优化算法

优化算法是解决优化问题的一类算法，在深度学习中一般是帮助模型找到寻找最佳参数的算法。

优化问题

一般形式为：

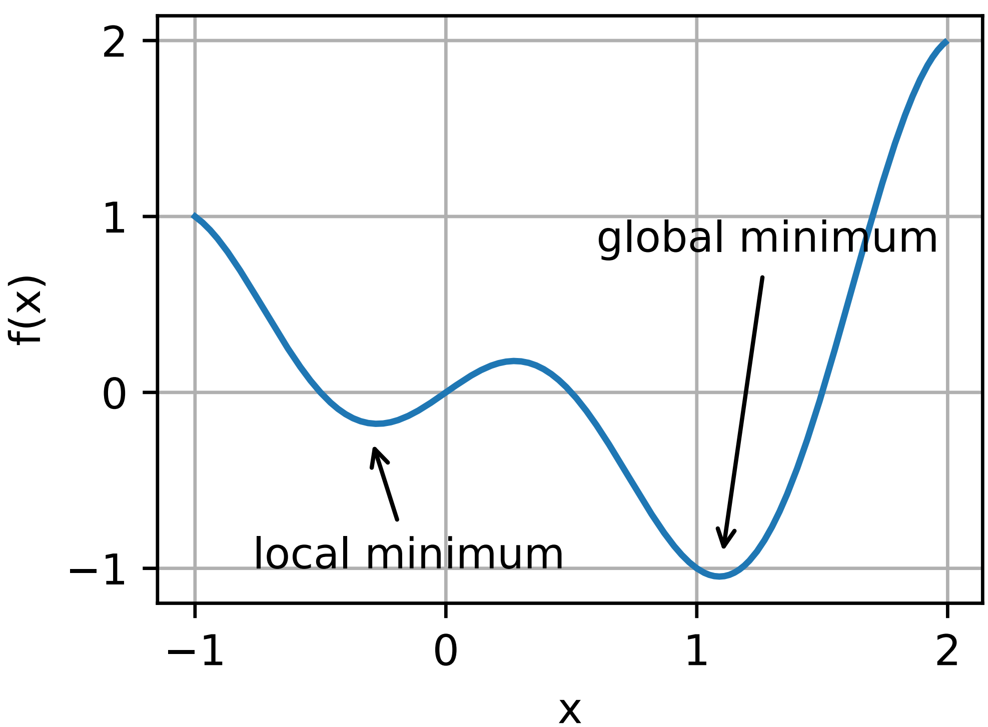
$$\text{minimize } f(x), \text{subject to } x \in C \tag{1}$$

其中目标函数$f(x): \mathbb{R}^n \to \mathbb{R}$，$C$为限制集合。

优化问题的最小值分为两种：局部最小和全局最小。

全局最小（定义域内的最小值）： $x^*: \forall x \in C, f(x^*) \le f(x)$

局部最小（一定范围内的最小值）： $x^*: \exists \epsilon,$ 使得$\forall x: \|x - x^*\| \le \epsilon, f(x^*) \le f(x)$

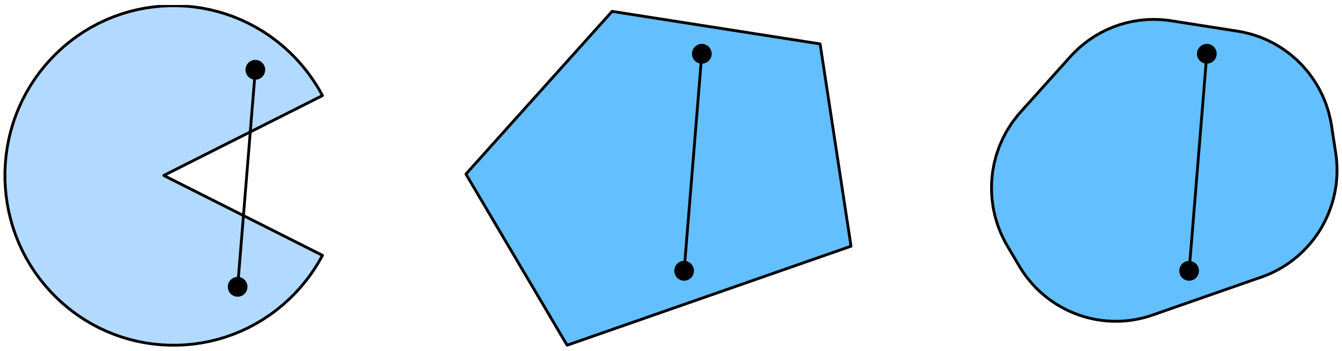


凸性

凸集

在集合上任取两个点连成一条线段，这个线段上所有的点都在集合内。数学表达为：

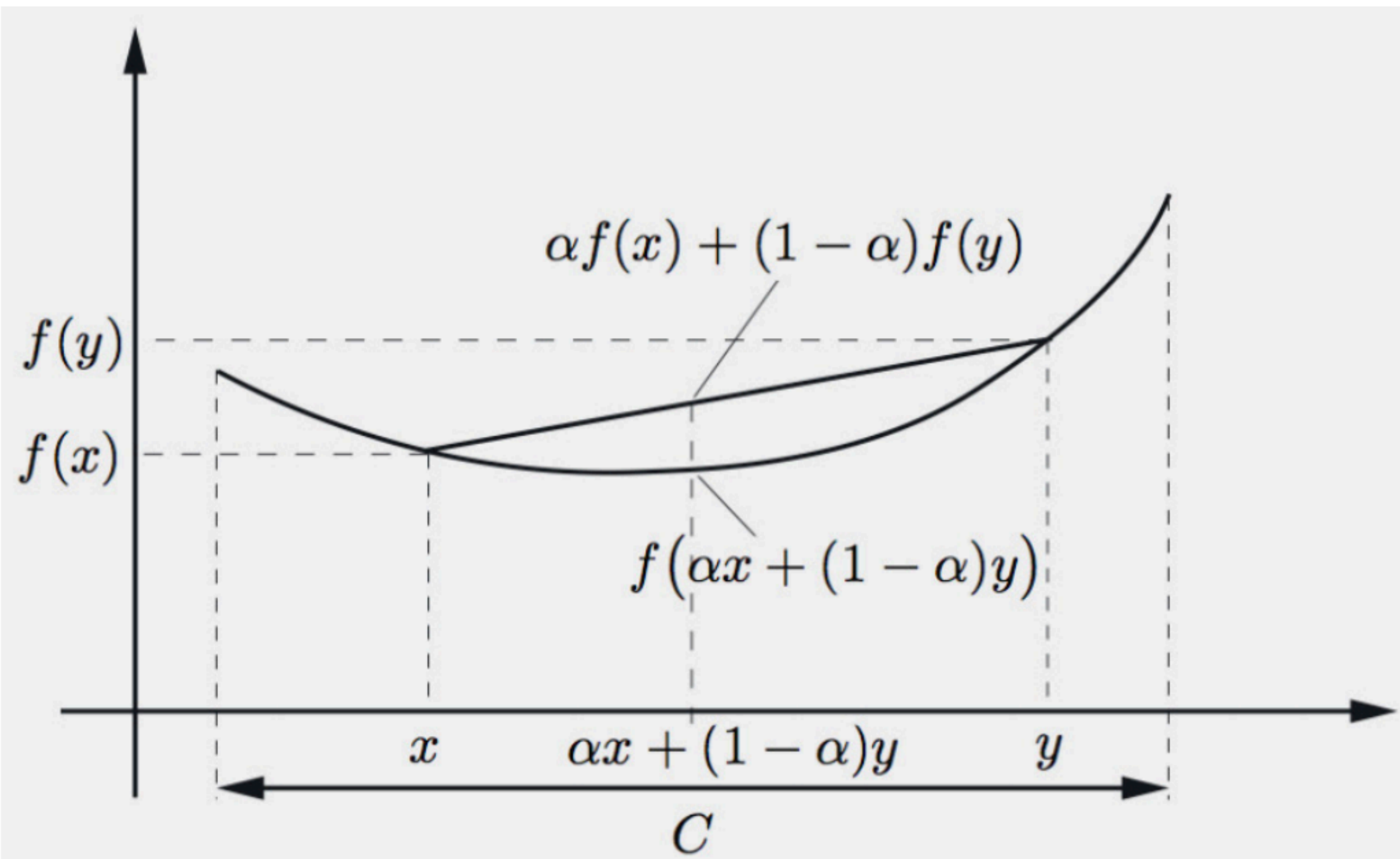
$$\lambda a + (1-\lambda) b \in \mathcal{X} \text{ 当 } a, b \in \mathcal{X}. \tag{2}$$



凸函数

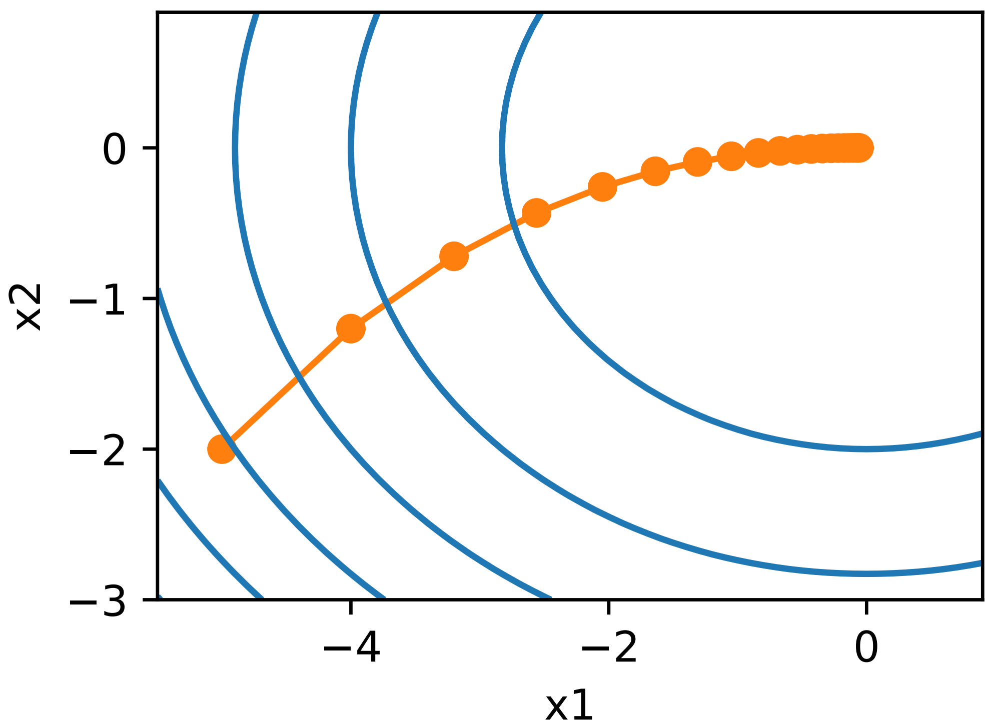
在函数上任取两个点连成一条线，这两个点之间所有的函数值都在这条线之下。数学表达式为：

$$\lambda f(x) + (1-\lambda) f(y) \geq f(\lambda x + (1-\lambda) y). \tag{3}$$



梯度下降

- 选取开始点x_0
- 对$t = 1, 2, \ldots, T$，$x_t = x_{t-1} - \eta \nabla f(x_{t-1})$，$\eta$为学习率



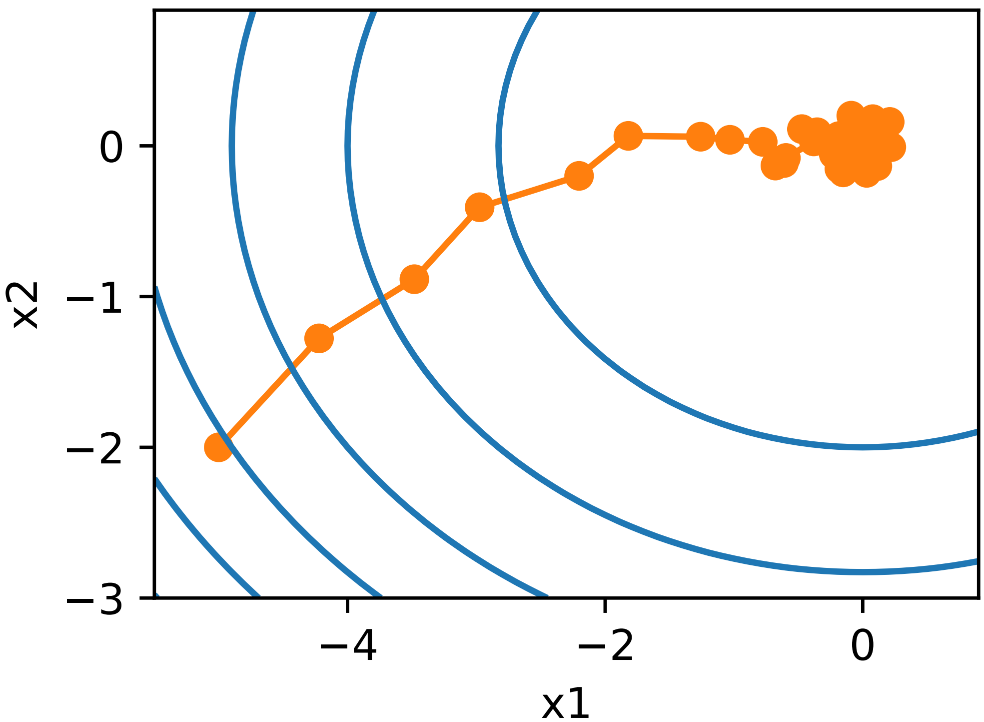
随机梯度下降

由于同时计算所有样本的平均梯度太贵，可以使用单个样本的梯度来近似所有样本的平均梯度，这是由于：

$$\mathbb{E}[\nabla \ell_i(x)] = \mathbb{E}[\nabla f(x)] \tag{4}$$

更新方式为：

$$x_t = x_{t-1} - \eta_t \nabla \ell_{t_i}(x_{t-1}) \tag{5}$$



小批量随机梯度下降

小批量随机梯度下降是在原数据集中选择一个小的子集，计算这个子集的平均梯度并以此来代替全集的梯度。与随机梯度下降一样，小批量随机梯度下降方法也是一个对于全集梯度的无偏近似，但它拥有更小的方差。

更新方式为：

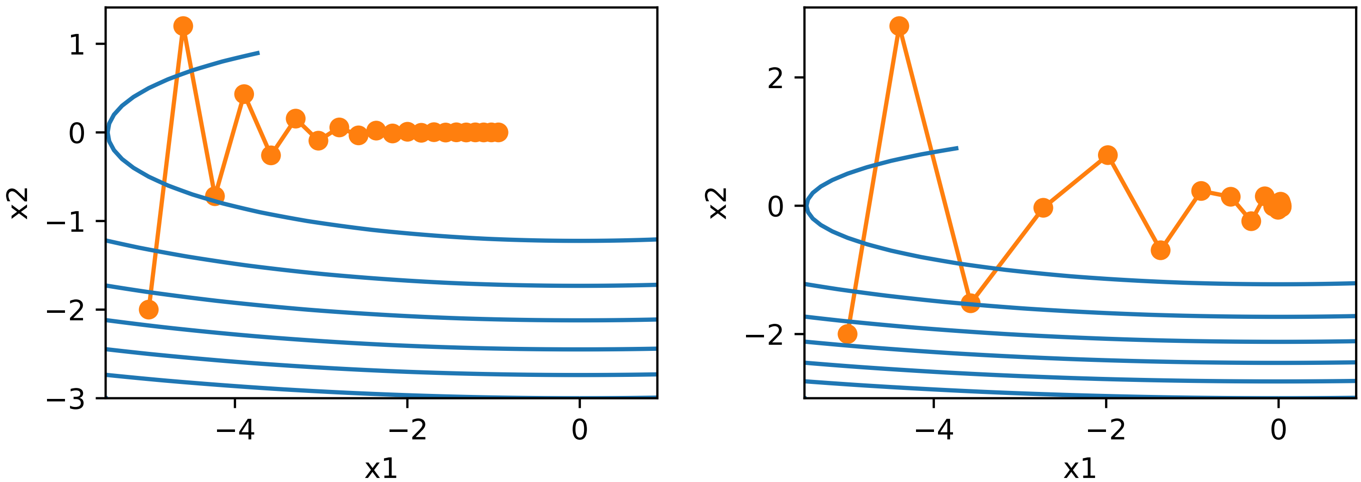
- 在t时刻采集一个随机子集$I_t \subset \{1, \ldots, n\}$使得$|I_t| = b$
- $x_t = x_{t-1} - \frac{\eta_t}{b} \sum_{i \in I_t} \nabla \ell_i(x_{t-1})$

冲量法

梯度下降方法可能会造成梯度震荡的情况，为了应对这种情况，可以引入冲量的概念对梯度进行平滑。

$$
\begin{aligned}
g_t &= \frac{1}{b} \sum_{i \in I_t} \nabla \ell_i(x_{t-1}) \\
V_t &= \beta V_{t-1} + g_t \\
W_t &= W_{t-1} - \eta V_t
\end{aligned} \tag{6}
$$

- V_t为平滑梯度，其展开为$V_t = g_t + \beta g_{t-1} + \beta^2 g_{t-2} + \beta^3 g_{t-3} + \ldots$
- β的常见取值为$[0.5, 0.9, 0.95, 0.99]$



Adam

Adam是一个汇集了多种技术的高效优化算法，它非常的平滑，对学习率不敏感。该算法主要分为两部分：

Part 1：

$$\begin{aligned} V_t &= \beta_1 V_{t-1} + (1-\beta_1) g_t \\ &= (1-\beta_1)(g_t + \beta_1 g_{t-1} + \beta_1^2 g_{t-2} + \ldots) \end{aligned} \tag{7}$$

- 通常，β_1的取值为0.9。与冲量法相比，Adam更多的考虑了过往梯度的影响，使得梯度更加平滑。
- 在g_t前面加上$(1-\beta_1)$的目的是为了使权重和为1，这是因为$\sum\limits_{i=1}^{\infty} \beta_1^t = \frac{1}{1-\beta_1}$。
- 但是，当t的值不够大时，我们需要对其进行修正，方法如下：

$$\hat{V}_t = \frac{V_t}{1-\beta_1^t} \tag{8}$$

Part 2:

为了使该优化算法对学习率不敏感，需要对梯度向量进行一个类似归一化的处理，避免梯度爆炸或梯度消失的情况。

$$\begin{aligned} S_t &= \beta_2 S_{t-1} + (1-\beta_2) g_t^2 \\ \hat{S}_t &= \frac{S_t}{1-\beta_2^t} \\ g_t' &= \frac{\hat{V}_t}{\sqrt{\hat{S}_t} + \epsilon} \end{aligned} \tag{9}$$

- 通常β_2的取值为0.999，这使得S_t的变化非常平滑。
- 与V_t一样，S_t也需要进行修正处理。
- g_t'为经过归一化处理的梯度，控制梯度的每个维度值在合适的大小，ϵ的作用是防止分母为0。

最终权重的更新为：

$$W_t = W_{t-1} - \eta g_t' \tag{10}$$