优化算法

动手学深度学习 v2

李沐・AWS

优化问题

- 一般形式

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

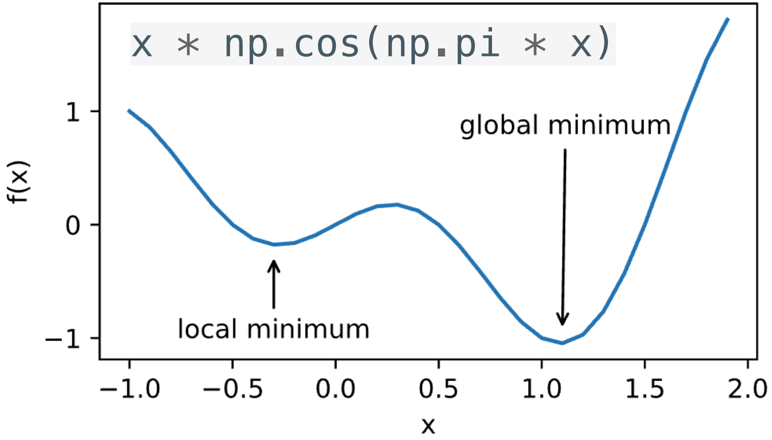
  - 目标函数 $f: \mathbb{R}^n \to \mathbb{R}$
  - 限制集合例子

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

  - 如果 $C = \mathbb{R}^n$ 那就是不受限

局部最小 vs 全局最小

- 全局最小 $\mathbf{x}^*$： $f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad \forall \mathbf{x} \in C$
- 局部最小 $\mathbf{x}^*$： 存在 ε， 使得 $f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad \forall \mathbf{x} : \|\mathbf{x} - \mathbf{x}^*\| \leq \varepsilon$
- 使用迭代优化算法来求解，一般只能保证找到局部最小值

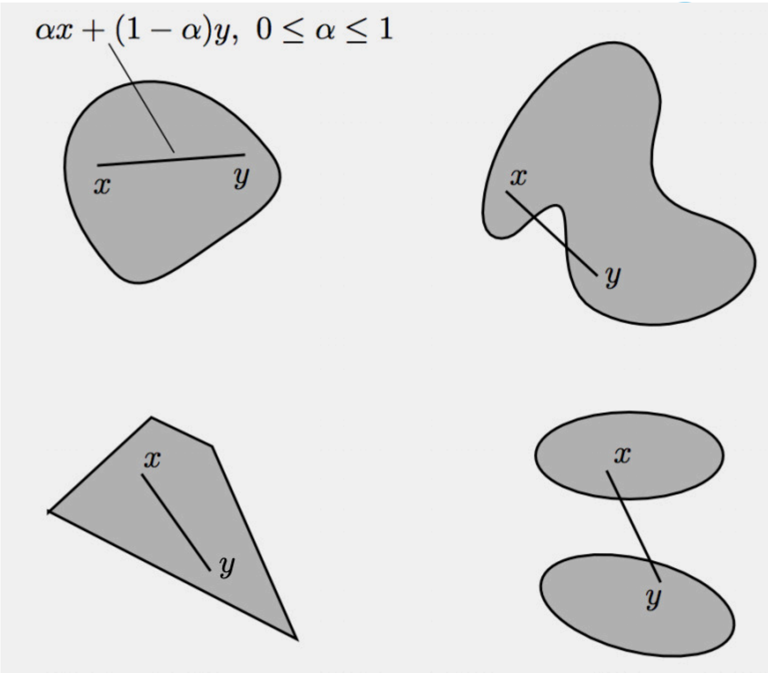


凸集

- 一个 $\mathbb{R}^n$ 的子集 C 是凸当且仅当

$$\alpha \mathbf{x} + (1 - \alpha)\mathbf{y} \in C$$

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



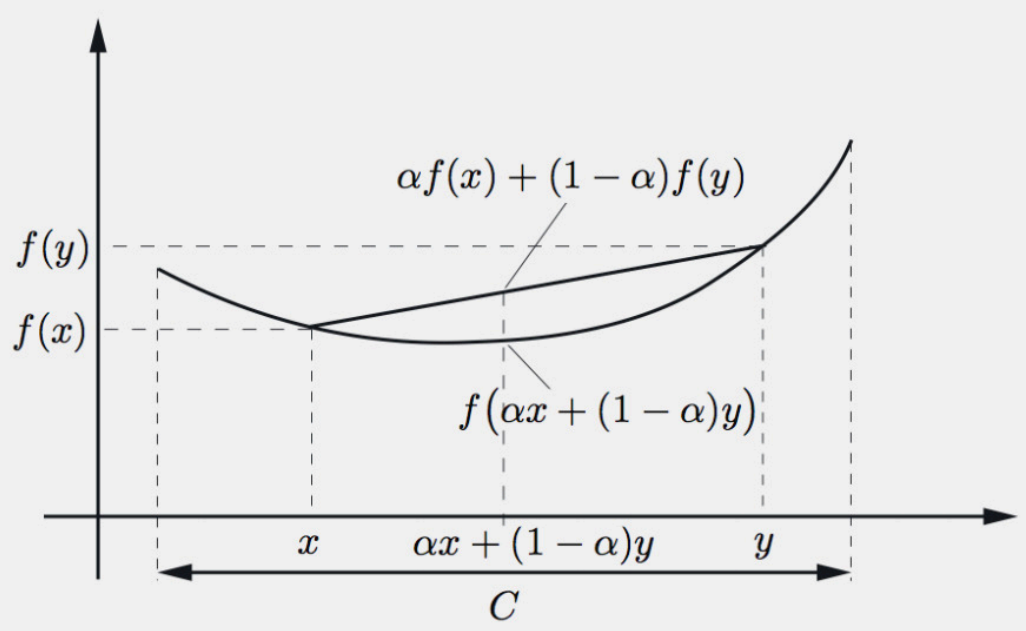
凸函数

- 函数 $f: C \to \mathbb{R}$ 是凸当且仅当

$$f(\alpha \mathbf{x} + (1-\alpha)\mathbf{y}) \leq \alpha f(\mathbf{x}) + (1-\alpha) f(\mathbf{y})$$

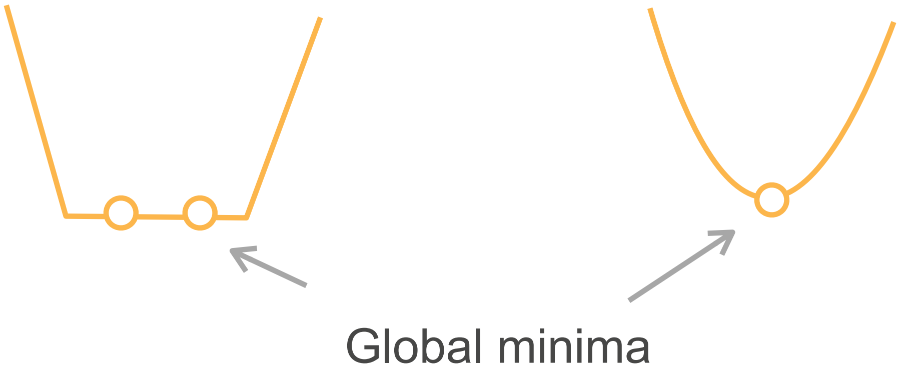
$$\forall \alpha \in [0,1] \quad \forall \mathbf{x}, \mathbf{y} \in C$$

- 如果$\mathbf{x} \neq \mathbf{y}, \alpha \in (0,1)$时不等式严格成立，那么叫严格凸函数



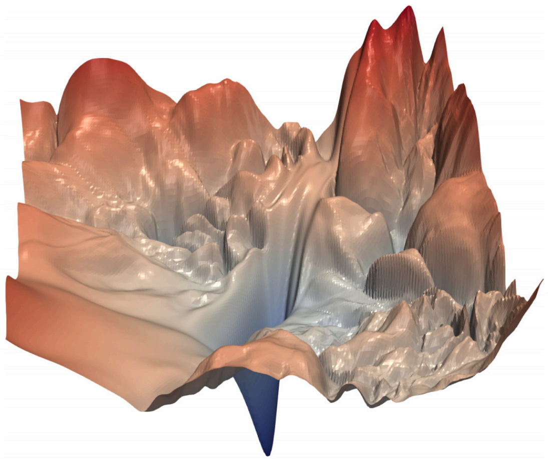
凸函数优化

- 如果代价函数 f 是凸的，且限制集合 C 是凸的，那么就是凸优化问题，那么局部最小一定是全局最小
- 严格凸优化问题有唯一的全局最小



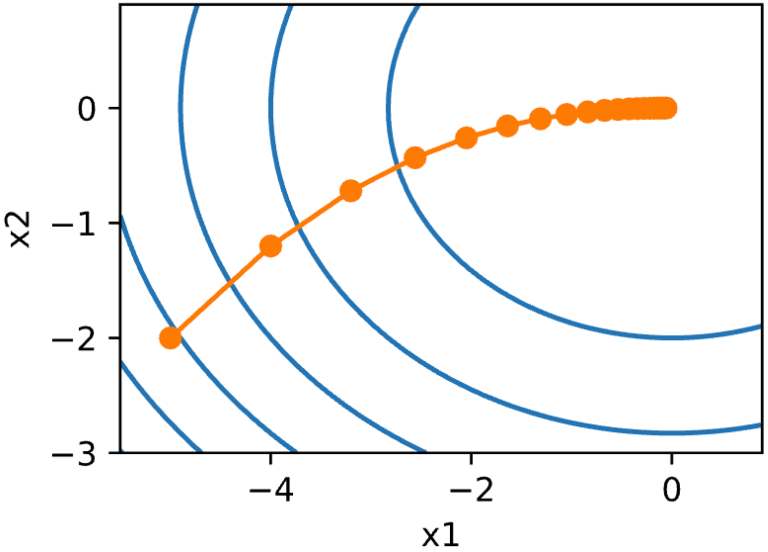
凸和非凸例子

- 凸
  - 线性回归 $f(\mathbf{x}) = \|\mathbf{W}\mathbf{x} - \mathbf{b}\|_2^2$
  - Softmax 回归
- 非凸：其他
  - MLP, CNN, RNN, attention, …



梯度下降

- 最简单的迭代求解算法
- 选取开始点 $\mathbf{x}_0$
- 对 $t = 1, \ldots, T$
  - $\mathbf{x}_t = \mathbf{x}_{t-1} - \eta \nabla f(\mathbf{x}_{t-1})$
- η 叫做学习率

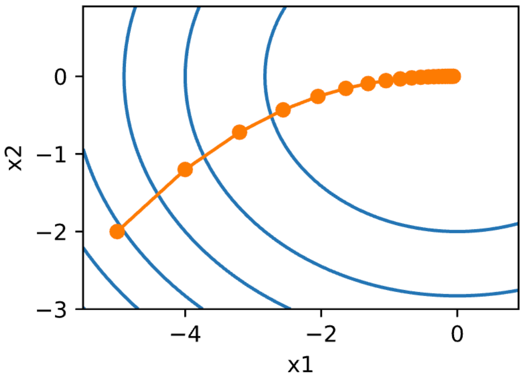


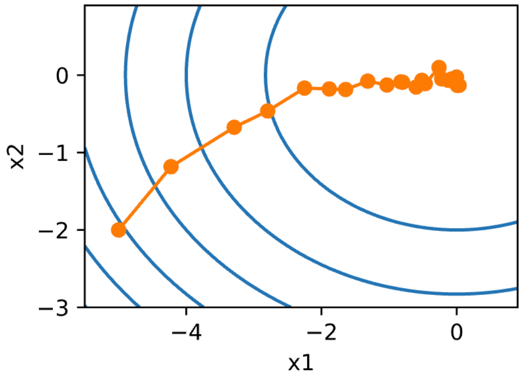
随机梯度下降

- 有n个样本时，计算 $f(\mathbf{x}) = \frac{1}{n}\sum_{i=0}^{n} \ell_i(\mathbf{x})$ 的导数太贵
- 随机梯度下降在时间 t 随机选项样本 t_i 来近似 $f(x)$

$$\mathbf{x}_t = \mathbf{x}_{t-1} - \eta_t \nabla \ell_{t_i}(\mathbf{x}_{t-1})$$

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





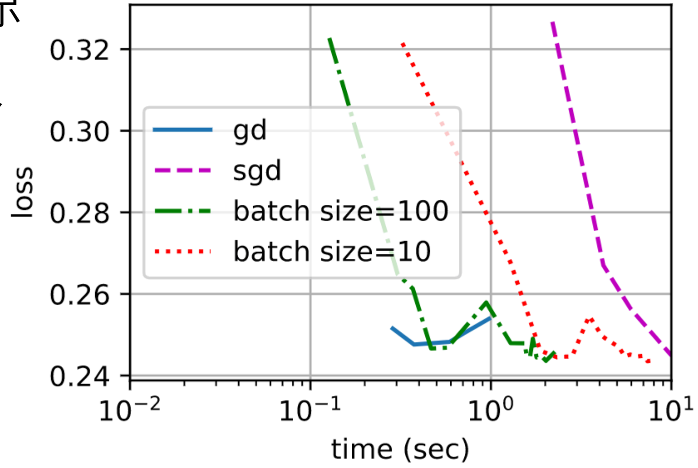
小批量随机梯度下降

- 计算单样本的梯度难完全利用硬件资源
- 小批量随机梯度下降在时间 t 采样一个随机子集 $I_t \subset \{1,\ldots,n\}$ 使得 $|I_t| = b$

$$\mathbf{x}_t = \mathbf{x}_{t-1} - \frac{\eta_t}{b} \sum_{i \in I_t} \nabla \ell_i(\mathbf{x}_{t-1})$$

- 同样，这是一个无偏的近似，但降低了方差

$$\mathbb{E}\left[\frac{1}{b} \sum_{i \in I_t} \nabla \ell_i(\mathbf{x})\right] = \nabla f(\mathbf{x})$$



冲量法

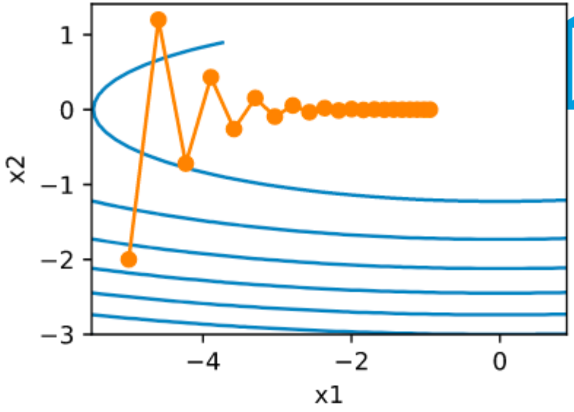
- 冲量法使用平滑过的梯度对权重更新

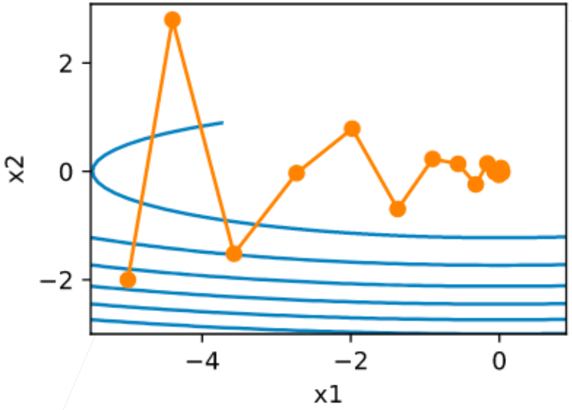
$$\mathbf{g}_t = \frac{1}{b}\sum_{i\in I_t}\nabla \ell_i(\mathbf{x}_{t-1})$$

$$\mathbf{v}_t = \beta\mathbf{v}_{t-1} + \mathbf{g}_t \qquad \mathbf{w}_t = \mathbf{w}_{t-1} - \eta\mathbf{v}_t$$

梯度平滑： $\mathbf{v}_t = \mathbf{g}_t + \beta\mathbf{g}_{t-1} + \beta^2\mathbf{g}_{t-2} + \beta^3\mathbf{g}_{t-3} + \ldots$

- β 常见取值 [0.5, 0.9, 0.95, 0.99]





Adam

- 记录 $\mathbf{v}_t = \beta_1 \mathbf{v}_{t-1} + (1-\beta_1)\mathbf{g}_t$ 通常 $\beta_1 = 0.9$
- 展开 $\mathbf{v}_t = (1-\beta_1)\left(\mathbf{g}_t + \beta_1 \mathbf{g}_{t-1} + \beta_1^2 \mathbf{g}_{t-2} + \beta_1^3 \mathbf{g}_{t-3} + \ldots\right)$
- 因为 $\sum_{i=0}^{\infty} \beta_1^t = \frac{1}{1-\beta_1}$，所以权重和为1
- 由于 $\mathbf{v}_0 = 0$，且 $\sum_{i=0}^{t} \beta_1^t = \frac{1-\beta_1^t}{1-\beta_1}$，

  修正 $\hat{\mathbf{v}}_t = \frac{\mathbf{v}_t}{1-\beta_1^t}$

Adam

- 类似记录 $\mathbf{s}_t = \beta_2 \mathbf{s}_{t-1} + (1-\beta_2)\mathbf{g}_t^2$，通常$\beta_2 = 0.999$，且修正 $\hat{\mathbf{s}}_t = \dfrac{\mathbf{s}_t}{1-\beta_2^t}$
- 计算重新调整后的梯度 $\mathbf{g}_t' = \dfrac{\hat{\mathbf{v}}_t}{\sqrt{\hat{\mathbf{s}}_t} + \epsilon}$
- 最后更新 $\mathbf{w}_t = \mathbf{w}_{t-1} - \eta \mathbf{g}_t'$

总结

- 深度学习模型大多是非凸
- 小批量随机梯度下降是最常用的优化算法
- 冲量对梯度做平滑
- Adam对梯度做平滑，且对梯度各个维度值做重新调整