



Лекція 7. Організація наукових обчислень за допомогою пакета NumPy

Лектор – доцент, кандидат фізико-математичних наук
Кириченко Віктор Вікторович



ПЛАН ЛЕКЦІЇ

1. Встановлення пакету NumPy
2. Методи та функції пакету NumPy.
3. Створення масивів. Операції над масивами.
4. Створення матриць та числових діапазонів.
5. Дії з матрицями.



Встановлення пакету NumPy

NumPy (Numerical Python) — це бібліотека для високопродуктивних операцій з багатовимірними масивами та матрицями, яка також містить велику колекцію математичних функцій.

Команда для встановлення:

```
pip install numpy
```

Використання:

```
import numpy as np
```



Встановлення пакету NumPy

Набір високошвидкісних рутин (функцій) для обробки масивів, включаючи:

- математичні операції
- логічні операції
- зміни форми (shape manipulation)
- сортування та фільтрацію
- введення/виведення даних
- дискретні перетворення Фур'є (DFT)
- базову лінійну алгебру
- статистичні обчислення
- генерацію випадкових чисел та симуляції



Встановлення пакету NumPy

Приклад використання:

```
import numpy as np
```

```
a = np.array([[1, 2], [3, 4]])
```

```
print(a.shape)    # (2, 2)
```

```
print(a.dtype)    # int64 (залежить від платформи)
```



Методи та функції пакету NumPy

- Математичні операції: `np.sum`, `np.mean`, `np.exp`, `np.log`
- Логічні операції: `np.all`, `np.any`
- Зміна форми: `reshape`, `flatten`, `transpose`
- Сортування і вибірка: `sort`, `argsort`, `where`, `extract`
- Лінійна алгебра: `dot`, `matmul`, `inv`, `eig`, `det`
- Статистика: `mean`, `std`, `var`, `median`
- Випадкові значення: `np.random.rand`, `np.random.normal`
- Введення/виведення: `np.loadtxt`, `np.savetxt`, `np.save`,
`np.load`



Створення масивів. Операції над масивами.

Функція	Опис	Приклад
<code>np.array()</code>	Створює масив з Python-списку	<code>np.array([1, 2, 3])</code>
<code>np.zeros(shape)</code>	Масив із нулями	<code>np.zeros((2, 3))</code>
<code>np.ones(shape)</code>	Масив з одиницями	<code>np.ones((3, 3))</code>
<code>np.empty(shape)</code>	Порожній масив (випадкові значення)	<code>np.empty((2, 2))</code>
<code>np.arange(start, stop, step)</code>	Масив із кроком	<code>np.arange(0, 10, 2)</code>
<code>np.linspace(start, stop, num)</code>	Масив із рівними інтервалами	<code>np.linspace(0, 1, 5)</code>
<code>np.full(shape, value)</code>	Масив із заданим значенням	<code>np.full((2, 2), 7)</code>



Створення масивів. Операції над масивами.

Математичні функції

Функція	Опис
<code>np.add(a, b)</code>	Додавання поелементно
<code>np.subtract(a, b)</code>	Віднімання
<code>np.multiply(a, b)</code>	Множення
<code>np.divide(a, b)</code>	Ділення
<code>np.power(a, b)</code>	Піднесення до степеня
<code>np.sqrt(a)</code>	Квадратний корінь
<code>np.exp(a)</code>	Експонента
<code>np.log(a)</code>	Натуральний логарифм
<code>np.abs(a)</code>	Абсолютне значення
<code>np.round(a, n)</code>	Округлення до n знаків



Створення масивів. Операції над масивами.

Статистичні функції

Функція	Опис
<code>np.mean(a)</code>	Середнє значення
<code>np.median(a)</code>	Медіана
<code>np.std(a)</code>	Стандартне відхилення
<code>np.var(a)</code>	Дисперсія
<code>np.min(a)</code>	Мінімум
<code>np.max(a)</code>	Максимум
<code>np.sum(a)</code>	Сума елементів



Створення масивів. Операції над масивами.

Операції з формою масиву

Функція	Опис
<code>a.shape</code>	Повертає розмір масиву
<code>a.reshape(newshape)</code>	Змінює форму
<code>a.flatten()</code>	Перетворює в 1D масив
<code>a.T</code>	Транспонування
<code>np.concatenate()</code>	Об'єднання масивів
<code>np.vstack()</code>	Вертикальне об'єднання
<code>np.hstack()</code>	Горизонтальне об'єднання
<code>np.split(a, n)</code>	Розбиття масиву на n частин



Створення масивів. Операції над масивами.

Генерація випадкових чисел

```
np.random.seed(0)                # Фіксація генератора
np.random.rand(3, 2)             # Випадкові числа [0,1)
np.random.randint(1, 10, (2, 3)) # Цілі числа
np.random.normal(0, 1, (2, 2))   # Нормальний розподіл
```



Створення масивів. Операції над масивами.

Логічні операції

```
a = np.array([1, 2, 3, 4])  
  
a > 2                # маска [False, False, True, True]  
np.where(a > 2)       # індекси, де умова істинна  
np.any(a > 3)         # чи є хоча б один True  
np.all(a > 0)         # чи всі > 0
```



Створення матриць та числових діапазонів.

У NumPy "матриця" — це, зазвичай, двовимірний масив (ndarray). Також існує клас `np.matrix`, але його використання не рекомендоване у нових проєктах — краще працювати з ndarray.

Двовимірний масив (матриця)

```
A = np.array([[1, 2], [3, 4]])
```

Одинична матриця

```
I = np.eye(3)
```

Матриця з нулями

```
Z = np.zeros((2, 3))
```

Матриця з одиницями

```
O = np.ones((3, 2))
```

Матриця з випадковими числами

```
R = np.random.rand(2, 2)
```



Створення матриць та числових діапазонів.

Числовий діапазон цілих чисел.

```
np.arange(start, stop, step, dtype=None)
```

> Відкрита верхня межа (*stop* не входить), як у *range()*

Числовий діапазон нецілих чисел.

```
np.linspace(start, stop, num=50, endpoint=True,  
retstep=False, dtype=None)
```

`np.logspace()` — логарифмічний (експоненціальний) крок

`np.geomspace()` — геометрична прогресія



Дії з матрицями.

У NumPy матрицею вважається двовимірний масив типу `ndarray`.

Наприклад:

```
import numpy as np
A = np.array([[1, 2], [3, 4]])
```

Операція	Опис	Приклад
Транспонування	Заміна рядків на стовпці	<code>A.T</code>
Поелементне додавання	Сума відповідних елементів	<code>A + B</code>
Поелементне віднімання	Різниця відповідних елементів	<code>A - B</code>
Поелементне множення	Добуток відповідних елементів	<code>A * B</code>
Поелементне ділення	Частка відповідних елементів	<code>A / B</code>



Дії з матрицями.

Операція

Приклад

Обернена матриця

```
np.linalg.inv(A)
```

Визначник (детермінант)

```
np.linalg.det(A)
```

Ранг матриці

```
np.linalg.matrix_rank(A)
```

Трасування

```
np.trace(A)
```

Норма матриці

```
np.linalg.norm(A)
```

Округлення

```
np.round(A, 2)
```

Одинична матриця

```
np.identity
```

Діагональна матриця

```
np.diag([1, 2, 3])
```



ДЯКУЮ ЗА УВАГУ