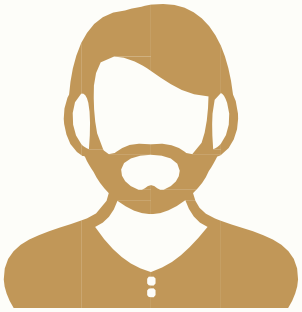
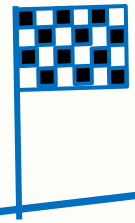
第3章 矩阵

第06讲 矩阵的应用

传媒与信息工程学院
欧 新 宇

- 矩阵的定义及基本操作
- 基于矩阵的向量
- 特殊形态的矩阵
- 矩阵的四则运算
- 矩阵的秩和矩阵的迹
- 矩阵的分块
- 张量的常用操作
- 矩阵的应用





从线性变换的角度看矩阵与向量的乘法

从线性变换的角度看矩阵与向量的乘法

矩阵与向量的乘法可以理解成：向量 $\boldsymbol{x}$ 到向量 $\boldsymbol{y}$ 的线性变换。

【**定义6.1**】对于向量 $\boldsymbol{y} = [y_1, y_2, \ldots, y_m]^T$，若它能由向量 $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$ 线性表示，即有：

$$\begin{cases} y_1 = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ y_2 = a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \cdots \\ y_m = a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{cases}$$

则称此关系式为向量 $\boldsymbol{x}$ 到向量 $\boldsymbol{y}$ 的**线性变换**，可以写成输出向量 $\boldsymbol{y}$ 等于系数矩阵 $\boldsymbol{A}$ 左乘输入向量 $\boldsymbol{x}$：

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_m \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_m \end{bmatrix} = \begin{bmatrix} a_{11}x1 + a_{12}x2 + \ldots + a_{1n}xn \\ a_{21}x1 + a_{22}x2 + \ldots + a_{2n}xn \\ \cdots \\ a_{m1}x1 + a_{m2}x2 + \ldots + a_{mn}xn \end{bmatrix} = AX$$

从线性变换的角度看矩阵与向量的乘法

在进行向量与矩阵乘法的时候，矩阵A在左，列向量x在右，即Ax的顺序不能变。对照矩阵与矩阵的乘法规则，我们可以总结矩阵与向量的乘法规则：当把列向量看作是一个列数为1的特殊矩阵时，那么运算过程就变得比较简单了。

- 矩阵在左，列向量在右，矩阵的列数和列向量的维数必须相等
- 矩阵和列向量相乘的结果也是一个列向量；
- 矩阵的行数就是结果向量的维数；
- 乘法运算的实施过程就是：矩阵的每行和列向量的对应元素分别相乘后，再相加。

从线性变换的角度看矩阵与向量的乘法

多次线性变换等价于矩阵连乘

设 $\begin{cases} y_1 = a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \\ y_2 = a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \end{cases}$ 可写成 $y = Ax$

及 $\begin{cases} x_1 = b_{11}t_1 + b_{12}t_2 \\ x_2 = b_{21}t_1 + b_{22}t_2 \\ x_3 = b_{31}t_1 + b_{32}t_2 \end{cases}$ 可写成 $x = Bt$

则有：

$$\mathbf{Y} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = \mathbf{ABT}$$

$$= \begin{bmatrix} a_{11}b_{11}+a_{12}b_{21}+a_{13}b_{31} & a_{11}b_{12}+a_{12}b_{22}+a_{13}b_{32} \\ a_{21}b_{11}+a_{22}b_{21}+a_{23}b_{31} & a_{21}b_{12}+a_{22}b_{22}+a_{23}b_{32} \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

从线性变换的角度看矩阵与向量的乘法

从矩阵乘法的角度看线性方程组

线性方程组

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases}$$

可看成系数矩阵A与输入变量X的乘积：A*X=b

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

从线性变换的角度看矩阵与向量的乘法

【例6.1】 给出一个矩阵与向量相乘的例子：

$$Ax = \begin{bmatrix} 1 & 2 \\ 4 & 5 \\ 7 & 8 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1*3+2*4 \\ 4*3+5*4 \\ 7*3+8*4 \end{bmatrix} = \begin{bmatrix} 11 \\ 32 \\ 53 \end{bmatrix} = B$$

```
import numpy as np
A = np.array([[1,2],
              [4,5],
              [7,8]])
x = np.array([[3,4]]).T
print(np.dot(A,x))
```

```
[[11]
 [32]
 [53]]
```

【结果分析】

从程序运行的结果来看，原始向量 x 表示二维平面的一个点，其在二维平面中的坐标为A(3，4)，经过与矩阵A的乘法运算后，最终将原始点A转换为新的目标点B，其空间坐标为B(11,32,53)。

从以上的例子可以总结出矩阵所发挥的重要作用：在指定矩阵的作用下，原始空间中的向量被**映射**转换到了目标空间的新坐标，向量的空间位置由此发生了变化，甚至在映射后，目标空间的维数想较于原始空间都有可能发生改变。

课堂互动一 Link

从向量的角度看矩阵乘法

从向量的角度看矩阵乘法

矩阵的行向量形式和列向量形式

如果***A***是一个$m \times n$的矩阵，***A***的每一行为一个实的n元组，于是可以将其看成是$\mathbf{R}^{1\times n}$中的一个向量。对应于***A***的m个行的向量称为***A***的行向量（row vector），记作：

$$A = \begin{bmatrix} a_{row_1} \\ a_{row_2} \\ \vdots \\ a_{row_m} \end{bmatrix}$$

如果***A***是一个$m \times n$的矩阵，则***A***的每一列可以看成是 $\mathbf{R}^{m}$ 中的一个向量，且称这n个向量为***A***的列向量(column vector)，记作：

$$A = [a_{col_1}, a_{col_1}, \ldots, a_{col_1}]$$

从向量的角度看矩阵乘法

行空间和列空间

【定义6.1】

- 如果$\boldsymbol{A}$为一个$m \times n$的矩阵，由$\boldsymbol{A}$的行向量张成的$\mathbf{R}^{1\times n}$的子空间称为$\boldsymbol{A}$的行空间(row space)。
- 由$\boldsymbol{A}$的各列张成的$\mathbf{R}^{m}$的子空间称为$\boldsymbol{A}$的列空间(column space)。

从向量的角度看矩阵乘法

例题讲解

【例6.2】 令 $A=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}$，则:

- ***A*** 的行空间有如下形式的三元组：$\alpha(1,0,0)+\beta(0,1,0)=(\alpha,\beta,0)$
- ***A*** 的列空间是所有如下形式的向量：$\alpha\begin{bmatrix}1\\0\end{bmatrix}+\beta\begin{bmatrix}0\\1\end{bmatrix}+\gamma\begin{bmatrix}0\\0\end{bmatrix}=\begin{bmatrix}\alpha\\\beta\end{bmatrix}$

因此，***A***的行空间为一个 $\mathbf{R}^{1\times 3}$ 的二维子空间，且A的列空间为 $\mathbf{R}^2$。

从向量的角度看矩阵乘法

基于行视角的矩阵乘法

下面我们来回顾一下矩阵和向量相乘的运算法则。假设存在一个 m 维向量 x 和一个 $m \times n$ 的矩阵A，它们相乘的规则如下：

$$Ax = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \end{bmatrix}$$

不难发现，矩阵A第i行的行向量的各成分和列向量$\boldsymbol{x}$各成分分别相乘后再相加，得到的就是结果向量$\boldsymbol{Ax}$中的第i个成分。这个计算方法就是**矩阵的各行分别于向量进行点乘运算的定义式**，即：

$$Ax = \begin{bmatrix} row_1 \\ row_2 \\ \dots \\ row_m \end{bmatrix} x = \begin{bmatrix} row_1 \cdot x \\ row_2 \cdot x \\ \dots \\ row_m \cdot x \end{bmatrix}$$

从向量的角度看矩阵乘法

基于列视角的矩阵乘法

依然以一个 m 维向量 $\boldsymbol{x}$ 和一个 $m \times n$ 的矩阵 $\boldsymbol{A}$ 的乘法为例:

$$Ax = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \dots \\ a_{m1} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \dots \\ a_{m2} \end{bmatrix} + \dots + x_n \begin{bmatrix} a_{1n} \\ a_{2n} \\ \dots \\ a_{mn} \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \end{bmatrix}$$

不难发现，矩阵***A***第j列的列向量的各成分和列向量***x***对应的第j行相乘，得到的是结果向量***Ax***中的第j个成分。***Ax***相乘的运算就简化成了**原矩阵各列与向量对应位置的线性组合**，即：

$$A = [col_1, col_2, \dots, col_n] \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$$

这个结果很有趣，从列的角度来看，矩阵***A***与向量***x***的乘法，实质上对**矩阵*A***的各个列向量进行线性组合的过程，每个列向量的系数就是**向量*x***的各个对应成分。

从向量的角度看矩阵乘法

例题讲解

【例6.3】给出矩阵乘法：

$$Ax = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 3 \\ 5 \end{bmatrix} = 3 \begin{bmatrix} 1 \\ 3 \end{bmatrix} + 5 \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

通过乘法计算，最终所得到的结果向量是：

位于矩阵第一列的列向量 $\begin{bmatrix} 1 \\ 3 \end{bmatrix}$ 的3倍加上位于第二列的列向量的 $\begin{bmatrix} 2 \\ 4 \end{bmatrix}$ 的5倍。

小节

综上所述，一个矩阵和一个向量相乘的过程可以理解为：对位于原矩阵各列的列向量重新进行线性组合的过程，而在线性组合的运算过程中，结果中的各个系数就是乘法运算的列向量中所对应的各个成分。

这是一种从列的角度去看待矩阵与向量乘法的新视角，对于理解空间坐标的概念非常有帮助。

课堂互动二 Link

矩阵的应用案例

矩阵的应用案例

【例6.4】特殊矩阵的生成

生成特殊规则的矩阵，可用单列乘单行的矩阵乘法。如：

$$\mathbf{X}_{10\times 21}=\begin{bmatrix} -10 & -9 & \cdots & 0 & \cdots & 9 & 10 \\ -10 & -9 & \cdots & 0 & \cdots & 9 & 10 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ -10 & -9 & \cdots & 0 & \cdots & 9 & 10 \end{bmatrix}$$

可令 v_1 = [-10, -9, ..., 9, 10], v_2 = [1; 1; ...; 1]（10行1列），则 $X = v_2^T v_1$ 是一个10×21的矩阵。

矩阵的应用案例

【例6.5】 生产成本核算问题

某厂生产三种产品，它的成本分为三类。每一类成本中，给出生产单个产品时估计需要的量。同时给出每季度生产每种产品数量的估计。该公司希望在股东会上用一个表格展示出每一季度三类成本的数量：原料费、工资和管理费。

成本（元）	产品A	产品B	产品C
原材料	0.1	0.3	0.15
劳动	0.3	0.4	0.25
企业管理费	0.1	0.2	0.15

产品	夏	秋	冬	春
A	4000	4500	4500	4000
B	2000	2600	2400	2200
C	5800	6200	6000	6000

解：我们用矩阵的方法来考虑此问题，设产品分类成本矩阵为M，季度产量矩阵为P，则有：

$$M=\begin{bmatrix}0.1 & 0.3 & 0.15\\ 0.3 & 0.4 & 0.25\\ 0.1 & 0.2 & 0.15\end{bmatrix},\quad P=\begin{bmatrix}4000 & 4500 & 4500 & 4000\\ 2000 & 2600 & 2400 & 2200\\ 5800 & 6200 & 6000 & 6000\end{bmatrix}.$$

矩阵的应用案例

【例6.5】 生产成本核算问题

成本（元）	产品A	产品B	产品C	产品	夏	秋	冬	春
原材料	0.1	0.3	0.15	A	4000	4500	4500	4000
劳动	0.3	0.4	0.25	B	2000	2600	2400	2200
企业管理费	0.1	0.2	0.15	C	5800	6200	6000	6000

- 如果按 $Q=M*P$ 构造乘积，则MP的第一列表示夏季的成本。
  - 原料费： 0.1∗4000+0.3∗2000+0.15∗5800=1870
  - 工资： 0.3∗4000+0.4∗2000+0.25∗5800=3450
  - 管理费和其他： 0.1∗4000+0.2∗2000+0.15∗5800=1670
- MP的第二列表示秋季的成本。
  - 原料费： 0.1∗4500+0.3∗2600+0.15∗6200=2100
  - 工资: 0.3∗4500+0.4∗2600+0.25∗6200
  - 管理费和其他: 0.1∗4500+0.2∗2600+0.15∗6200
- MP的第三列和第四列表示冬季和秋季的成本。

矩阵的应用案例

【例6.5】 生产成本核算问题

成本（元）	产品A	产品B	产品C
原材料	0.1	0.3	0.15
劳动	0.3	0.4	0.25
企业管理费	0.1	0.2	0.15

产品	夏	秋	冬	春
A	4000	4500	4500	4000
B	2000	2600	2400	2200
C	5800	6200	6000	6000

MP第一行的元素表示四个季度原料的总成本，第二和第三行的元素分别表示四个季度中每一季度工资和管理的成本。每一类成本的年度总成本可由矩阵的每一行元素相加得到。每一列元素相加，即可得到每一季度的总成本。下表汇总了总成本。

成本（元）	夏	秋	冬	春	全年
原材料	1870	2160	2070	1960	8060
劳动	3450	3940	3810	3580	14780
企业管理费	1670	1900	1830	1740	7140
总成本	6990	8000	7710	7280	29980

矩阵的应用案例

【例6.5】 生产成本核算问题

```
import numpy as np
import pandas as pd

M = np.array([[0.1,0.3,0.15],[0.3,0.4,0.25],[0.1,0.2,0.15]])
P = np.array([[4000,4500,4500,4000],[2000,2600,2400,2200],[5800,6200,6000,6000]])
Q = np.dot(M,P)

# 将numpy数组转换为DataFrame
df = pd.DataFrame(Q)
# 添加行列的标题
df.columns = ['夏','秋','东','春']
df.index = ['原材料','劳动','企业管理费']
# 计算各列数据总和并作为新列添加到末尾
df['全年'] = df.apply(lambda x:x.sum(), axis=1)
# 计算各列数据总和并作为新行添加到末尾
df.loc['总成本'] = df.apply(lambda x:x.sum())

# 输出pandas数据
display(df)
```

	夏	秋	东	春	全年
原材料	1870.0	2160.0	2070.0	1960.0	8060.0
劳动	3450.0	3940.0	3810.0	3580.0	14780.0
企业管理费	1670.0	1900.0	1830.0	1740.0	7140.0
总成本	6990.0	8000.0	7710.0	7280.0	29980.0

矩阵的应用案例

【例6.6】 婚姻状况计算模型

某城镇中，有8000位已婚女性和2000位单身女性，假设每年有30%的已婚女性离婚，20%的单身女性结婚。假设所有女性的总数为一常数，试求1年后有多少已婚女性和单身女性？2年后呢？10年后呢?

解：我们可以用如下的思路构建矩阵A。

1). 矩阵A的第一行元素分别为1年后仍处于婚姻状态的已婚女性和已婚的单身女性百分比;

2). 第二行元素分别为1年后离婚的已婚女性和未婚的单身女性的百分比。则：

$$A = \begin{bmatrix} 0.7 & 0.2 \\ 0.3 & 0.8 \end{bmatrix}$$

矩阵的应用案例

【例6.6】 婚姻状况计算模型

若令 $x=\begin{bmatrix}8000\\2000\end{bmatrix}$，则1年后已婚女性和单身女性人数可以用 A 乘以x计算.

- 8000*0.7表示仍然在婚姻状态的已婚女性，2000*0.2为转变为已婚的单身女性，两者相加就是1年后已婚女性的总人数；
- 8000*0.3表示1年后离婚的已婚女性，2000*0.8表示仍然未婚的未婚女性，两者相加就是1年后未婚女性的总人数。

$$Ax=\begin{bmatrix}0.7 & 0.2\\0.3 & 0.8\end{bmatrix}\begin{bmatrix}8000\\2000\end{bmatrix}=\begin{bmatrix}6000\\4000\end{bmatrix}$$

1年后将有6000位已婚女性，4000位单身女性。

矩阵的应用案例

【例6.6】 婚姻状况计算模型

要求2年后已婚女性和单身女性的数量，即计算

$$A^2x = A(Ax)$$

$$= \begin{bmatrix} 0.7 & 0.2 \\ 0.3 & 0.8 \end{bmatrix} \begin{bmatrix} 6000 \\ 4000 \end{bmatrix}$$

$$= \begin{bmatrix} 0.7 & 0.2 \\ 0.3 & 0.8 \end{bmatrix} \begin{bmatrix} 0.7 & 0.2 \\ 0.3 & 0.8 \end{bmatrix} \begin{bmatrix} 6000 \\ 4000 \end{bmatrix} = \begin{bmatrix} 5000 \\ 5000 \end{bmatrix}$$

2年后，一半的女性将为未婚，一般的女性将为单身。

一般地，n年后已婚女性和单身女性的数量可由$A^n x$求得。

问题是，当n很大的时候，计算会变得很复杂！

矩阵的应用案例

【例6.6】 婚姻状况计算模型

按照以上的分析，我们将使用Python代码实现后续的计算，求解10年后的已婚女性和单身女性的数量。

```
import numpy as np
A = np.array([[0.7,0.2],[0.3,0.8]])
x = np.array([[8000],[2000]])

n = 8 # n=1:A^2; n=2:A^3
product = np.dot(A,x)

for i in range(n-1):
    product = np.dot(A, product)

print(product)
```

```
[[4015.625]
 [5984.375]]
```

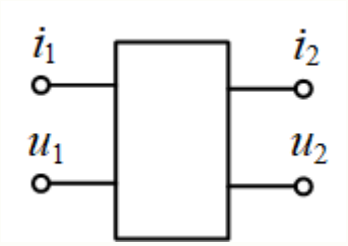
【结果分析】

由程序的输出结果可知，10年后已婚女性和诞生女性的数量分别为：4016人和5984人。

矩阵的应用案例

【例6.7】 网络的矩阵分割和连接

在电路设计中，经常要把复杂的电路分割为局部电路，每一个电路都用一个网络“黑盒子”来表示。“黑盒子”的输入为u_1，i_1，输出为u_2，i_2，都有两个变量，因此其输入输出关系用2×2矩阵***A***来表示（如右图），***A***被称为该电路的**传输矩阵**。



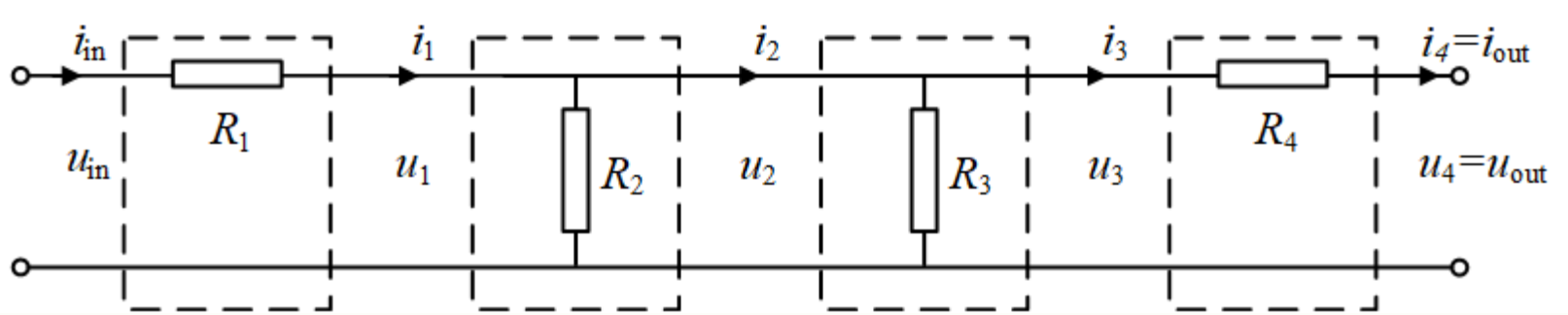
$$\begin{bmatrix} u_2 \\ i_2 \end{bmatrix} = \mathbf{A} \begin{bmatrix} u_1 \\ i_1 \end{bmatrix}$$

传输矩阵的元素可以用理论计算，也可用实验测试的方法取得。把复杂的电路分成许多串接局部电路，分别求出或测出它们的传输矩阵，再相乘起来，得到总的传输矩阵，可以使分析和测量电路的工作简化。

矩阵的应用案例

【例6.7】 网络的矩阵分割和连接

给出如下电路图，同时已知输入电流i_{in}和输入电压u_{in}，并且已知四个电阻的电阻值，求输出电流i_{out}和输出电压u_{out}。



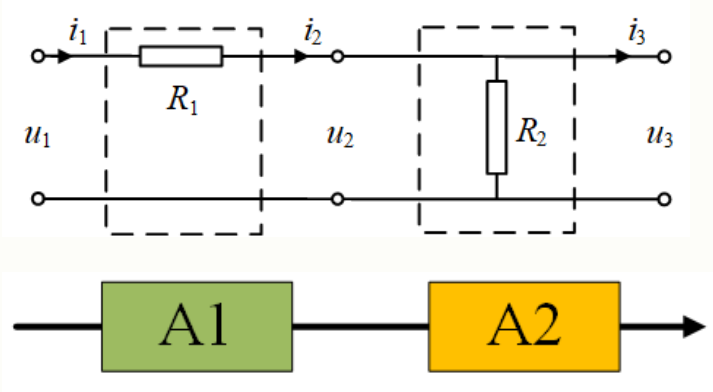
为了求输出电流、电压，我们可以将以上电路图抽象化为黑盒图，并拆分成不同电路模块的组合，再通过矩阵乘法求解最终的输出矩阵。如下图所示，其中*A1*、*A4*为串联电路，*A2*、*A3*为并联电路。：

矩阵的应用案例

【例6.7】 网络的矩阵分割和连接

首先给出一个简化版的例子：

1. 求第一个子网络（串联电路）的传输矩阵



1). 根据电路原理可以得到

$$i_2 = i_1,\ u_2 = u_1 - i_1\ R_1$$

2). 将公式转换为矩阵方程

$$\begin{bmatrix} u_2 \\ i_2 \end{bmatrix} = \begin{bmatrix} u_1 - R_1 i_1 \\ 0u_1 + i_1 \end{bmatrix} = \begin{bmatrix} 1 & -R_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ i_1 \end{bmatrix} = A_1 \begin{bmatrix} u_1 \\ i_1 \end{bmatrix}$$

3). 获得第一个子网络的传输矩阵，即单一串联电路的传输矩阵。

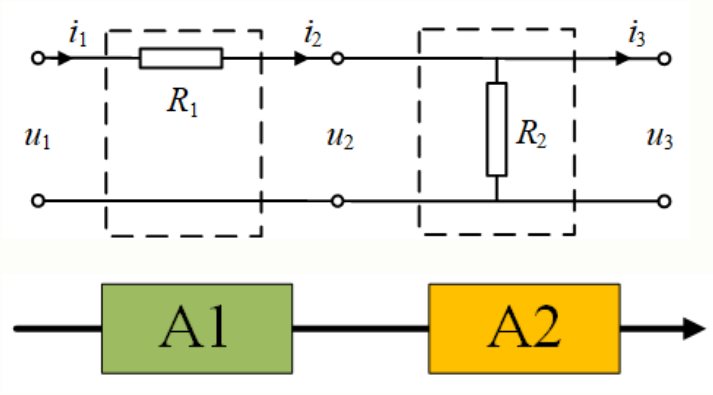
$$A_1 = \begin{bmatrix} 1 & -R_1 \\ 0 & 1 \end{bmatrix}$$

矩阵的应用案例

【例6.7】 网络的矩阵分割和连接

首先给出一个简化版的例子：

2. 求第二个子网络（并联电路）的传输矩阵



1). 根据电路原理可以得到

$$i_3 = i_2 - u_2/R_2, \quad u_3 = u_2$$

2). 将公式转换为矩阵方程

$$\begin{bmatrix} u_3 \\ i_3 \end{bmatrix} = \begin{bmatrix} u_2 + 0i_2 \\ -u_2/R_2 + i_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1/R_2 & 1 \end{bmatrix} \begin{bmatrix} u_2 \\ i_2 \end{bmatrix} = A_2 \begin{bmatrix} u_2 \\ i_2 \end{bmatrix}$$

3). 获得第二个子网络的传输矩阵，即单一并联电路的传输矩阵。

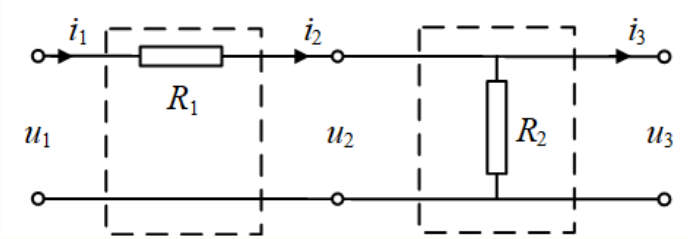
$$A_2 = \begin{bmatrix} 1 & 0 \\ -1/R_2 & 1 \end{bmatrix}$$

矩阵的应用案例

【例6.7】 网络的矩阵分割和连接

首先给出一个简化版的例子：

3. 求总电路的传输矩阵



1). 联立以上两个网络获得输入输出的关系

$$\begin{bmatrix} u_3 \\ i_3 \end{bmatrix} = A_2 \begin{bmatrix} u_2 \\ i_2 \end{bmatrix} = A_2 A_1 \begin{bmatrix} u_1 \\ i_1 \end{bmatrix}$$

2). 获得总电路传输矩阵$\boldsymbol{A}$

$$A = A_2 A_1 = \begin{bmatrix} 1 & 0 \\ -1/R_2 & 1 \end{bmatrix} \begin{bmatrix} 1 & -R_1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -R_1 \\ -1/R_2 & 1 + R_1/R_2 \end{bmatrix}$$

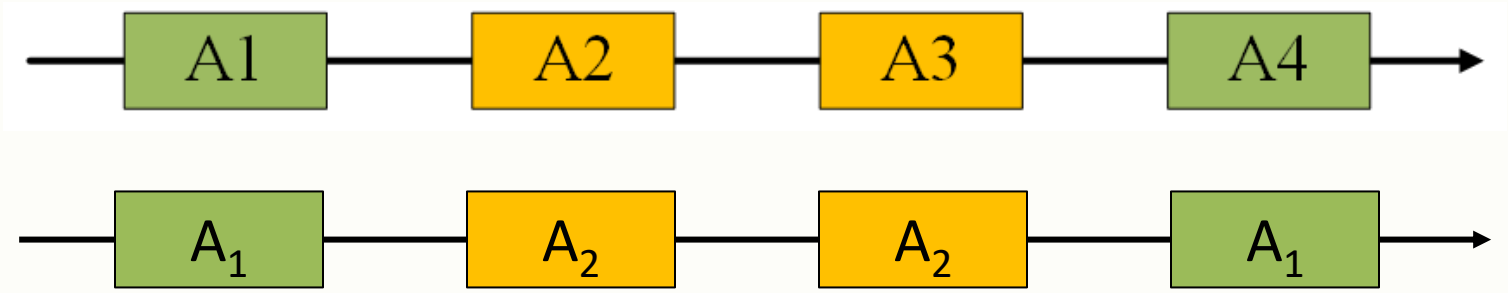
3). 利用传输矩阵求输出电压和输出电流

$$\begin{bmatrix} u_3 \\ i_3 \end{bmatrix} = A \begin{bmatrix} u_1 \\ i_1 \end{bmatrix} = \begin{bmatrix} 1 & -R_1 \\ -1/R_2 & 1 + R_1/R_2 \end{bmatrix} \begin{bmatrix} u_1 \\ i_1 \end{bmatrix}$$

矩阵的应用案例

【例6.7-总结】 网络的矩阵分割和连接

对于由单一电路构成的集成电路，我们可以使用矩阵连乘的方法进行求解，值得注意的是多个单一电路合并时的顺序与矩阵连乘的顺序密切相关。事实上，对于复杂电路也可以用类似的方法求解。



$$A = A1A2A3A3 = A_1A_2A_2\,A_1$$

$$\begin{bmatrix} u_{out} \\ i_{out} \end{bmatrix} = A \begin{bmatrix} u_{in} \\ i_{in} \end{bmatrix}$$

本例Python代码，请参看在线教案。

读万卷书 行万里路 只为最好的修炼

QQ：14777591 (宇宙骑士)

Email：ouxinyu@alumni.hust.edu.cn

Tel：18687840023