

# Basic Linear Algebra with Numpy

Hoàng-Nguyên Vũ

## 1 Giới Thiệu về NumPy

NumPy (**N**umerical **P**ython) là một thư viện mạnh mẽ trong Python được sử dụng để xử lý dữ liệu số, đặc biệt là các phép toán trên **mảng đa chiều** (NumPy array) và **các phép toán đại số tuyến tính**. NumPy giúp tối ưu tốc độ tính toán, hỗ trợ nhiều thuật toán khoa học, Machine Learning và xử lý dữ liệu.

### 1.1 Cài Đặt NumPy

Bạn có thể cài đặt NumPy bằng lệnh:

```
1 pip install numpy
```

### 1.2 Khởi Tạo Mảng NumPy

```
1 import numpy as np
2
3 # Tạo mảng 1D (vector)
4 a = np.array([1, 2, 3, 4])
5
6 # Tạo mảng 2D (ma trận)
7 b = np.array([[1, 2], [3, 4]])
8
9 # Tạo mảng số 0, số 1 và mảng ngẫu nhiên
10 zero_matrix = np.zeros((3,3))
11 ones_matrix = np.ones((2,2))
12 rand_matrix = np.random.rand(3,3)
13
14 # Tạo ma trận đơn vị
15 identity_matrix = np.eye(3)
```

## 2 Các Phép Toán Cơ Bản với NumPy

### 2.1 Phép Cộng, Trừ Hai Ma Trận

Cho hai ma trận  $A$  và  $B$  cùng kích thước  $m \times n$ :

$$C = A + B, \quad D = A - B$$

Công thức tổng quát:

$$c_{ij} = a_{ij} + b_{ij}, \quad d_{ij} = a_{ij} - b_{ij}, \quad \forall i, j$$

Ví dụ trong NumPy:

```
1 A = np.array([[1, 2], [3, 4]])
2 B = np.array([[5, 6], [7, 8]])
3
4 C = A + B # Cộng ma trận
5 D = A - B # Trừ ma trận
```

## 2.2 Phép Nhân Từng Phần Tử (Element-wise)

Nhân từng phần tử của hai ma trận:

$$E = A \odot B, \quad e_{ij} = a_{ij} \cdot b_{ij}$$

```
1 E = A * B # Nhân từng phần tử
```

## 2.3 Phép Nhân Ma Trận (Matrix Multiplication)

$$C = A \times B, \quad c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$$

Ví dụ trong NumPy:

```
1 C = A @ B # Hoặc np.dot(A, B)
```

# 3 Các Phép Toán Đại Số Tuyến Tính trong NumPy

## 3.1 Tính Định Thức Ma Trận

$$\det(A) = a_{11}a_{22} - a_{12}a_{21}$$

Ví dụ trong NumPy:

```
1 det_A = np.linalg.det(A)
```

## 3.2 Ma Trận Nghịch Đảo

Một ma trận  $A$  khả nghịch nếu tồn tại  $A^{-1}$  sao cho:

$$AA^{-1} = A^{-1}A = I$$

Công thức tính:

$$A^{-1} = \frac{1}{\det(A)} \cdot \text{adj}(A)$$

Ví dụ trong NumPy:

```
1 A_inv = np.linalg.inv(A)
```

### 3.3 Hạng của Ma Trận

Hạng của ma trận là số hàng độc lập tuyến tính:

```
1 rank_A = np.linalg.matrix_rank(A)
```

### 3.4 Giải Hệ Phương Trình Tuyến Tính

Giải hệ phương trình  $Ax = B$ :

$$x = A^{-1}B$$

Ví dụ trong NumPy:

```
1 x = np.linalg.solve(A, B)
```

### 3.5 Tính Giá Trị Riêng và Vector Riêng

$$Av = \lambda v$$

Giải phương trình đặc trưng:

$$\det(A - \lambda I) = 0$$

Ví dụ trong NumPy:

```
1 eig_values, eig_vectors = np.linalg.eig(A)
```

### 3.6 Phân Rã SVD (Singular Value Decomposition)

$$A = USV^T$$

Ví dụ trong NumPy:

```
1 U, S, V = np.linalg.svd(A)
```

## 4 Các Phép Toán Vector

### 4.1 Tích Vô Hướng (Dot Product)

$$v_1 \cdot v_2 = \sum_{i=1}^n v_{1i}v_{2i}$$

Ví dụ trong NumPy:

```
1 dot_product = np.dot(v1, v2)
```

## 4.2 Tích Có Hướng (Cross Product)

$$v_1 \times v_2 = \begin{vmatrix} i & j & k \\ v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \end{vmatrix}$$

Ví dụ trong NumPy:

```
1 cross_product = np.cross(v1, v2)
```

## 4.3 Chuẩn của Vector (Norm)

Chuẩn Euclidean:

$$\|v\| = \sqrt{\sum_i v_i^2}$$

Ví dụ trong NumPy:

```
1 norm_v = np.linalg.norm(v)
```

## 4.4 Chuyển Vị Ma Trận

$A^T$  = hoán đổi hàng và cột của  $A$

Ví dụ trong NumPy:

```
1 A_transpose = A.T
```

# 5 Kết Luận

NumPy là một thư viện mạnh mẽ cho tính toán khoa học, giúp xử lý các phép toán đại số tuyến tính nhanh chóng. Việc hiểu rõ NumPy giúp ứng dụng hiệu quả vào Machine Learning, AI, và xử lý dữ liệu.

## Bài tập: Hãy viết chương trình Python sử dụng Numpy để thực hiện các phép toán

### Phép Cộng và Trừ Ma Trận

Bài toán 1: Cho hai ma trận:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

Tính:

$$C = A + B, \quad D = A - B$$

Kết quả:

$$C = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}, \quad D = \begin{bmatrix} -4 & -4 \\ -4 & -4 \end{bmatrix}$$

Bài toán 2: Tính tổng và hiệu của:

$$C = \begin{bmatrix} 2 & -3 \\ 4 & -5 \end{bmatrix}, \quad D = \begin{bmatrix} 1 & 4 \\ -2 & 6 \end{bmatrix}$$

Kết quả:

$$C + D = \begin{bmatrix} 3 & 1 \\ 2 & 1 \end{bmatrix}, \quad C - D = \begin{bmatrix} 1 & -7 \\ 6 & -11 \end{bmatrix}$$

### Phép Nhân Hadamard

Bài toán 3: Tính nhân Hadamard của hai ma trận:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

Kết quả:

$$A \odot B = \begin{bmatrix} 5 & 12 \\ 21 & 32 \end{bmatrix}$$

Bài toán 4: Cho ma trận như bên dưới hãy tính tích Hadamard  $E \odot F$

$$E = \begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{bmatrix}, \quad F = \begin{bmatrix} 5 & 4 & 3 \\ 2 & 1 & 0 \end{bmatrix}$$

Kết quả:

$$E \odot F = \begin{bmatrix} 0 & 4 & 6 \\ 6 & 4 & 0 \end{bmatrix}$$

## Tích Vô Hướng (Dot Product)

**Bài toán 5: Cho hai vector:**

$$v_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad v_2 = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

Tính tích vô hướng:  $v_1 \cdot v_2$ .

**Kết quả:**

$$v_1 \cdot v_2 = 1(4) + 2(5) + 3(6) = 32$$

## Tích Chập (Convolution)

**Bài toán 6: Cho ma trận đầu vào và bộ lọc:**

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad K = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Thực hiện tích chập  $C = A * K$ .

**Kết quả:**

$$C = \begin{bmatrix} -4 & -4 \\ -4 & -4 \end{bmatrix}$$

## Xoay Vector Một Góc $\theta$

**Bài toán 8: Xoay vector  $v = [1, 0]$  một góc  $\theta = 45^\circ$  quanh gốc tọa độ.**

**Kết quả:**

$$v_{\text{rotated}} = [0.52532199, -0.85090352]$$