

Laboratorium 1 – podstawy zastosowania NUMPY

Streszczenie

Moduł NUMPY jest zestawem narzędzi umożliwiającym zaawansowane obliczenia matematyczne. Rozszerza on możliwości PYTHON o nowe typy danych, operacje na nich oraz funkcje przyspieszające obliczenia. Jednym z podstawowych obiektów jest 'array', który stanowi kontener dla macierzy i tablic, ułatwiający m.in. obliczenia na dużych zbiorach danych.

1 CEL

Zapoznanie się z pakietem NUMPY oraz jego zastosowaniem w analizach. Ważniejsze funkcje: 'arange, linspace, random, zeros, ones, shape, reshape, sort, argsort, dot, strides'

2 Treść i zadania

Wykonaj zadania i wyświetl wyniki. Kod zapisuj w jednym pliku do wglądu dla prowadzącego.

Sprawozdanie z zajęć zawierające wykorzystane fragmenty programu, wyniki działań itp. umieść na portalu elerning w zakładce 'PiAD/Laboratoria/Lab1'. elerning

Proszę się zalogować poświadczeniami używanymi w aplikacji 'edziekanat' i potwierdzić udział w kursie.

- Tablice:

Tablica jednowymiarowa: `a=np.array([1,2,3,4,5,6,7])`

Tablica 2D: `b=np.array([[1,2,3,4,5], [6,7,8,9,10]])`

Wykonaj transpozycję tablicy 'b' za pomocą funkcji `transpose`.

Utwórz i wyświetl tablicę składającą się ze 100 elementów za pomocą funkcji `arange`.

Utwórz i wyświetl tablicę składającą się z 10 liczb w zakresie od 0 do 2. Użyj funkcji `linspace`.

Za pomocą `arange` utwórz tablicę pomiędzy wartościami od 0 do 100 i skoku wartości co 5.

- Liczby losowe:
 Za pomocą funkcji `random` utwórz tablicę z 20 liczb losowych rozkładu normalnego, zaokrąglonych do dwu miejsc po przecinku.
 Wygeneruj losowo 100 liczb całkowitych w zakresie od 1 do 1000.
 Za pomocą funkcji `'zeros'` i `'ones'` wygeneruj dwie macierze o rozmiarze 3x2.
 Utwórz macierz losową złożoną z liczb całkowitych o rozmiarze 5x5 i nadaj jej typ 32bit.
Zadanie: wygeneruj tablicę złożoną z losowo wybranych liczb dziesiętnych od 0-10 (a).
 Zamień wartości na `'integer'` i wstaw w nową tablicę (b).
 Znajdź funkcję `'numpy'`, która zaokrągli tablicę (a) do liczb całkowitych. Zamień je następnie na typ `'integer'`.
 Porównaj wyniki z a i b.
- Selekcja danych:
`b=np.array([[1,2,3,4,5], [6,7,8,9,10]],dtype=np.int32)`
 za pomocą funkcji `'ndim'` sprawdź ile wymiarów ma tablica b.
 za pomocą `'size'`, sprawdź z ilu elementów składa się tablica b.
 Wybierz wartości 2 i 4 z tablicy b.
 Wybierz pierwszy wiersz tablicy b.
 Wybierz wszystkie wiersze z kolumny 1.
 Wygeneruj macierz losową o rozmiarze 20x7, złożoną liczb całkowitych w przedziale 0-100. Wyświetl wszystkie wiersze dla czterech pierwszych kolumn.
- Operacje matematyczne i logiczne:
 Stwórz dwie macierze w przedziale 0-10 o rozmiarach 3x3 (a i b). Dodaj, pomnóż, podziel, spotęguj je przez siebie.
 Sprawdź czy wartość macierzy a jest większa lub równa 4.
 Sprawdź czy wartość macierzy a $1_i = j=4$.
 Znajdź funkcję w numpy do obliczenia sumy głównej przekątnej macierzy b.
- Dane statystyczne:
 Oblicz sumę, wartość minimum, maksimum, odchylenie standardowe w macierzy b. Oblicz średnią dla wierszy w macierzy b. Oblicz średnią dla kolumn macierzy b.
- Rzutowanie wymiarów za pomocą `shape` lub `resize`:
 Utwórz tablicę składającą się z 50 liczb.
 Za pomocą funkcji `'reshape'` utwórz macierz 10x5. To samo za pomocą `resize`.

Sprawdź do czego służy komenda `ravel`.

Stwórz dwie tablice o rozmiarach 5 i 4 i dodaj je do siebie. Sprawdź do czego służy funkcja `'NEWAXIS'` i wykorzystaj ją.

- Sortowanie danych:

Sprawdź składnię funkcji `sort` i `argsort`.

`a=np.random.randn(5,5)`.

Posortuj wiersze rosnąco. Posortuj kolumny malejąco.

`b=np.array([(1,'MZ','mazowieckie'),(2,'ZP','zachodniopomorskie'),(3,'ML','małopolskie')])`

Zadanie: Na podstawie powyższej tablicy zrób macierz 3x3. Posortuj dane rosnąco po kolumnie 2.

Wyświetl nazwę województwa zachodniopomorskiego.

3 Zadania podsumowujące

1. Utwórz macierz składającą się z pięciu kolumn i 10 wierszy losowo wybranych liczb całkowitych i policz sumę głównej przekątnej macierzy, używając funkcji `'trace'` a następnie wyświetl wartości używając funkcji `'diag'`.
2. Utwórz dwie tablice z losowo wybranych liczb dziesiętnych z rozkładu normalnego i przemnoż je przez siebie.
3. Utwórz dwie tablice z losowo wybranych liczb całkowitych w zakresie od 1-100. Stwórz z nich macierze o 5 kolumnach i dodaj je do siebie.
4. Stwórz macierz o 5 kolumnach i 4 wierszach oraz 4 kolumnach i 5 wierszach i dodaj je do siebie używając transformacji wymiarów.
5. Pomnoż kolumny 3 i 4, stworzonych przez siebie macierzy.
6. Wygeneruj dwie macierze o rozkładzie normalnym (`np.random.normal`) i jednostajnym (`np.random.uniform`). Policz wartość średnią, odchylenie standardowe, wariancję itp. Porównaj wyniki z obu zbiorów danych.zobacz
7. Wygeneruj dwie macierze kwadratowe `a` i `b`, pomnoż je przez siebie używając (`a*b`) oraz funkcji `dot`.zobacz Jaka jest różnica? Napisz kie-

dy warto wykorzystać funkcję dot?

8. Sprawdź funkcję strides oraz as_strided. Zastosuj je do wyboru danych z macierzy np. 5 kolumn z trzech pierwszych wierszy.
9. Wygeneruj dwie tablice a i b. Połącz je z użyciem funkcji vstack i hstack. Czym one się różnią? Zastanów się i napisz, w jakich przypadkach warto je zastosować?
10. Użyj funkcji strides i as_strided do obliczenia wartości maksymalnej bloków danych z macierzy (rysunek)

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23

Korzystano:

<https://numpy.org/>

<https://github.com/yongtwang/engineering-python>