

# 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

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

- $x = (x_1, x_2, \dots, x_n)^\top \in \mathbb{R}^n$  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** 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

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

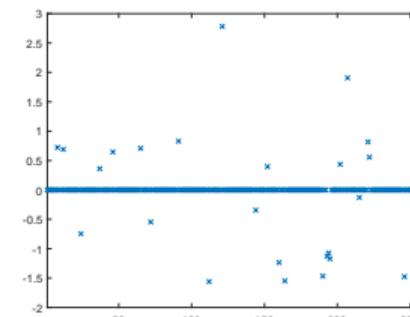
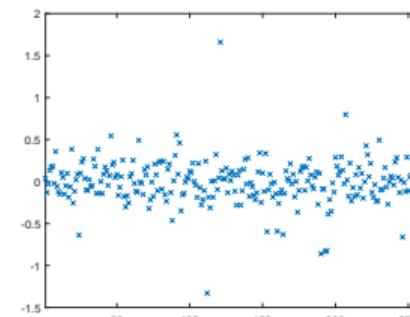
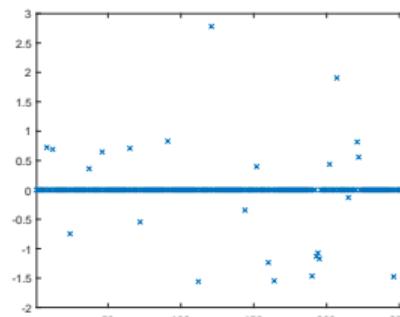
- 压缩感知 (compressive sensing), 即通过部分信息恢复全部信息的解决方案

# 稀疏优化

## ■ 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



## Compressed sensing

[DL Donoho - IEEE Transactions on information theory, 2006 - ieeexplore.ieee.org](#)

Suppose  $x$  is an unknown vector in  $\mathbb{R}^m$  (a digital image or signal); we measure  $n$  general linear functionals of  $x$  and then reconstruct. If  $x$  is known compressible by ...

☆ 被引用次数: 34750 相关文章 ≫

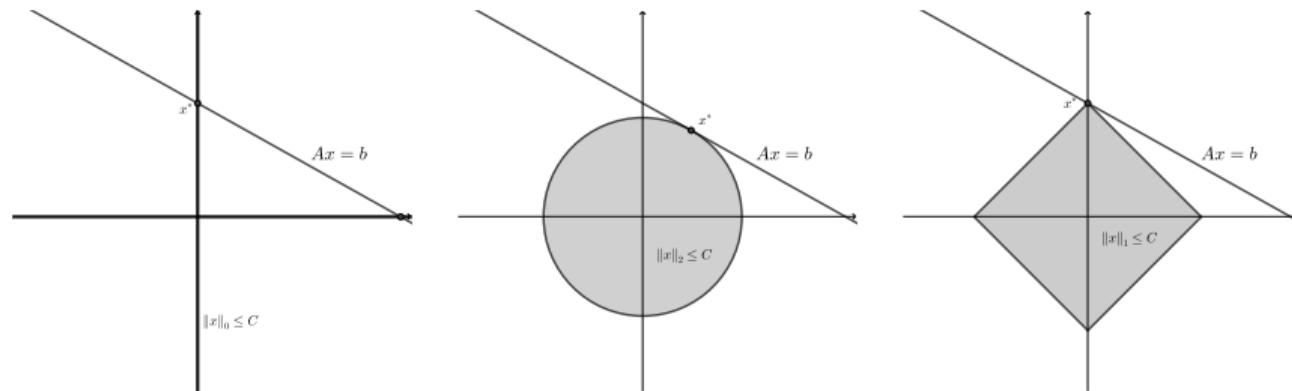
## Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information

[EJ Candès, J Romberg, T Tao - IEEE Transactions on ..., 2006 - ieeexplor](#)

... to **Uncertainty Principles** From a certain point of view, our results are connected to the so-called **uncertainty principles** [4]... will be a novel **uncertainty principle** for generic sets , . . . .

☆ 双引用 被引用次数: 19715 相关文章 所有 27 个版本 ≫

## ■ 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

# LASSO 问题

## ■ 考虑带 $\ell_1$ 范数正则项的优化问题

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



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

- $\mu > 0$  是给定的正则化参数
- 称为 LASSO (least absolute shrinkage and selection operator)
- 本课程大部分算法都将针对(2)和(3)给出

# LASSO 代表作

## Regression shrinkage and selection via the lasso

R Tibshirani - Journal of the Royal Statistical Society Series B ..., 1996  
- academic.oup.com

... methods for estimation of prediction error and the **lasso** shrinkage parameter ...  
Bayes model for the **lasso** is briefly mentioned in Section 5. We describe the algorithm in Section 6. ...

☆ 被引用次数: 60458 相关文章 »

## The adaptive lasso and its oracle properties

H Zou - Journal of the American statistical association, 2006 - Taylor & Francis

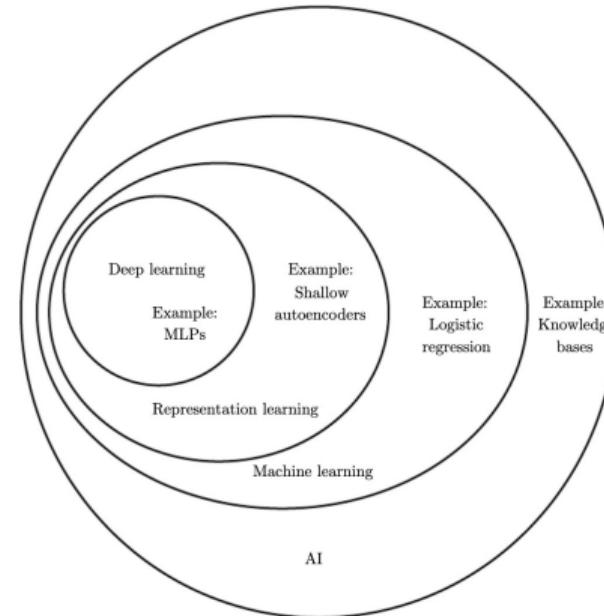
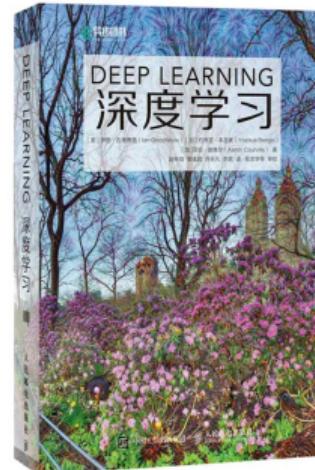
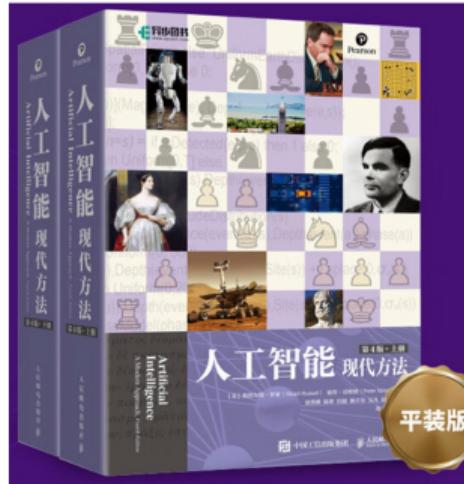
... for the **lasso** variable selection to ... **lasso**, called the adaptive **lasso**, adaptive weights are used for penalizing different coefficients in the L1 problem ... show that the adaptive **lasso** ...

☆ 被引用次数: 8879 相关文章 »

- 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

- 深度学习 (deep learning) 是机器学习的一个子领域
- 起源可以追溯至 20 世纪 40 年代，雏形出现在控制论



## ■ 常见的激活函数类型

### □ Sigmoid 函数

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

### □ Heaviside 函数

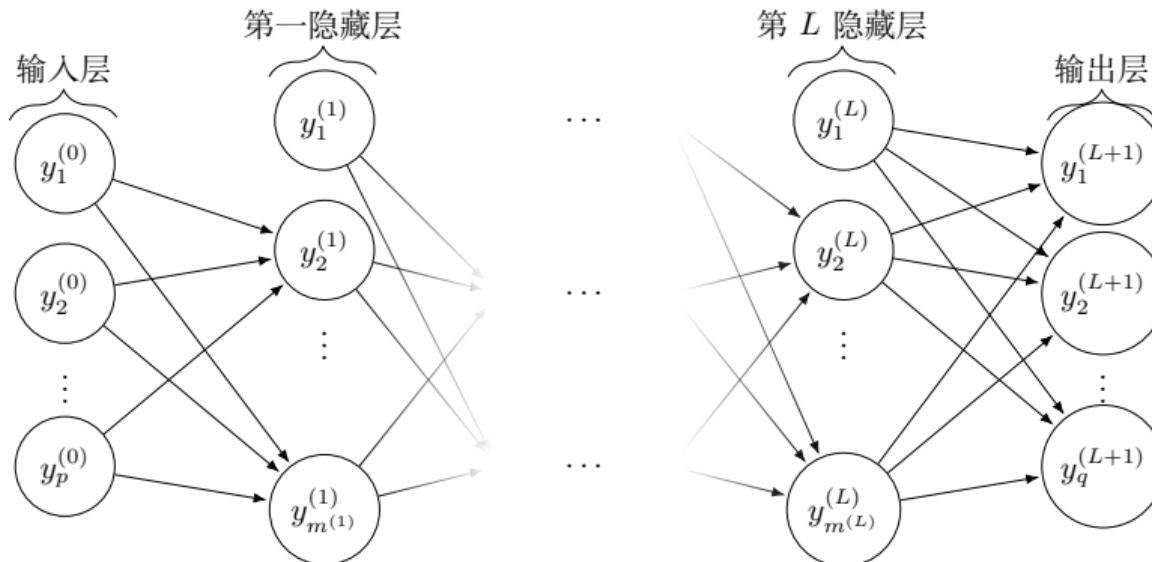
$$t(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$$

### □ ReLU 函数

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

# 多层感知机

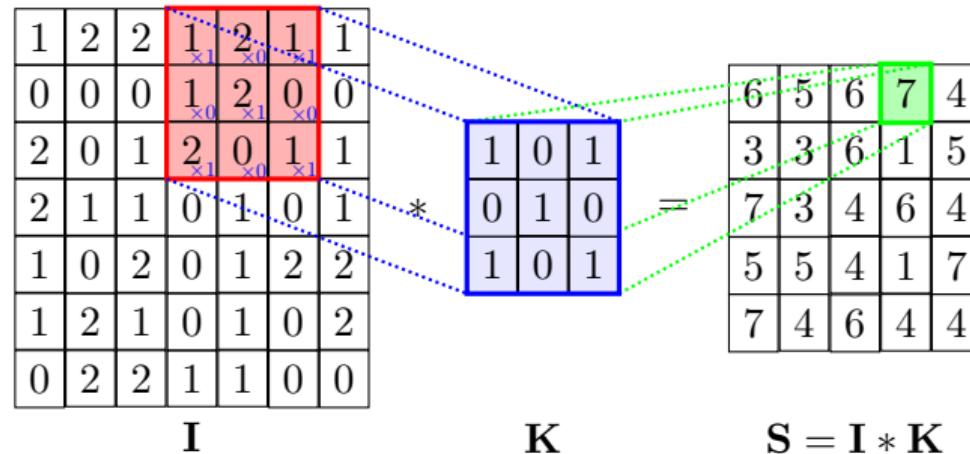
- 多层感知机 (multi-layer perceptron, MLP) 也叫前馈神经网络
- 通过已有的知识来对未知事物进行预测



# 卷积神经网络

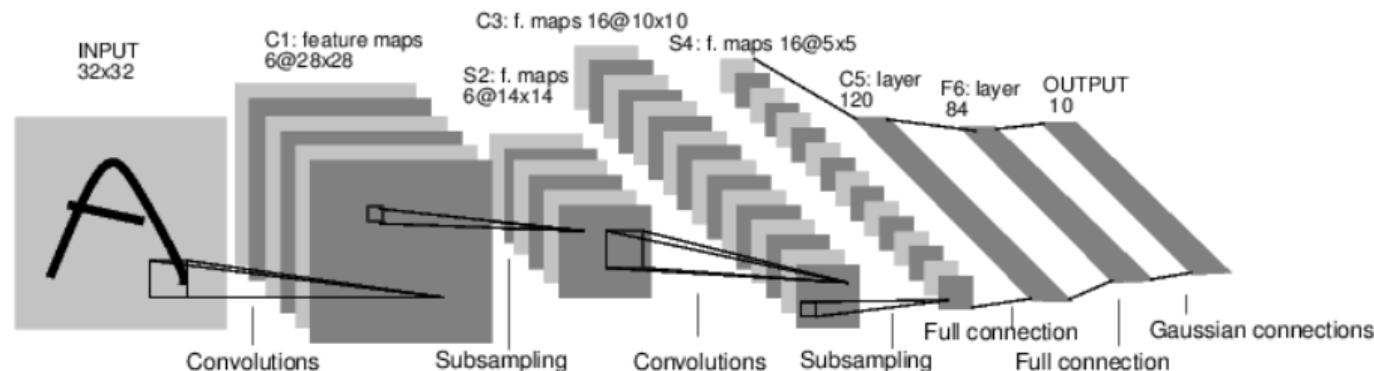
- 卷积神经网络 (convolutional neural network, CNN)
- 给定二维图像  $I \in \mathbb{R}^{n \times n}$  和卷积核  $K \in \mathbb{R}^{k \times k}$ , 定义卷积操作  $S = I * K$ , 即

$$S_{i,j} = \langle I(i:i+k-1, j:j+k-1), K \rangle$$



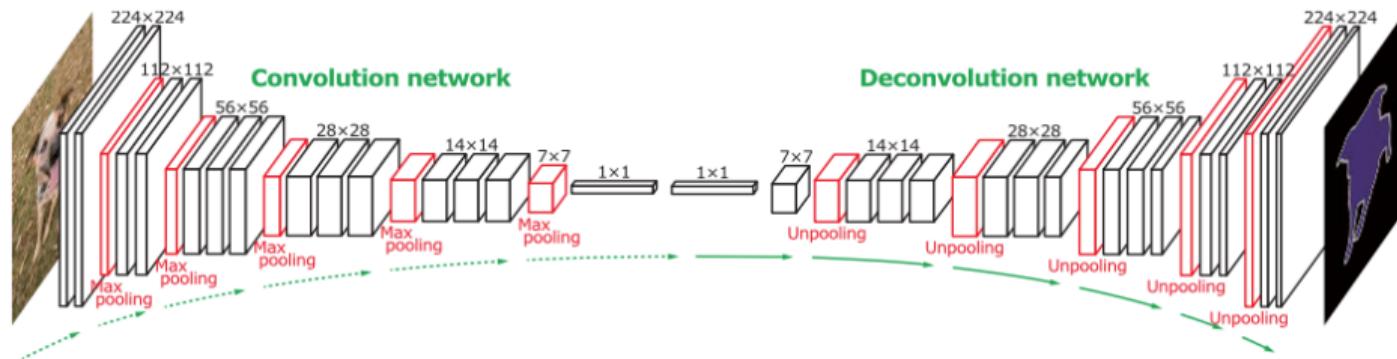
# 卷积神经网络

- LeCun 等人开创性的建立了数字分类的神经网络 LeNet-5，成功在银行识别支票上的手写数字



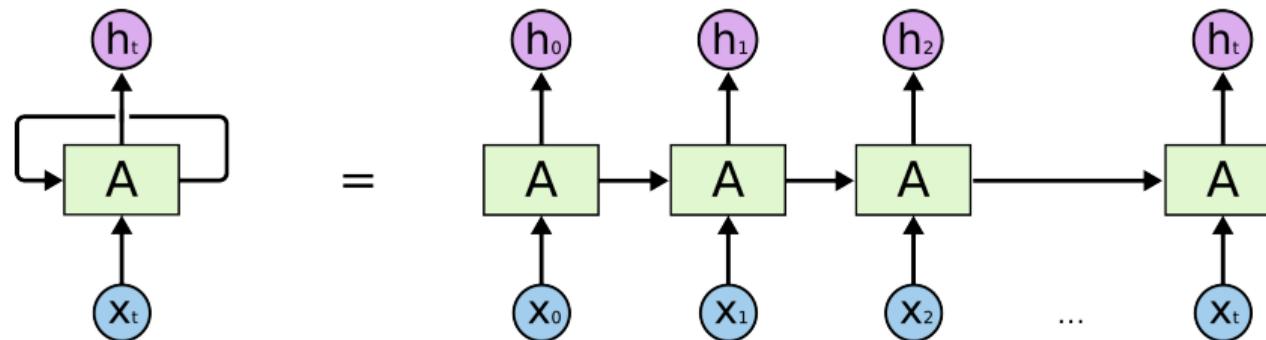
# 反卷积网络

- 生成网络是一种特殊的卷积网络，它使用转置卷积，也称为反卷积，常用的有生成对抗网络、变分自编码器、扩散模型



# 递归神经网络

- 递归神经网络 (recurrent neural networks, RNN) 建立在与前馈神经网络相同的计算单元上，但不必分层组织，并且允许定向循环



# 深度学习中的优化算法

## ■ 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)

## Imagenet classification with deep convolutional neural networks

[A Krizhevsky, I Sutskever... - Advances in neural ...](#), 2012 - proceedings.

We trained a large, deep convolutional neural network to classify the 1.3 million resolution images in the LSVRC-2010 **ImageNet** training set into the 1000 classes. On the ...

☆  引用 被引用次数: 132672 相关文章 所有 98 个版本 HTML |

## Deep residual learning for image recognition

[K He, X Zhang, S Ren, J Sun - Proceedings of the IEEE ...](#), 2016 - openaccess.thecvf.com

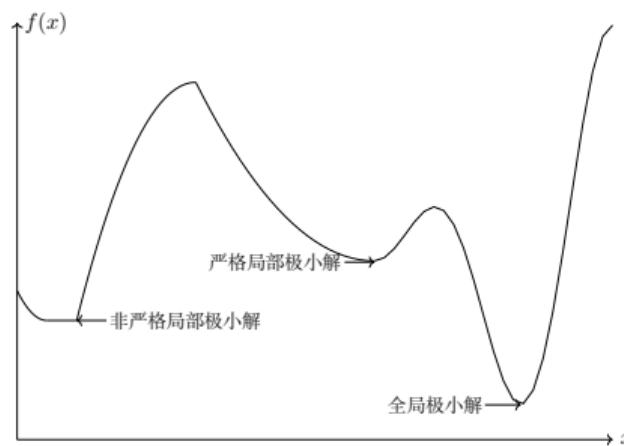
... **Deeper** neural networks are more difficult to train. We present a **res** framework to ease the training of networks that are substantially **deeper** than those used previously. ...

☆  引用 被引用次数: 232781 相关文章 所有 65 个版本 HTML |

- 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, 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，则称该算法**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**

# 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

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

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

# 渐进收敛速度

- 设  $\{x^k\}$  为算法产生的迭代点列且收敛于  $x^*$

□ Q-线性收敛

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

□ Q-次线性收敛

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

□ Q-超线性收敛

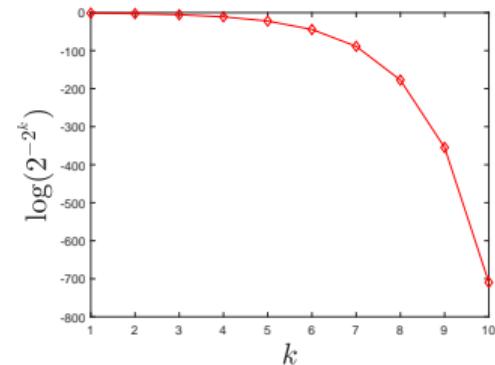
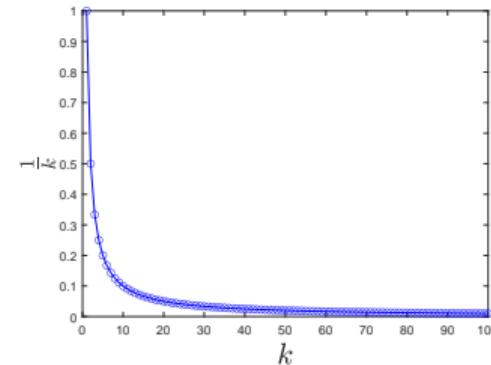
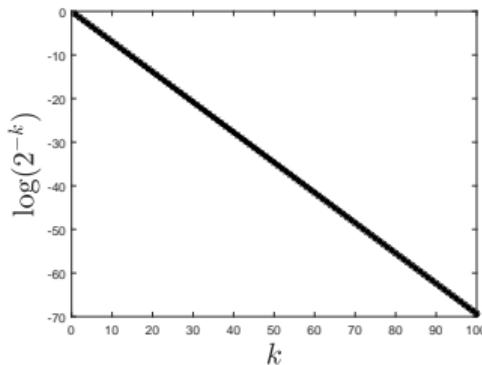
$$\lim_{k \rightarrow \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} = 0$$

□ Q-二次收敛

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

# 渐进收敛速度

- 点列  $\{2^{-k}\}$  是 Q-线性收敛的
- 点列  $\{1/k\}$  是 Q-次线性收敛的
- 点列  $\{2^{-2^k}\}$  是 Q-二次收敛的, 也是 Q-超线性收敛的



一般来说, 选择 Q-超线性收敛和 Q-二次收敛的算法

*Q&A*

*Thank you!*

感谢您的聆听和反馈