# 第一章 引言

人工智能的真正挑战在于解决那些对人来说很容易执行、但很难形式化描述的任务

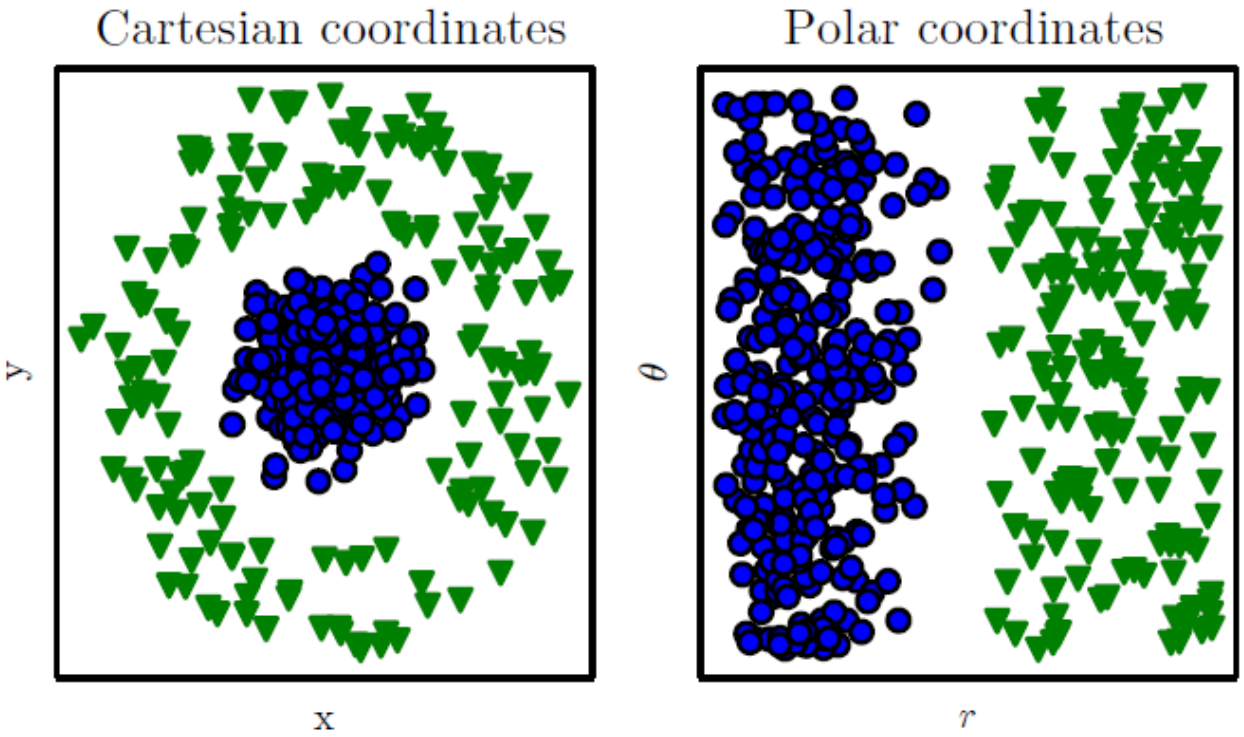
- 让计算机从经验中学习并根据层次化的概念体系来理解世界
- 层次化的概念让计算机构建较简单的概念来学习复杂概念

人工智能的一个关键挑战就是如何将这些**非形式化的知识**传达给计算机。

- 依靠硬编码的知识体系不能解决
- AI系统需要具备自己获取知识的能力——机器学习

# 第一章 引言

- 机器学习算法的性能在很大程度上依赖于给定数据表示
- 表示的选择会对机器学习算法的性能产生巨大的影响



# 第一章 引言

很多AI任务的一个通用解决方案：

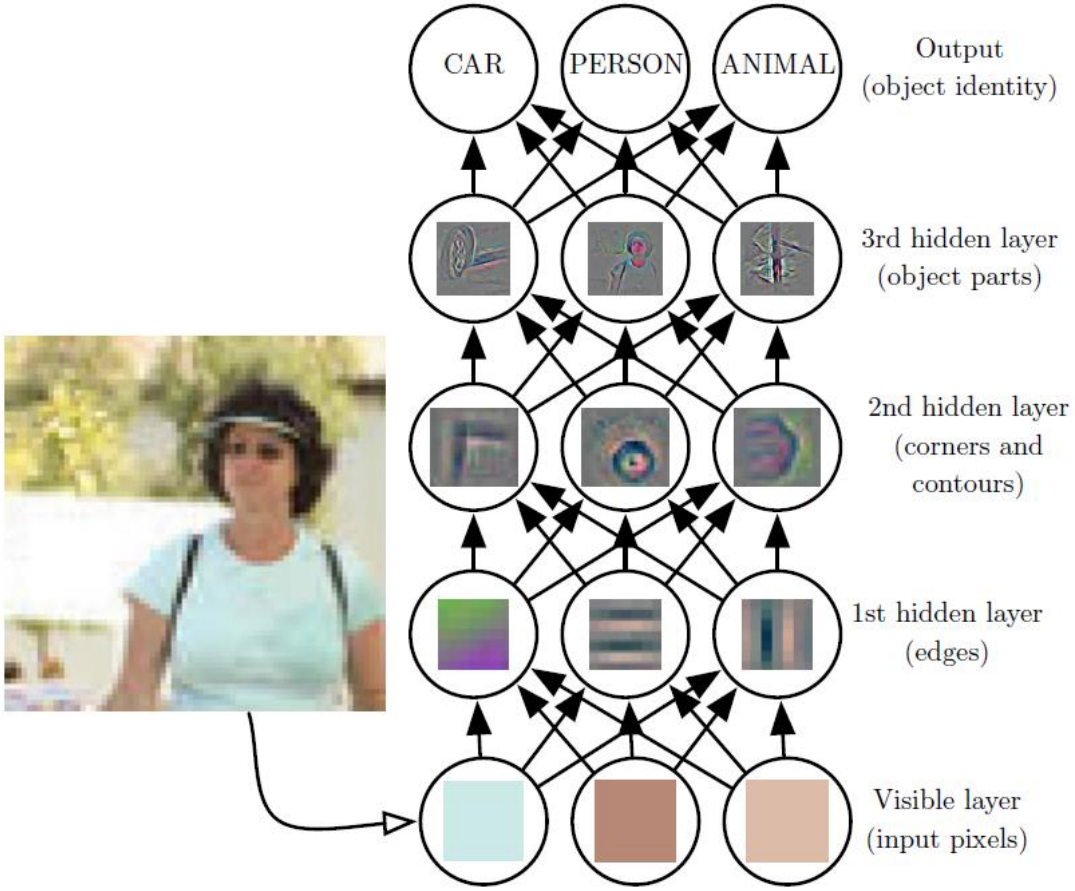
- 提取一个合适的特征集
- 将这些特征提供给简单的机器学习算法

然而，有很多任务很难知道应该提取哪些特征：

- 表示学习——使用机器学习来发掘表示本身
- 深度学习——通过其他较简单的表示来表达复杂表示

# 第一章 引言

深度学习模型示意图：




输出层：

对象识别

隐藏层3：

第三层检测特定对象的整个部分

隐藏层2：

第二层搜索可识别为角和扩展轮廓的边的集合

隐藏层1：

第一层识别边缘

可见层（输入层）：

输入像素

# 第一章 引言

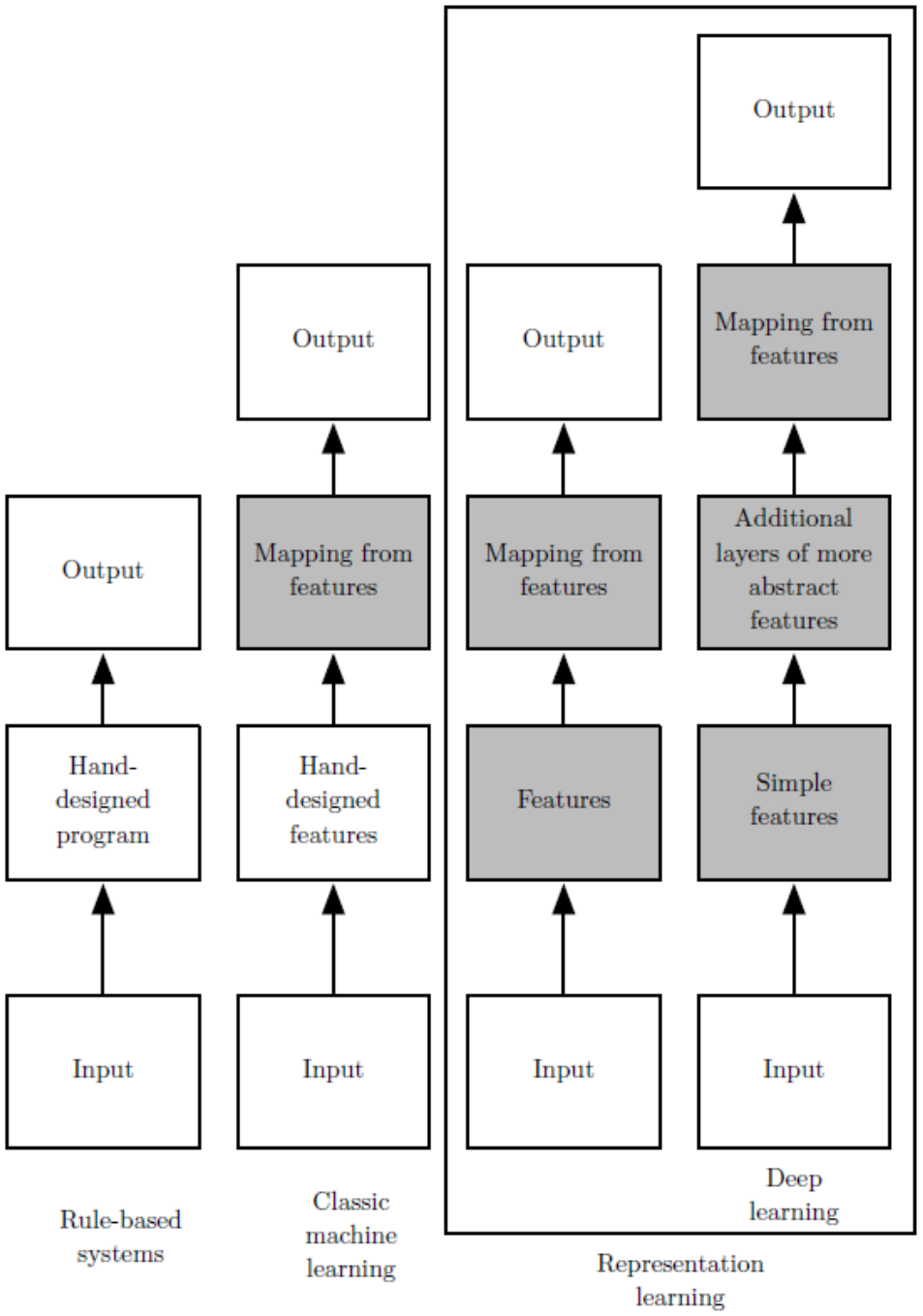
解释深度学习：

- 学习数据的正确表示
- 深度促使计算机学习一个多步骤的计算机程序

两种度量模型深度的方式：

- 基于评估架构所需执行的顺序指令的数
- 将描述概念彼此如何关联的图的深度视为模型深度

# 第一章 引言


Output
Output
Output
Mapping from features
Output
Mapping from features
Mapping from features
Additional layers of more abstract features
Hand-designed program
Hand-designed features
Features
Simple features
Input
Input
Input
Input
Rule-based systems
Classic machine learning
Representation learning
Deep learning

# 第一章 引言

本书组织为三个部分：

- 第一部分介绍基本的数学工具和机器学习的概念
  - 第二到第五章
- 第二部分介绍最成熟的深度学习算法
  - 第六到第十二章
- 第三部分讨论某些具有展望性的想法
  - 第十三到第二十章

# 第二章 线性代数

标量、向量、矩阵和张量：

- 标量：一个标量就是一个单独的数
- 向量：一个向量是一列数
- 矩阵：矩阵是一个二维数组，$\boldsymbol{A}$
- 张量：超过两维的数组，**A**

转置：矩阵的转置是以主对角线为轴的镜像，$\boldsymbol{A}^{\top}$

矩阵和向量相加：

- 向量b 和矩阵A 的每一行相加
- 这种隐式地复制向量b到很多位置的方式，被称为**广播**

# 第二章 线性代数

矩阵和向量相乘：

- 两个矩阵$\boldsymbol{A}$和$\boldsymbol{B}$的矩阵乘积是第三个矩阵$\boldsymbol{C}$
- 乘法操作定义为

$$C_{i,j} = \sum_k A_{i,k} B_{k,j}.$$

- 两个维度相同的向量$x$和$y$的点积可看作是矩阵乘积$x^T y$

- 计算$\boldsymbol{C}_{i,j}$的步骤可以看作计算$\boldsymbol{A}$的第$i$行和$\boldsymbol{B}$的第$j$列的点积

矩阵乘积满足分配率和结合律，不满足交换律，向量点积满足交换律

# 第二章 线性代数

单位矩阵和逆矩阵：

- 保持$n$维向量不变的单位矩阵记作$\boldsymbol{I}_n$
- 形式上，$\boldsymbol{I}_n \in \mathbb{R}^{n\times n}$

$$\forall \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{I}_n \boldsymbol{x} = \boldsymbol{x}.$$

- 单位矩阵的结构很简单：所有沿主对角线的元素都是1，而所有其他位置的元素都是0

矩阵$\boldsymbol{A}$ 的矩阵逆矩阵（matrix inversion）记作$\boldsymbol{A}^{-1}$，

$$\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}_n.$$

# 第二章 线性代数

线性相关和生成子空间：

- 一组向量的线性组合，指的是每个向量乘以对应标量系数之后的和 $\sum_i c_i \boldsymbol{v}^{(i)}$.
- 如果一组向量中的任意一个向量都不能表示成其他向量的线性组合，那么这组向量称为线性无关（linearly independent）
- 反之，这组向量线性相关
- 一组向量的生成子空间（span）是原始向量线性组合后所能抵达的点的集合

# 第二章 线性代数

范数（norm）：

- $L^p$ 范数定义如下：

$$\|\boldsymbol{x}\|_p = \left(\sum_i |x_i|^p\right)^{\frac{1}{p}} \qquad p \in \mathbb{R},\ p \geq 1$$

- 范数是满足下列性质的任意函数：
  - $f(\boldsymbol{x}) = 0 \Rightarrow \boldsymbol{x} = \boldsymbol{0}$
  - $f(\boldsymbol{x}+\boldsymbol{y}) \leq f(\boldsymbol{x}) + f(\boldsymbol{y})$ （**三角不等式**（triangle inequality））
  - $\forall \alpha \in \mathbb{R},\ f(\alpha \boldsymbol{x}) = |\alpha| f(\boldsymbol{x})$
- $L^2$范数被称为欧几里得范数，常简化表示为$||x||$，可以简单地通过点积$\boldsymbol{x}^\top \boldsymbol{x}$ 计算

# 第二章 线性代数

范数（norm）：

- $L^1$ 范数常作为表示非零元素数目的替代函数
- $L^\infty$范数，也被称为最大范数，表示**向量中具有最大幅值的元素的绝对值**：

$$\|\boldsymbol{x}\|_\infty = \max_i |x_i|.$$

- **Frobenius** 范数，衡量矩阵的大小：

$$\|\boldsymbol{A}\|_F = \sqrt{\sum_{i,j} A_{i,j}^2},$$

- 两个向量的点积（dot product）可以用范数来表示

$$\boldsymbol{x}^\top \boldsymbol{y} = \|\boldsymbol{x}\|_2 \|\boldsymbol{y}\|_2 \cos\theta$$

# 第二章 线性代数

特殊类型的矩阵和向量：

- 对角矩阵：只在主对角线上含有非零元素，其他位置都是零
- 对称矩阵：矩阵是转置和自己相等的矩阵， $\boldsymbol{A} = \boldsymbol{A}^\top$.
- 单位向量：具有单位范数的向量， $\|\boldsymbol{x}\|_2 = 1$.
- 正交：$\boldsymbol{x}^\top \boldsymbol{y} = 0$，向量$\boldsymbol{x}$ 和向量$\boldsymbol{y}$ 互相正交
- 如果这些向量不仅互相正交，并且范数都为1，那么我们称它们**标准正交**
- 正交矩阵：指行向量和列向量是分别标准正交的方阵

$$\boldsymbol{A}^\top \boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^\top = \boldsymbol{I}. \qquad \boldsymbol{A}^{-1} = \boldsymbol{A}^\top$$

# 第二章 线性代数

特征分解：

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}.$$

$\boldsymbol{v}$是特征向量，标量$\lambda$被称为这个特征向量对应的特征值

$\boldsymbol{A}$ 的特征分解：

$$\boldsymbol{A} = \boldsymbol{V}\mathrm{diag}(\boldsymbol{\lambda})\,\boldsymbol{V}^{-1}.$$

$$\boldsymbol{V} = [\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(n)}]$$

$$\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_n]^\top$$

每个实对称矩阵都可以分解成实特征向量和实特征值：

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top.$$

其中$\boldsymbol{Q}$ 是$\boldsymbol{A}$ 的特征向量组成的正交矩阵，$\boldsymbol{\Lambda}$是对角矩阵

# 第二章 线性代数

特征分解：

- 正定：所有特征值都是正数的矩阵被称为正定
- 半正定：所有特征值都是非负数的矩阵被称为半正定
- 负定：所有特征值都是负数的矩阵被称为负定
- 半负定：所有特征值都是非正数的矩阵被称为半负定

# 第二章 线性代数

奇异值分解：

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\top}.$$

$\boldsymbol{A}$ 是$m \times n$的矩阵，$\boldsymbol{U}$ 是$m \times m$的矩阵
$\boldsymbol{D}$ 是$m \times n$的矩阵，$\boldsymbol{V}$ 是$n \times n$的矩阵

这些矩阵中的每一个经定义后都拥有特殊的结构：

- $\boldsymbol{U}$和$\boldsymbol{V}$都定义为正交矩阵
- 矩阵$\boldsymbol{D}$ 定义为对角矩阵，注意， $\boldsymbol{D}$不一定是方阵

# 第二章 线性代数

**Moore-Penrose** 伪逆：

矩阵$\boldsymbol{A}$的伪逆定义为：

$$\boldsymbol{A}^{+} = \lim_{\alpha \searrow 0} (\boldsymbol{A}^{\top}\boldsymbol{A} + \alpha \boldsymbol{I})^{-1}\boldsymbol{A}^{\top}.$$

计算伪逆的实际算法是使用下面的公式：

$$\boldsymbol{A}^{+} = \boldsymbol{V}\boldsymbol{D}^{+}\boldsymbol{U}^{\top}.$$

矩阵$\boldsymbol{U}$，$\boldsymbol{D}$ 和$\boldsymbol{V}$ 是矩阵$\boldsymbol{A}$奇异值分解后得到的矩阵
对角矩阵$\boldsymbol{D}$ 的伪逆$\boldsymbol{D}^{+}$ 是其非零元素取倒数之后再转置得到的

# 第二章 线性代数

**Moore-Penrose** 伪逆：

矩阵$\boldsymbol{A}$不是方阵：

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$$

如果矩阵$\boldsymbol{A}$ 的行数小于列数，那么上述方程可能有多个解

如果矩阵$\boldsymbol{A}$ 的行数大于列数，那么上述方程可能没有解

如果矩阵$\boldsymbol{A}$ 的行数小于列数，$\boldsymbol{x} = \boldsymbol{A}^{+}\boldsymbol{y}$是方程所有可行解中欧几里得范数最小的一个

如果矩阵$\boldsymbol{A}$ 的行数大于列数，可能没有解，在这种情况下，通过伪逆得到的$\boldsymbol{x}$使得$\boldsymbol{A}\boldsymbol{x}$和$\boldsymbol{y}$的欧几里得距离$||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}||_{\boldsymbol{2}}$最小

# 第二章 线性代数

## 迹运算：

迹运算返回的是矩阵主对角线上的元素的和：

$$\mathrm{Tr}(\boldsymbol{A}) = \sum_i \boldsymbol{A}_{i,i}.$$

迹运算提供了另一种描述矩阵Frobenius范数的方式：

$$\|A\|_F = \sqrt{\mathrm{Tr}(\boldsymbol{A}\boldsymbol{A}^\top)}.$$

多个矩阵相乘得到的方阵的迹，和将这些矩阵中的最后一个挪到最前面之后相乘的迹是相同的

$$\mathrm{Tr}(\prod_{i=1}^{n} \boldsymbol{F}^{(i)}) = \mathrm{Tr}(\boldsymbol{F}^{(n)} \prod_{i=1}^{n-1} \boldsymbol{F}^{(i)}).$$

标量在迹运算后仍然是它自己

# 实例 主成分分析（PCA）

假设在$\mathbb{R}^n$空间中有$m$个点$\{x^{(1)}, \ldots, x^{(m)}\}$：

对这些点进行有损压缩，使用更少的内存，但损失一些精度，我们希望损失的精度尽可能少。

方法：

$x^{(i)} \in \mathbb{R}^n$ -> 编码向量$c^{(i)} \in \mathbb{R}^l$， $l < n$

编码函数，根据输入返回编码： $f(\boldsymbol{x}) = \boldsymbol{c}$

解码函数，给定编码的重构输入： $\boldsymbol{x} \approx g(f(\boldsymbol{x}))$

简化解码器：令 $g(\boldsymbol{c}) = \boldsymbol{D}\boldsymbol{c}$， $\boldsymbol{D} \in \mathbb{R}^{n \times l}$

限制1：***D***的列向量都有单位范数（为了获得唯一解）

限制2：***D***的列向量彼此正交（简化解码器的最优编码问题）

# 实例 主成分分析（PCA）

解最优编码问题：

思路：最小化原始输入向量 $\boldsymbol{x}$ 和重构向量$g(\boldsymbol{c})$之间的距离
可以使用$L^2$范数：

$$\boldsymbol{c}^* = \arg\min_{\boldsymbol{c}} \|\boldsymbol{x} - g(\boldsymbol{c})\|_2 .$$

因为$L^2$范数非负，可以使用平方$L^2$范数：

$$\boldsymbol{c}^* = \arg\min_{\boldsymbol{c}} \|\boldsymbol{x} - g(\boldsymbol{c})\|_2^2 .$$

根据定义：

$$\begin{aligned} ||\boldsymbol{x} - g(\boldsymbol{c})||_2^2 &= \big(\boldsymbol{x} - g(\boldsymbol{c})\big)^T \big(\boldsymbol{x} - g(\boldsymbol{c})\big) \\ &= \boldsymbol{x}^T\boldsymbol{x} - 2\boldsymbol{x}^T g(\boldsymbol{c}) + g(\boldsymbol{c})^T g(\boldsymbol{c}) \end{aligned}$$

# 实例 主成分分析（PCA）

解最优编码问题：

$\boldsymbol{x}^T\boldsymbol{x}$不依赖与$\boldsymbol{c}$，可以直接忽略：

$$\begin{aligned}\boldsymbol{c}^* &= \arg\min_{\boldsymbol{c}} -2\boldsymbol{x}^\top g(\boldsymbol{c}) + g(\boldsymbol{c})^\top g(\boldsymbol{c}). \\ &= \arg\min_{\boldsymbol{c}} -2\boldsymbol{x}^\top \boldsymbol{D}\boldsymbol{c} + \boldsymbol{c}^\top \boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{c} \\ &= \arg\min_{\boldsymbol{c}} -2\boldsymbol{x}^\top \boldsymbol{D}\boldsymbol{c} + \boldsymbol{c}^\top \boldsymbol{I}_l \boldsymbol{c} \\ &= \arg\min_{\boldsymbol{c}} -2\boldsymbol{x}^\top \boldsymbol{D}\boldsymbol{c} + \boldsymbol{c}^\top \boldsymbol{c}\end{aligned}$$

微积分求解：

$$\begin{aligned}\nabla_{\boldsymbol{c}}(-2\boldsymbol{x}^\top \boldsymbol{D}\boldsymbol{c} + \boldsymbol{c}^\top \boldsymbol{c}) &= 0 \\ -2\boldsymbol{D}^\top \boldsymbol{x} + 2\boldsymbol{c} &= 0 \\ \boldsymbol{c} &= \boldsymbol{D}^\top \boldsymbol{x}.\end{aligned}$$

于是：

$$f(\boldsymbol{x}) = \boldsymbol{D}^\top \boldsymbol{x}. \qquad r(\boldsymbol{x}) = g(f(\boldsymbol{x})) = \boldsymbol{D}\boldsymbol{D}^\top \boldsymbol{x}.$$

# 实例 主成分分析（PCA）

挑选编码矩阵$\boldsymbol{D}$：

回顾问题：最小化原始输入向量 $\boldsymbol{x}$ 和重构向量$g(\boldsymbol{c})$之间的距离

$$\boldsymbol{c}^* = \arg\min_{\boldsymbol{c}} \|\boldsymbol{x} - g(\boldsymbol{c})\|_2$$

此时，必须最小化所有维数和所有点上的误差矩阵，所以使用 Frobenius 范数：

$$\boldsymbol{D}^* = \arg\min_{\boldsymbol{D}} \sqrt{\sum_{i,j} \left(\boldsymbol{x}_j^{(i)} - r(\boldsymbol{x}^{(i)})_j\right)^2} \text{ subject to } \boldsymbol{D}^\top \boldsymbol{D} = \boldsymbol{I}_l.$$

我们首先考虑$l = 1$的情况，此时， $\boldsymbol{D}$是一个向量$\boldsymbol{d}$：

$$\boldsymbol{d}^* = \arg\min_{\boldsymbol{d}} \sum_i \left\| \boldsymbol{x}^{(i)} - \boldsymbol{d}\boldsymbol{d}^\top \boldsymbol{x}^{(i)} \right\|_2^2 \text{ subject to } \|\boldsymbol{d}\|_2 = 1.$$

# 实例 主成分分析（PCA）

挑选编码矩阵**D**：

$$\boldsymbol{d}^* = \arg\min_{\boldsymbol{d}} \sum_i \left\| \boldsymbol{x}^{(i)} - \boldsymbol{d}\boldsymbol{d}^\top \boldsymbol{x}^{(i)} \right\|_2^2 \text{ subject to } \|\boldsymbol{d}\|_2 = 1.$$

$d^T x^{(i)}$是标量

$$\boldsymbol{d}^* = \arg\min_{\boldsymbol{d}} \sum_i \left\| \boldsymbol{x}^{(i)} - \boldsymbol{d}^\top \boldsymbol{x}^{(i)} \boldsymbol{d} \right\|_2^2 \text{ subject to } \|\boldsymbol{d}\|_2 = 1.$$

写到左边

$$\boldsymbol{d}^* = \arg\min_{\boldsymbol{d}} \sum_i \left\| \boldsymbol{x}^{(i)} - \boldsymbol{x}^{(i)\top} \boldsymbol{d}\boldsymbol{d} \right\|_2^2 \text{ subject to } \|\boldsymbol{d}\|_2 = 1.$$

转置

将表示各点的向量按行堆叠成一个矩阵**X**：

$$\boldsymbol{d}^* = \arg\min_{\boldsymbol{d}} \left\| \boldsymbol{X} - \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^\top \right\|_F^2 \text{ subject to } \boldsymbol{d}^\top \boldsymbol{d} = 1.$$

# 实例 主成分分析（PCA）

挑选编码矩阵$\boldsymbol{D}$：

将表示各点的向量按行堆叠成一个矩阵$\boldsymbol{X}$：

$$\boldsymbol{d}^* = \arg\min_{\boldsymbol{d}} \left\| \boldsymbol{X} - \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^\top \right\|_F^2 \text{ subject to } \boldsymbol{d}^\top \boldsymbol{d} = 1.$$

不考虑约束，简化Frobenius 范数：

$$\begin{aligned}
\arg\min_{\boldsymbol{d}} \left\| \boldsymbol{X} - \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^\top \right\|_F^2 &= \arg\min_{\boldsymbol{d}} \operatorname{Tr}\left( \left( \boldsymbol{X} - \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^\top \right)^\top \left( \boldsymbol{X} - \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^\top \right) \right) \\
&= \arg\min_{\boldsymbol{d}} \operatorname{Tr}\left( \boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^\top - \boldsymbol{d}\boldsymbol{d}^\top \boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{d}\boldsymbol{d}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^\top \right) \\
&= \arg\min_{\boldsymbol{d}} \operatorname{Tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \operatorname{Tr}(\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^\top) - \operatorname{Tr}(\boldsymbol{d}\boldsymbol{d}^\top \boldsymbol{X}^\top \boldsymbol{X}) + \operatorname{Tr}(\boldsymbol{d}\boldsymbol{d}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^\top) \\
&= \arg\min_{\boldsymbol{d}} -\operatorname{Tr}(\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^\top) - \operatorname{Tr}(\boldsymbol{d}\boldsymbol{d}^\top \boldsymbol{X}^\top \boldsymbol{X}) + \operatorname{Tr}(\boldsymbol{d}\boldsymbol{d}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^\top) \\
&= \arg\min_{\boldsymbol{d}} -2\operatorname{Tr}(\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^\top) + \operatorname{Tr}(\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^\top \boldsymbol{d}\boldsymbol{d}^\top)
\end{aligned}$$

# 实例 主成分分析（PCA）

挑选编码矩阵$\boldsymbol{D}$：

加上约束：

$$\arg\min_{\boldsymbol{d}} -2\mathrm{Tr}(\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^\top) + \mathrm{Tr}(\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^\top \boldsymbol{d}\boldsymbol{d}^\top) \text{ subject to } \boldsymbol{d}^\top \boldsymbol{d} = 1$$

$$= \arg\min_{\boldsymbol{d}} -2\mathrm{Tr}(\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^\top) + \mathrm{Tr}(\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^\top) \text{ subject to } \boldsymbol{d}^\top \boldsymbol{d} = 1$$

$$= \arg\min_{\boldsymbol{d}} -\mathrm{Tr}(\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^\top) \text{ subject to } \boldsymbol{d}^\top \boldsymbol{d} = 1$$

$$= \arg\max_{\boldsymbol{d}} \mathrm{Tr}(\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{d}\boldsymbol{d}^\top) \text{ subject to } \boldsymbol{d}^\top \boldsymbol{d} = 1$$

$$= \arg\max_{\boldsymbol{d}} \mathrm{Tr}(\boldsymbol{d}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{d}) \text{ subject to } \boldsymbol{d}^\top \boldsymbol{d} = 1.$$

# 实例 主成分分析（PCA）

挑选编码矩阵$\boldsymbol{D}$：

$$\arg\max_{\boldsymbol{d}} \mathrm{Tr}(\boldsymbol{d}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{d}) \text{ subject to } \boldsymbol{d}^\top \boldsymbol{d} = 1.$$

这个优化问题可以通过特征分解来求解。具体来讲，最优的$\boldsymbol{d}$是$\boldsymbol{X}^T\boldsymbol{X}$最大特征值对应的特征向量

以上推导特定于$l = 1$的情况，仅得到了第一个主成分。

当我们希望得到主成分的基时，矩阵$\boldsymbol{D}$是由$\boldsymbol{X}^T\boldsymbol{X}$的前$l$个最大的特征值对应的特征向量组成的。