

# Numpy实验楼报告

原创

Best99317 于 2018-07-07 14:18:35 发布 1993 收藏 3

分类专栏: [Python Numpy 机器学习](#) 文章标签: [Python Numpy 机器学习](#) [编程语言](#) [实验楼](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/weixin\\_36531862/article/details/79112360](https://blog.csdn.net/weixin_36531862/article/details/79112360)

版权



[Python](#) 同时被 3 个专栏收录

2 篇文章 0 订阅

订阅专栏



[Numpy](#)

1 篇文章 0 订阅

订阅专栏



[机器学习](#)

1 篇文章 0 订阅

订阅专栏

## 【引言】

最近入门Machine Learning, 再看Peter Harrington的《Machine Learning in Action》, 发现Numpy真的是Python语言里非常重要的一个库, 尤其在机器学习算法的编程中有广泛的应用。于是笔者到“实验楼”上去学习了一波, 下面吧连接、Numpy的介绍和实验报告分享给大家

## 【目录】

- 1.引言
- 2.目录
- 3.Numpy简介
- 4.Numpy数值类型

## 【Numpy 简介】

Numpy 的英文全称为 Numerical Python, 指Python 面向数值计算的第三方库。Numpy 的特点在于, 针对 Python 内建的数组类型做了扩充, 支持更高维度的数组和矩阵运算, 以及更丰富的数学函数。Numpy 是 Scipy.org 中最重要的库之一, 它同时也被 Pandas, Matplotlib 等我们熟知的第三方库作为核心计算库。

NumPy (Numeric Python) 提供了许多高级的数值编程工具, 如: 矩阵数据类型、矢量处理, 以及精密的运算库。专为进行严格的数字处理而产生。多为很多大型金融公司使用, 以及核心的科学计算组织如: Lawrence Livermore, NASA用其处理一些本来使用C++, Fortran或Matlab等所做的任务。

Numpy包括了: 1、一个强大的N维数组对象Array; 2、比较成熟的(广播)函数库; 3、用于整合C/C++和Fortran代码的工具包; 4、实用的线性代数、傅里叶变换和随机数生成函数。Numpy和稀疏矩阵运算包scipy配合使用更加方便。(参考连接: [实验楼Numpy使用教程](#))

## 【Numpy 数值类型】

1. **bool** : 布尔类型, 1 个字节, 值为 True 或 False。
2. **int** : 整数类型, 通常为 int64 或 int32 。
3. **intc** : 与 C 里的 int 相同, 通常为 int32 或 int64。
4. **intp** : 用于索引, 通常为 int32 或 int64。
5. **int8** : 字节 (从 -128 到 127 )
6. **int16** : 整数 (从 -32768 到 32767 )
7. **int32** : 整数 (从 -2147483648 到 2147483647 )
8. **int64** : 整数 (从 -9223372036854775808 到 9223372036854775807 )
9. **uint8** : 无符号整数 (从 0 到 255 )
10. **uint16** : 无符号整数 (从 0 到 65535 ) [http://blog.csdn.net/weixin\\_36531862](http://blog.csdn.net/weixin_36531862)
11. **uint32** : 无符号整数 (从 0 到 4294967295 )
12. **uint64** : 无符号整数 (从 0 到 18446744073709551615 )
13. **float** : float64 的简写。
14. **float16** : 半精度浮点, 5 位指数, 10 位尾数
15. **float32** : 单精度浮点, 8 位指数, 23 位尾数
16. **float64** : 双精度浮点, 11 位指数, 52 位尾数
17. **complex** : complex128 的简写。
18. **complex64** : 复数, 由两个 32 位浮点表示。
19. **complex128** : 复数, 由两个 64 位浮点表示。[http://blog.csdn.net/weixin\\_36531862](http://blog.csdn.net/weixin_36531862)

## 【Numpy中的对象及函数】

### 1.dtype()

我们可以用 `numpy.dtype(object, align, copy)` 来指定数值类型。而在数组里面, 可以用 `dtype=` 参数。

```

Terminal 终端 - python
[10:13:23]
ish: command not found: import
shyanlou:~/ $ python
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
Type "help", "copyright", "credits" or "license()" for more information.
>>> import numpy as np
>>> a = np.array([1.1, 2.2, 3.3], dtype = np.float64)
>>> a
array([1.1, 2.2, 3.3])
>>> a.astype(int)
array([1, 2, 3])
>>> a.dtype
dtype('float64')
>>>
  
```

## 2.多维数组

### 2.1 ndarray

#### 2.1.1 ndarray的简介

numpy有2种基本对象, ndarray (N-dimensional array object) 和 ufunc (universal function object) ndarray是存储单一数据类型的数据的多维数组。

Python的内建对象数组可以有三种形式:

- list 列表: [1, 2, 3]
- Tuple 元组: (1, 2, 3, 4, 5)
- Dict 字典: {A:1, B:2}

列表为大家所熟知, 元组与列表相似, 区别在于元组里的值无法修改。在此着重介绍一下字典。字典是另一种可变容器模型, 且可存储任意类型对象。字典对象由键和值组成, 每个键值(key=>value)对用冒号(:)分割, 每个对之间用逗号(,)分割, 整个字典包括在花括号({})中, 格式如下所示:

```
d = {key1 : value1, key2 : value2 }
```

值可以取任何数据类型, 但键必须是不可变的, 如字符串, 数字或元组。一个简单的字典实例:

```
dict = {'Alice': '2341', 'Beth': '9102', 'Cecil': '3258'}
```

Numpy 最核心且最重要的一个特性就是 ndarray 多维数组对象, 它区别于 python 的标准类, 拥有对高维数组的处理能力, 这也是数值计算过程中缺一不可的重要特性。

Numpy 中, ndarray 类具有六个参数, 它们分别为:

- **shape**: 数组的形状。
- **dtype**: 数据类型。
- **buffer**: 对象暴露缓冲区接口。
- **offset**: 数组数据的偏移量。
- **strides**: 数据步长。

- order: {'C', 'F'}, 以行或列为主排列顺序。

### 2.1.2 ndarray 的创建方法

在 numpy 中，主要通过以下 5 种途径创建数组，它们分别是：

- 从 Python 数组结构列表，元组等转换。
- 使用 np.arange、np.ones、np.zeros 等 numpy 原生方法。
- 从存储空间读取数组。
- 通过使用字符串或缓冲区从原始字节创建数组。
- 使用特殊函数，如 random。

## 2.2 列表或元组向 ndarray 的转换

### 2.2.1 转换方法（函数）

```
numpy.array(object, dtype=None, copy=True, order=None, subok=False, ndmin=0)
```

其中，参数：

- object: 列表、元组等。
- dtype: 数据类型。如果未给出，则类型为被保存对象所需的最小类型。
- copy: 布尔来写，默认 True，表示复制对象。
- order: 顺序。
- subok: 布尔类型，表示子类是否被传递。
- ndmin: 生成的数组应具有的最小维数。

### 2.2.2 实验过程

```
>>> import numpy as np
3],[1,2,3]])[[[1,2,3],[1,3,3],[1,2,3]],[[1,2,3],[1,2,3],[1,2,3]],[[1,2,3],[1,2,
array([[1, 2, 3],
        [1, 3, 3],
        [1, 2, 3]],

        [[1, 2, 3],
         [1, 2, 3],
         [1, 2, 3]],

        [[1, 2, 3],
         [1, 2, 3],
         [1, 2, 3]])
>>> np.array([(1,2),(3,4),(5,6)])
array([[1, 2],
       [3, 4],
       [5, 6]])
>>>
```

[http://blog.csdn.net/weixin\\_36531862](http://blog.csdn.net/weixin_36531862)

## 2.3 arrange()方法创建规律性多维数组

arrange()方法是在一定给定区间内创建一些列等间隔的值，方法如下：

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

先设置值所在的区间，这里为 [开始, 停止)，你应该能发现这是一个半开半闭区间。然后，在设置 step 步长用于设置值之间的间隔。最后的可选参数 dtype 可以设置返回 ndarray 的值类型。例如：

```
>>> np.arange(3,7,0.5,dtype='float32')
array([3. , 3.5, 4. , 4.5, 5. , 5.5, 6. , 6.5], dtype=float32)
>>>
```

[http://blog.csdn.net/weixin\\_36531862](http://blog.csdn.net/weixin_36531862)

