NUMPY

What is NumPy?

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

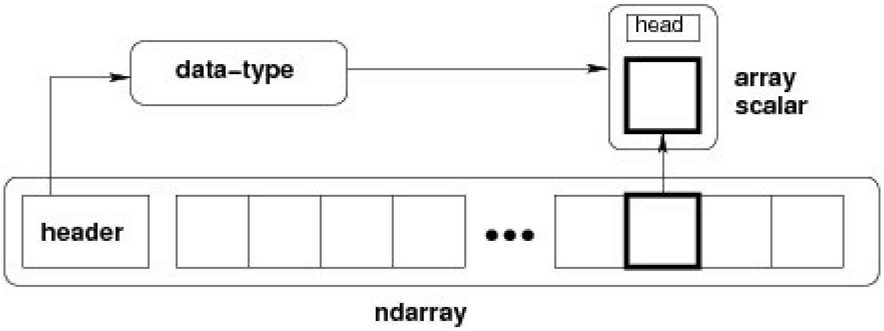
- NumPy was created in 2005 by Travis Oliphant.
- NumPy stands for Numerical Python.
- It is an open source project and you can use it freely.

Why use NumPy?

- In Python we have lists that serve the purpose of arrays, but they are slow to process.
- NumPy aims to provide an array object that is up to 50x faster than traditional python lists.

The array object in NumPy is called

`ndarray`



- The source code for NumPy is located at this github repository https://github.com/numpy/numpy

Why is NumPy Faster Than Lists?

- NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently.
- This behavior is called locality of reference in computer science.
- NumPy is a Python library and is written partially in Python, but most of the parts that require fast computation are written in C or C++.

Numpy Installation

Mac and Linux users can install NumPy via pip command

```
pip install numpy
```

1. To install NumPy, run the following command.

```
Python setup.py install
```

1. To test whether NumPy module is properly installed, try to import it from Python prompt.

```
import numpy
```

1. Alternatively, NumPy package is imported using the following syntax – `import numpy as np

Array Creation

An instance of ndarray class can be constructed by different array creation routines described later in the tutorial. The basic ndarray is created using an array function in NumPy as follows –

```
numpy.array
```

Example:

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

```
[1 2 3]
```

NumPy - Data Types

1. bool_ :- Boolean stored as a byte
2. int_ :- Default integer type

3. float_ :- Shorthand for float64
4. complex_ :- Shorthand for complex128

- A dtype object is constructed using the following syntax –

```
numpy.dtype(object, align, copy)
```

What Makes NumPy So Good?

NumPy has a syntax which is simultaneously compact, powerful and expressive. It allows users to manage data in vectors, matrices and higher dimensional arrays. Within those data structures, it allows users to:

- Access
- Manipulate
- Compute

NumPy Foundations

1. Shape :- Every array has a shape. The shape is defined by (n,m) with n the number of rows, and m the number of columns.

Example:

```
This array is of shape (2,2)
[1,2
 3,4]
```

1. Strides :- Strides describe the number of bytes required to move forward in memory.

Operations Using NumPy

1. Indexing
2. Slicing
3. Copying
4. Vectorization
5. Broadcasting
6. Reduction