

NumPy

Akitaka Matsuo

NumPy

- Numerical Python
- NumPy Arrays
 - Foundation for Python computation
 - Fast vectorized computation (avoiding loop)
 - Could be extended to higher dimensions
 - Efficient operations
 - Conditional logic

1D ARRAY

1d array, construction

- To construct ndarrays, we can:
 - Convert lists to an array
 - `array_a = np.array([0, 1, 2])`
 - Use various generators
 - `np.zeros()`, `np.ones()`
 - `np.random.*()`
 - `np.random.rand()`, `np.random.randn()`, `np.random.randint()`
 - `np.linspace()`

1d array, indexing

- There are quite many ways to access the elements in ndarray
 - Using “:”
 - Same as basic python lists
 - Boolean logic
 - Select elements with conditional statement
 - Fancy indexing
 - Giving a list to select the elements

Demo 1d array

- Generate different types of arrays
 - From list
 - Use
 - `np.zeros()`, `np.ones()`
 - `np.random.[methods]()`
- Working with arrays
 - Index slicing
 - Normal
 - Boolean
 - Fancy indexing

2D ARRAY

2d array, construction, indexing

- 2d array: a matrix-like object in Numpy
- Construction of two-dimensional array
 - Convert from list
 - Use various generators
 - Reshape one dimensional array
- Also, indexing works the same as the 1d array
 - Index slicing
 - Fancy indexing
 - Boolean indexing

NumPy array operations

- There are some functions for ndarrays
 - Arithmetics:
 - $+$, $-$, $*$, $/$
 - Broadcasting
 - Functions:
 - `np.mean()`
 - `np.sqrt()`
 - `np.max()`
 - Functions work for entire array or row-wise, column-wise

Demo 2d array

- Generate different types of arrays
 - From list
 - Use
 - `np.zeros()`, `np.ones()`
 - `np.random.[methods]()`
- Operations
 - Index slicing
 - Arithmetic operations
 - Functions