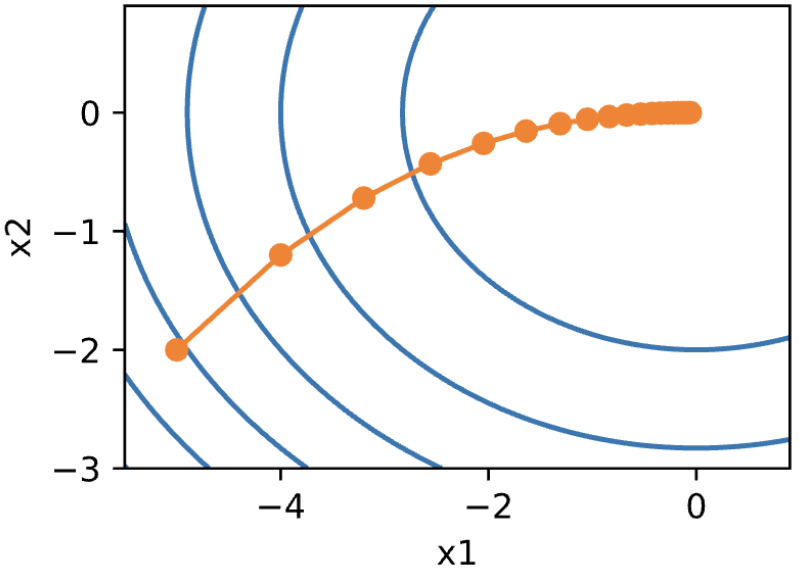
因此, 在梯度下降中, 我们首先选择初始值 x 和常数 $\eta > 0$, 然后使用它 们连续迭代 x, 直到停止条件达成

算法

- 选择初始的$\mathbf{x}_0$
- 在时间 $t = 1, \ldots, T$

$$\mathbf{x}_t = \mathbf{x}_{t-1} - \eta \nabla f(\mathbf{x}_{t-1})$$

  - η 被称为学习率



学习率的选择

- 在 $\| \boldsymbol{\Delta} \| < \varepsilon$ 的条件下，对于任何 f，由泰勒扩展：

$$f(\mathbf{x} + \boldsymbol{\Delta}) \approx f(\mathbf{x}) + \boldsymbol{\Delta}^T \nabla f(\mathbf{x})$$

- 选择足够小的学习率 $\eta \leq \frac{\varepsilon}{\| \nabla f(\mathbf{x}) \|}$

$$\| -\eta \nabla f(\mathbf{x}) \| \leq \varepsilon$$

$$f(\mathbf{x} - \eta \nabla f(\mathbf{x})) \approx f(\mathbf{x}) - \eta \| \nabla f(\mathbf{x}) \|^2 \leq f(\mathbf{x})$$

收敛率

- 假设f是凸的，并且其梯度是 Lipschitz 连续的常数L

$$\| \nabla f(\mathbf{x}) - \nabla f(\mathbf{y}) \| \leq L \| \mathbf{x} - \mathbf{y} \|$$

渐变不会发生显着变化

- 如果使用学习率 $\eta \leq 1/L$，经过 T 步

$$f(\mathbf{x}_T) - f(\mathbf{x}^*) \leq \frac{\| \mathbf{x}_0 - \mathbf{x}^* \|^2}{2\eta T}$$

  - 收敛率 $O(1/T)$
  - 要获得 $f(\mathbf{x}_T) - f(\mathbf{x}^*) \leq \epsilon$，需要 $O(1/\epsilon)$ 次迭代

随机梯度下降(SGD)

1000 新加坡元 (SGD)
~740 USD

随机梯度下降

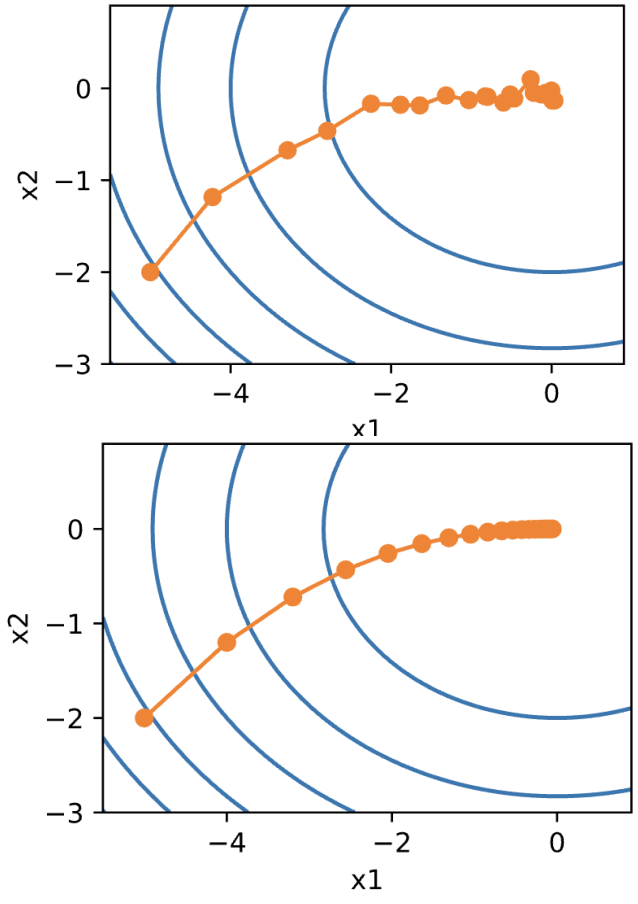
- 给定 n 个样本的训练数据集, 假设 $f_i(\mathbf{x})$ 是关于索引 i 的训练样本的损失函数，其中$\mathbf{x}$是参数向量
- 目标函数$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})$，$\mathbf{x}$的目标函数的梯度计算为$\nabla f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\mathbf{x})$
  - 如果使用梯度下降法，则每个自变量迭代的计算代价为$\mathcal{O}(n)$
- 在随机梯度下降的每次迭代中，对数据样本随机均匀采样一个索引i,其中$i \in \{1, \ldots, n\}$,并计算梯度$\nabla f_i(\mathbf{x})$以更新$\mathbf{x}$:

$$\mathbf{x} \leftarrow \mathbf{x} - \eta \nabla f_i(\mathbf{x})$$

  - 其中η是学习率。每次迭代的计算代价从梯度下降的$\mathcal{O}(n)$降至常数$\mathcal{O}(1)$
  - 随机梯度 $\nabla f_i(\mathbf{x})$ 是对完整梯度 $\nabla f(\mathbf{x})$ 的无偏估计，因为
    - $\mathbb{E}_i \nabla f_i(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\mathbf{x}) = \nabla f(\mathbf{x})$

随机梯度下降

- 随机梯度下降中变量的轨迹嘈杂。这是由于梯度的随机性质
  - 也就是说，即使我们接近最小值，我们仍然受到通过∇fi(x)的瞬间梯度所注入的不确定性的影响
- 改变学习率
  - 如果选择的学习率太小，我们一开始就不会取得任何有意义的进展。
  - 如果选择的学习率太大，将无法获得一个好的解决方案
- 解决这些相互冲突的目标的唯一方法是在优化过程中动态降低学习率



例子

- 在时间 t 举例的两个规则
  - 随机(random)规则：随机均匀选择 $i_t \in \{1, \dots, n\}$
  - 循环(cyclic)规则：选择 $i_t = 1,2, \dots, n, 1,2, \dots, n$
    - 通常称为增量梯度下降
- 随机规则在实践中更为常见

$$\mathbb{E}\left[\nabla \ell_{t_i}(\mathbf{x})\right] = \mathbb{E}[\nabla f(\mathbf{x})]$$

  - 对梯度的无偏见估计

收敛率

- 假设f是凸的，并且 η_t 逐步减小，例如 $\eta_t = O(1/t)$，

$$\mathbb{E}[f(\mathbf{x}_T)] - f(\mathbf{x}^*) = O(1/\sqrt{T})$$

- 在相同的假设下，对于梯度下降

$$f(\mathbf{x}_T) - f(\mathbf{x}^*) = O(1/\sqrt{T})$$

- 假设梯度 L-Lipschitz 并固定 η

$$f(\mathbf{x}_T) - f(\mathbf{x}^*) = O(1/T)$$

  - 对 SGD 没有改善

实际应用

- 我们不会如此戏剧性地降低学习率
  - 我们不关心优化到高精度
- 尽管收敛速度较慢，但在每次迭代中，SGD 计算梯度的速度要快于 GD
  - 特别适用于复杂模型和大型数据集的深度学习

动态学习率

- 随着时间推移调 整 η 时使用的一些基本策略

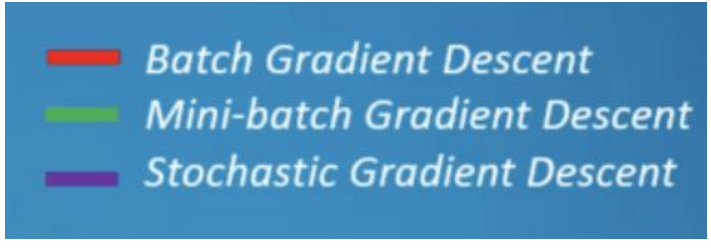
$$\eta(t) = \eta_i \text{ if } t_i \leq t \leq t_{i+1} \quad \text{分段常数}$$

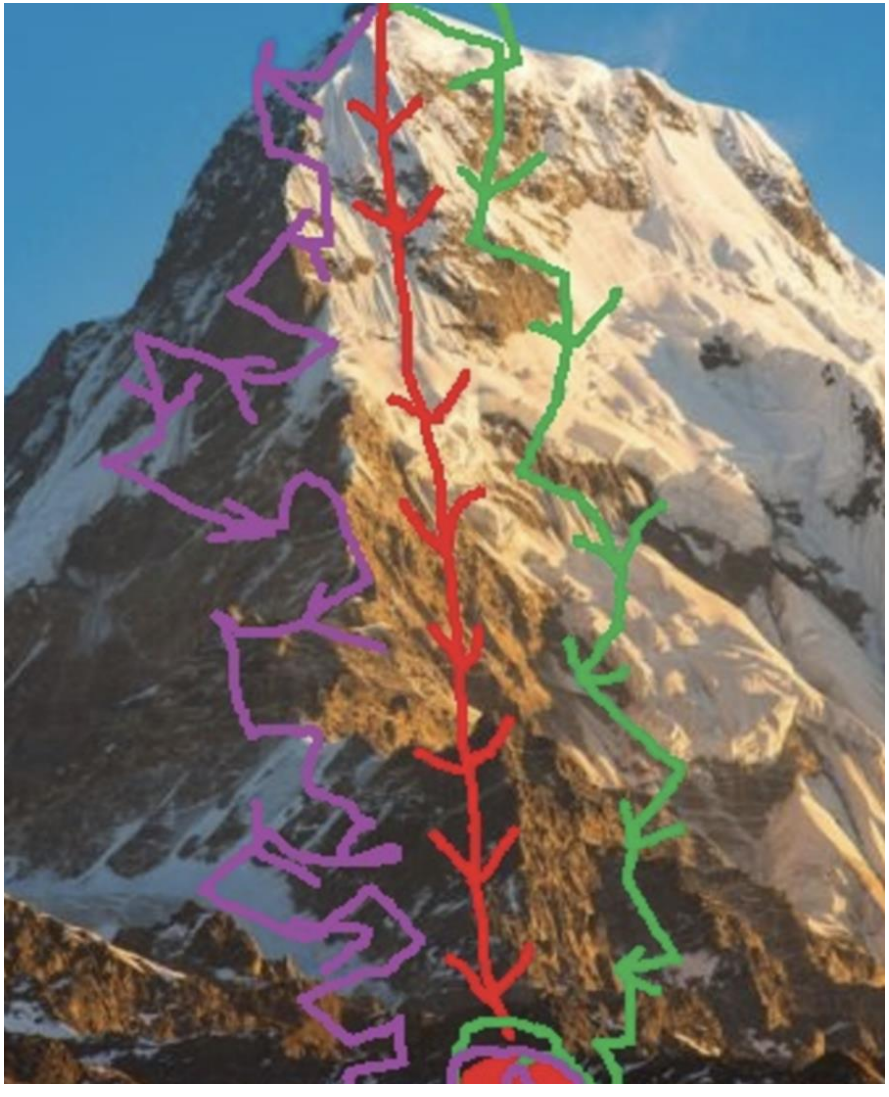
$$\eta(t) = \eta_0 \cdot e^{-\lambda t} \quad \text{指数衰减}$$

$$\eta(t) = \eta_0 \cdot (\beta t + 1)^{-\alpha} \quad \text{多项式衰减}$$

代码...

小批量随机梯度下降
(mini-batch SGD)





小批量随机梯度下降

- 每当我们执 行 $\mathbf{W} \leftarrow \mathbf{W} - \eta_t \mathbf{g}_t$ 时, 消耗巨大。其中

$$\mathbf{g}_t = \partial_{\mathbf{w}} f(\mathbf{x}_t, \mathbf{w})$$

- 可以通过将其应用于一个小批量观测值来提高此操作的计算效率。也就是说，我们将梯度 $\mathbf{g}_t$ 替换为一个 小批量而不是单个观测值

$$\mathbf{g}_t = \partial_{\mathbf{w}} \frac{1}{|\mathcal{B}_t|} \sum_{i \in \mathcal{B}_t} f(\mathbf{x}_i, \mathbf{w})$$

  - 由于 $\mathbf{x}_t$ 和小批量 $\mathcal{B}_t$ 的所有元素都是从训练集中随机抽出的，因此 梯度的期望保持不变。
  - 另一方面, 方差显著降低。由于小批量梯度由正在被平均计算的 $b := |\mathcal{B}_t|$ 个独立梯度组成，其标准差降低了 $b^{-\frac{1}{2}}$
  - 实践中选择一个足够大的小批量, 它可提供良好计算效率同时仍适合GPU的内存

代码...

动量法

动量法

- 使用 $\mathbf{v}_t$ 而 不是梯度 $\mathbf{g}_t$ 可以生成以下更新等式:

$$\mathbf{v}_t \leftarrow \beta \mathbf{v}_{t-1} + \mathbf{g}_{t,t-1}$$
$$\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} - \eta_t \mathbf{v}_t$$

- 汇总过去梯度的历史以加速收敛

Adam算法

Adam算法

- 我们学习了
  - 随机梯度下降在解决优化问题时比梯度下降更有效。
  - 在一个小批量中使用更大的观测值集，可以通过向量化提供额外效率。这是高效的多机、多GPU和整体并行处理的关键。
  - 添加了一种机制，用于汇总过去梯度的历史以加速收敛。
  - 通过对每个坐标缩放来实现高效计算的预处理器。
  - 通过学习率的调整来分离每个坐标的缩放
- Adam算法[Kingma & Ba, 2014]将所有这些技术汇总到一个高效的学习算法中。

总结

- 优化问题
  - 局部最小值和全局最小值
  - 凸集和凸函数
  - 凸优化证明
- 梯度下降
  - 学习率
  - 收敛率证明
  - 随机梯度下降
  - 小批量随机梯度下降

附录

收敛率证明

- 渐变 L-Lipschitz 意味着

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \frac{L}{2}\| \mathbf{y}-\mathbf{x} \|^2$$

- 插入 $\mathbf{y} = \mathbf{x} - \eta\nabla f(\mathbf{x})$

$$f(\mathbf{y}) \leq f(\mathbf{x}) - \left(1 - \frac{L\eta}{2}\right)\eta \| \nabla f(\mathbf{x}) \|^2$$

- 采用 $0 < \eta \leq \frac{1}{L}$

$$f(\mathbf{y}) \leq f(\mathbf{x}) - \frac{\eta}{2}\| \nabla f(\mathbf{x}) \|^2$$

f每次都减小

收敛率证明 II

- 根据凸性

$$f(\mathbf{x}) \leq f(\mathbf{x}^*) + \nabla f(\mathbf{x})^T(\mathbf{x} - \mathbf{x}^*)$$

- 插入

$$f(\mathbf{y}) \leq f(\mathbf{x}) - \frac{\eta}{2} \| \nabla f(\mathbf{x}) \|^2$$

$$f(\mathbf{y}) \leq f(\mathbf{x}^*) + \nabla f(\mathbf{x})^T(\mathbf{x} - \mathbf{x}^*) - \frac{\eta}{2} \| \nabla f(\mathbf{x}) \|^2$$

$$f(\mathbf{y}) - f(\mathbf{x}^*) \leq \frac{(2\eta \nabla f(\mathbf{x})^T(\mathbf{x} - \mathbf{x}^*) - \eta^2 \| \nabla f(\mathbf{x}) \|^2)}{2\eta}$$

$$= \frac{(\| \mathbf{x} - \mathbf{x}^* \|^2 + 2\eta \nabla f(\mathbf{x})^T(\mathbf{x} - \mathbf{x}^*) - \eta^2 \| \nabla f(\mathbf{x}) \|^2 - \| \mathbf{x} - \mathbf{x}^* \|^2)}{2\eta}$$

$$= \frac{(\| \mathbf{x} - \mathbf{x}^* \|^2 - \| \mathbf{x} - \eta \nabla f(\mathbf{x}) - \mathbf{x}^* \|^2)}{2\eta}$$

$$= (\| \mathbf{x} - \mathbf{x}^* \|^2 - \| \mathbf{y} - \mathbf{x}^* \|^2)/2\eta$$

收敛率证明 III

- 所有 T 步骤总和：

$$\sum_{t=1}^{T} f(\mathbf{x}_t) - f(\mathbf{x}^*) \leq \sum_{t=1}^{T} \frac{(\| \mathbf{x}_{t-1} - \mathbf{x}^* \|^2 - \| \mathbf{x}_t - \mathbf{x}^* \|^2)}{2\eta}$$
$$= (\| \mathbf{x}_0 - \mathbf{x}^* \|^2 - \| \mathbf{x}_T - \mathbf{x}^* \|^2)/2\eta \leq \| \mathbf{x}_0 - \mathbf{x}^* \|^2/2\eta$$

- f 每次都在减少：

$$f(\mathbf{x}_T) - f(\mathbf{x}^*) \leq \frac{1}{T}\sum_{t=1}^{T} f(\mathbf{x}_t) - f(\mathbf{x}^*) \leq \frac{\| \mathbf{x}_0 - \mathbf{x}^* \|^2}{2\eta T}$$

深度学习应用

- f是所有训练数据的损失之和的函数， $\mathbf{x}$是可学习的参数

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=0}^{n} \ell_i(\mathbf{x})$$

$\ell_i(\mathbf{x})$ 是第i个样本的损失

- f往往不是凸函数，所以不能应用之前的收敛性分析