

# 第一章 最优化简介

修贤超

<https://xianchaoxiu.github.io>

- 1.1 最优化问题概括
- 1.2 实例：稀疏优化
- 1.3 实例：深度学习
- 1.4 最优化的基本概念

# 最优化问题的一般形式

## ■ 最优化问题一般可以描述为

$$\begin{array}{ll}\min & f(x) \\ \text{s.t.} & x \in \mathcal{X}\end{array}\quad (1)$$

- $x = (x_1, x_2, \dots, x_n)^\top \in \mathbb{R}^n$  是**决策变量**
- $f: \mathbb{R}^n \rightarrow \mathbb{R}$  是**目标函数**
- $\mathcal{X} \subseteq \mathbb{R}^n$  是**约束集合或可行域**, 可行域包含的点称为**可行解或可行点**
- 当  $\mathcal{X} = \mathbb{R}^n$  时, 问题 (1) 称为**无约束优化问题**
- 集合  $\mathcal{X}$  通常可以由约束函数  $c_i(x): \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, 2, \dots, m + l$  表达为

$$\mathcal{X} = \{x \in \mathbb{R}^n \mid c_i(x) \leq 0, \quad i = 1, 2, \dots, m, \\ c_i(x) = 0, \quad i = m + 1, m + 2, \dots, m + l\}$$

# 最优化问题的一般形式

- 在所有满足约束条件的决策变量中，使目标函数取最小值的变量  $x^*$  称为优化问题 (1) 的**最优解**，即对任意  $x \in \mathcal{X}$  都有

$$f(x) \geq f(x^*)$$

- 如果求解目标函数  $f$  的最大值，则 “min” 应替换为 “max”
- 函数  $f$  的最小（最大）值不一定存在，但其下（上）确界总是存在的
- $x$  可以是矩阵、多维数组或张量等

# 最优化问题的类型

- **线性规划** 目标函数和约束函数均为线性函数的问题
- **整数规划** 变量只能取整数的问题
- **非线性规划** 目标函数和约束函数中至少有一个为非线性函数的问题
- **二次规划** 目标函数是二次函数而约束函数是线性函数的问题
- **半定规划** 在线性约束下极小化关于半正定矩阵的线性函数的问题
- **稀疏优化** 最优解只有少量非零元素的问题
- **非光滑优化** 包含非光滑函数的问题
- **低秩矩阵优化** 最优解是低秩矩阵的问题
- 鲁棒优化、组合优化、随机优化、零阶优化、流形优化、分布式优化等

- 1.1 最优化问题概括
- 1.2 实例：稀疏优化
- 1.3 实例：深度学习
- 1.4 最优化的基本概念

# 稀疏优化

- 给定  $b \in \mathbb{R}^m$ , 矩阵  $A \in \mathbb{R}^{m \times n}$ , 且向量  $b$  的维数远小于向量  $x$  的维数, 即  $m \ll n$ . 考虑线性方程组求解问题

$$Ax = b$$

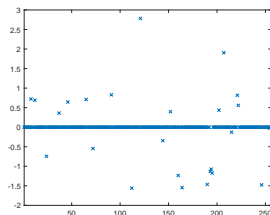
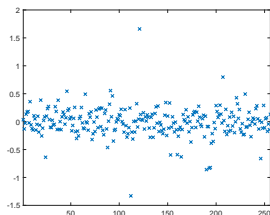
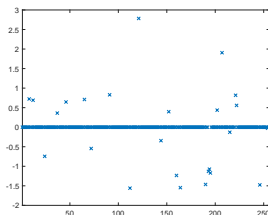
- 方程组欠定, 存在无穷多个解
- 原始信号中有较多的零元素, 即**稀疏解**
- $\ell_0$  是不连续的, 是 NP 难的

$$(\ell_0) \begin{cases} \min_{x \in \mathbb{R}^n} & \|x\|_0 \\ \text{s.t.} & Ax = b \end{cases} \Rightarrow (\ell_2) \begin{cases} \min_{x \in \mathbb{R}^n} & \|x\|_2 \\ \text{s.t.} & Ax = b \end{cases} \quad (\ell_1) \begin{cases} \min_{x \in \mathbb{R}^n} & \|x\|_1 \\ \text{s.t.} & Ax = b \end{cases}$$

- 广泛应用于**压缩感知** (compressive sensing), 即通过部分信息恢复全部信息的解决方案

## ■ MATLAB 仿真

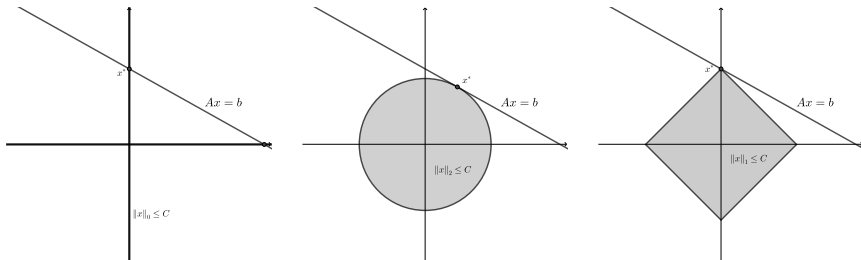
```
1 m = 128; n = 256;  
2 A = randn(m, n); u = sprandn(n, 1, 0.1);  
3 b = A * u;
```



■ 若  $A, b$  满足一定的条件, 向量  $u$  也是  $\ell_1$  范数优化问题的**唯一最优解**



## ■ 原点到仿射集 $Ax = b$ 的投影



■ 绝对值函数在零点处不可微，即非光滑

■  $A$  通常是稠密矩阵，甚至元素未知或者不能直接存储

## ■ 考虑带 $\ell_1$ 范数正则项的优化问题

$$\min_{x \in \mathbb{R}^n} \mu \|x\|_1 \quad \text{s.t. } Ax = b \quad (2)$$

$\Downarrow$

$$\min_{x \in \mathbb{R}^n} \mu \|x\|_1 + \frac{1}{2} \|Ax - b\|_2^2 \quad (3)$$

- $\mu > 0$  是给定的正则化参数
- 称为 LASSO (least absolute shrinkage and selection operator)
- R Tibshirani [JRSSB, 1996], Google Citation: 60087
- DL Donoho [IEEE TIT, 2006], Google Citation: 34645

## ■ 本课程大部分算法都将针对(2)和(3)给出

- 1.1 最优化问题概括
- 1.2 实例：稀疏优化
- 1.3 实例：深度学习
- 1.4 最优化的基本概念

- 深度学习 (deep learning) 是机器学习的一个子领域
- 深度学习的起源可以追溯至 20 世纪 40 年代, 其雏形出现在控制论中
- 常见的激活函数类型

- Sigmoid 函数

$$t(z) = \frac{1}{1 + \exp(-z)}$$

- Heaviside 函数

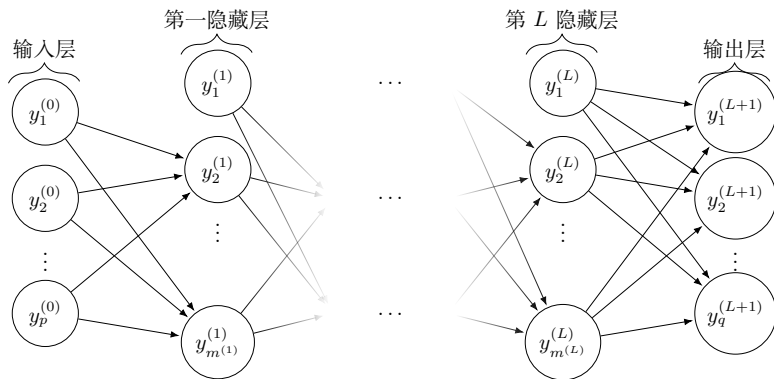
$$t(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$$

- ReLU 函数

$$t(z) = \max\{0, z\}$$

# 多层感知机

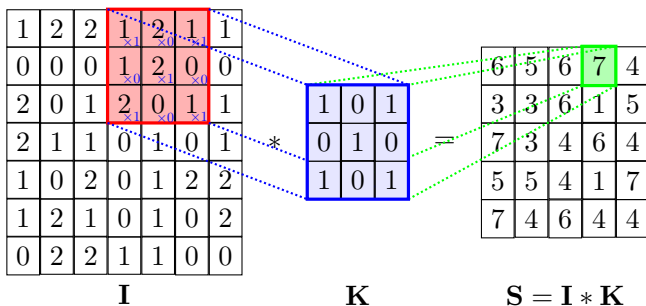
- 多层感知机 (multi-layer perceptron, MLP) 也叫前馈神经网络
- 通过已有的信息或者知识来对未知事物进行预测



# 卷积神经网络

- 卷积神经网络 (convolutional neural network, CNN)
- 给定二维图像  $\mathbf{I} \in \mathbb{R}^{n \times n}$  和卷积核  $\mathbf{K} \in \mathbb{R}^{k \times k}$ , 定义卷积操作  $\mathbf{S} = \mathbf{I} * \mathbf{K}$ , 即

$$\mathbf{S}_{i,j} = \mathbf{I}(i : i + k - 1, j : j + k - 1) \mathbf{K}$$



## ■ 典型的数学模型

$$\min_{x \in \mathcal{W}} \quad \frac{1}{N} \sum_{i=1}^N \ell(f(a_i, x), b_i) + \mu \varphi(x)$$

## ■ 随机梯度类算法

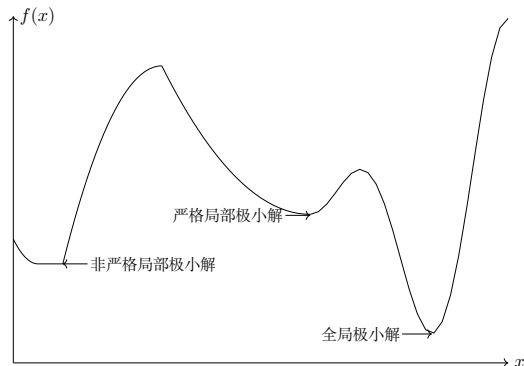
- pytorch/caffe2: adadelta, adagrad, adam, nesterov, rmsprop, YellowFin  
<https://github.com/pytorch/pytorch/tree/master/caffe2/sgd>
- pytorch/torch: sgd, asgd, adagrad, rmsprop, adadelta, adam, adamax  
<https://github.com/pytorch/pytorch/tree/master/torch/optim>
- tensorflow: Adadelta, AdagradDA, Adagrad, ProximalAdagrad, Ftrl, Momentum, adam, Momentum, CenteredRMSProp  
[https://github.com/tensorflow/tensorflow/blob/master/tensorflow/core/kernels/training\\_ops.cc](https://github.com/tensorflow/tensorflow/blob/master/tensorflow/core/kernels/training_ops.cc)

- 1.1 最优化问题概括
- 1.2 实例：稀疏优化
- 1.3 实例：深度学习
- 1.4 最优化的基本概念



# 全局和局部最优解

- 如果  $f(\bar{x}) \leq f(x), \forall x \in \mathcal{X}$ , 则称  $\bar{x}$  为**全局极小解**
- 如果存在  $N_\varepsilon(\bar{x})$  使得  $f(\bar{x}) \leq f(x), \forall x \in N_\varepsilon(\bar{x}) \cap \mathcal{X}$ , 则称  $\bar{x}$  为**局部极小解**.  
进一步地, 如果有  $f(\bar{x}) < f(x), \forall x \in N_\varepsilon(\bar{x}) \cap \mathcal{X}$ , 且  $x \neq \bar{x}$  成立, 则称  $\bar{x}$  为**严格局部极小解**



# 收敛性

- 给定初始点  $x^0$ , 记算法迭代产生的点列为  $\{x^k\}$ . 如果  $\{x^k\}$  在某种范数  $\|\cdot\|$  的意义下满足  $\lim_{k \rightarrow \infty} \|x^k - x^*\| = 0$ , 且收敛的点  $x^*$  为一个局部 (全局) 极小解, 则称该点列收敛到局部 (全局) 极小解, 相应的算法称为依点列收敛到局部 (全局) 极小解
- 如果从任意初始点  $x^0$  出发, 算法都是依点列收敛到局部 (全局) 极小解的, 则称该算法全局依点列收敛到局部 (全局) 极小解
- 记对应的函数值序列  $\{f(x^k)\}$ , 则称该算法 (全局) 依函数值收敛到局部 (全局) 极小值
- 除了点列和函数值的收敛外, 还有每个迭代点的最优性条件 (如无约束优化问题中的梯度范数, 约束优化问题中的最优性条件违反度等等) 的收敛

- Q-线性收敛 对充分大的  $k$  有

$$\frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} \leq a, \quad a \in (0, 1)$$

- Q-次线性收敛 对充分大的  $k$  有

$$\lim_{k \rightarrow \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} = 1$$

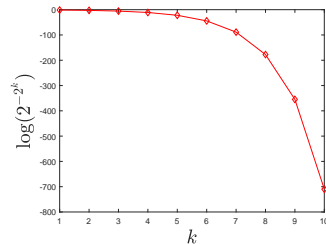
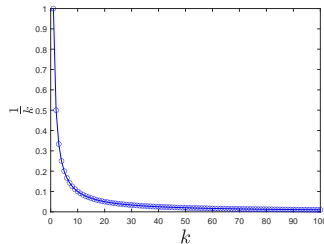
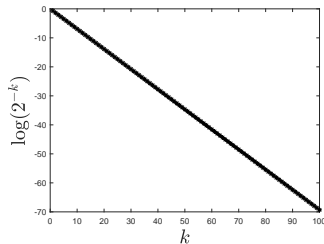
- Q-超线性收敛 对充分大的  $k$  有

$$\lim_{k \rightarrow \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} = 0$$

# 渐进收敛速度

- Q-二次收敛 对充分大的  $k$  有

$$\frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|^2} \leq a, \quad a > 0$$



- 一般来说, 选择 Q-超线性收敛速度和 Q-二次收敛速度的算法

# 收敛准则

- 对于无约束优化问题，常用的收敛准则有

$$\frac{f(x^k) - f^*}{\max\{|f^*|, 1\}} \leq \varepsilon_1, \quad \|\nabla f(x^k)\| \leq \varepsilon_2$$

如果最优解未知，通常使用相对误差

$$\frac{\|x^{k+1} - x^k\|}{\max\{\|x^k\|, 1\}} \leq \varepsilon_3, \quad \frac{|f(x^{k+1}) - f(x^k)|}{\max\{|f(x^k)|, 1\}} \leq \varepsilon_4$$

- 对于约束优化问题，还需要考虑约束违反度

$$c_i(x^k) \leq \varepsilon_5, \quad i = 1, 2, \dots, m,$$
$$|c_i(x^k)| \leq \varepsilon_6, \quad i = m + 1, m + 2, \dots, m + l$$

*Q&A*

*Thank you!*

感谢您的聆听和反馈