



1. 与 NumPy 交互

1.1 初始化

```
import numpy as np
a = np.array([1, 2, 3])
b = np.array([(1+5j, 2j, 3j), (4j, 5j, 6j)])
c = np.array([(1.5, 2, 3), (4, 5, 6)], [(3, 2, 1), (4, 5, 6)])
```

1.2 索引技巧

```
np.mgrid[0:5, 0:5] # 创建稠密栅格
np.ogrid[0:2, 0:2] # 创建开放栅格
# 按行纵向堆叠数组
np.r_[3, [0]*5, -1:1:10j]
np.c_[a, a] # 按列横向堆叠数组
```

1.3 操控形状

```
np.transpose(b) # 转置矩阵
b.flatten() # 拉平数组
np.hstack((c, c)) # 按列横向堆叠数组
np.vstack((a, b)) # 按行纵向堆叠数组
np.hsplit(c, 2) # 在索引 2 横向分割数组
np.vsplit(c, 2) # 在索引 2 纵向分割数组
```

1.4 多项式

```
from numpy import poly1d
p = poly1d([3, 4, 5]) # 创建多项式对象
```

1.5 矢量函数

```
def myfunc(a):
    if a < 0:
        return a*2
    else:
        return a/2
np.vectorize(myfunc) # 矢量函数
```

1.6 类型控制

```
np.real(c) # 返回数组元素的实部
np.imag(c) # 返回数组元素的虚部
# 如果复数接近 0, 返回实部
np.real_if_close(c, tol=1000)
np.cast['f'](np.pi) # 将对象转化为数据类型
```

1.7 常用函数

```
np.angle(b, deg=True) # 返回复数的辐角
# 创建等差数组 (样本数)
g = np.linspace(0, np.pi, num=5)
g[3:] += np.pi
np.unwrap(g) # 解包
# 创建等差数组 (对数刻度)
np.logspace(0, 10, 3)
np.select([c<4], [c*2])
# 根据条件返回数组列表的值
from scipy import special
special.factorial(a) # 因子
# 计算排列组合  $C_{10}^3$ 
import scipy
scipy.special.comb(10, 3, exact=True)
# NP 点中心导数的权重
from scipy import misc
misc.central_diff_weights(3)
# 查找函数在某点的第 n 个导数
misc.derivative(myfunc, 1.0)
```

2. 线性代数

使用 linalg 和 sparse 模块。

注意 scipy.linalg 包含了 numpy.linalg, 并扩展了其功能。

```
from scipy import linalg, sparse
```

2.1 创建矩阵

```
a = np.matrix(np.random.random((2, 2)))
b = np.asmatrix(b)
c = np.mat(np.random.random((10, 5)))
d = np.mat([[3, 4], [5, 6]])
```

2.2 基础矩阵操作

逆矩阵

```
a.I # 求逆矩阵
linalg.inv(A) # 求逆矩阵
a.T # 矩阵转置
a.H # 共轭转置
np.trace(A) # 计算对角线元素的和
```

范数

```
linalg.norm(A) # Frobeniu 范数
linalg.norm(A, 1) # L1 范数 (最大列汇总)
linalg.norm(A, np.inf) # L 范数 (最大列汇总)
```

排名

```
np.linalg.matrix_rank(C) # 矩阵排名
```

行列式

```
linalg.det(A) # 行列式
```

求解线性问题

```
linalg.solve(A, b) # 求解稠密矩阵
E = np.mat(a).T # 求解稠密矩阵
linalg.lstsq(D, D) # 最小二乘法求解线性方程
```

广义逆

```
linalg.pinv(C) # 计算矩阵的伪逆 (最小二乘求解器)
linalg.pinv2(C) # 计算矩阵的伪逆 (SVD)
```

2.3 创建稀疏矩阵

```
f = np.eye(3, k=1) # 创建 2X2 单位矩阵
g = np.mat(np.identity(2)) # 同上
c[C>0.5] = 0
```

```
h = sparse.csr_matrix(C) # 压缩稀疏行矩阵
i = sparse.csc_matrix(D) # 压缩稀疏列矩阵
```

```
J = sparse.dok_matrix(A) # DOK 矩阵
i.todense() # 将稀疏矩阵转为全矩阵
sparse.isspmatrix_csc(A) # 单位稀疏矩阵
```

2.4 稀疏矩阵操作

逆矩阵

```
import scipy.sparse.linalg as linalg
linalg.inv(I) # 求逆矩阵
```

范数

```
linalg.norm(I) # 范数
```

解决线性问题

```
linalg.spsolve(I, I) # 稀疏求解矩阵
```

2.5 稀疏矩阵函数

```
sparse.linalg.expm(I) # 稀疏矩阵指数
```

2.6 矩阵函数

加法

`np.add(A, D)` # 加法

减法

`np.subtract(A, D)` # 减法

除法

`np.divide(A, D)` # 除法

乘法

`np.multiply(D, A)` # 乘法

`np.dot(A, D)` # 点积

`np.vdot(A, D)` # 向量点积

`np.inner(A, D)` # 内积

`np.outer(A, D)` # 外积

`np.tensordot(A, D)` # 张量点积

`np.kron(A, D)` # Kronecker 积

指数函数

`linalg.expm(A)` # 矩阵指数

对数函数

`scipy.linalg.logm(A)` # 矩阵对数

三角函数

`scipy.linalg.sinm(D)` # 矩阵正弦

`scipy.linalg.cosm(D)` # 矩阵余弦

`scipy.linalg.tanm(A)` # 矩阵切线

双曲三角函数

`scipy.linalg.sinhm(D)` # 双曲矩阵正弦

`scipy.linalg.coshm(D)` # 双曲矩阵余弦

`scipy.linalg.tanhm(A)` # 双曲矩阵切线

矩阵符号函数

`np.sign(A)` # 矩阵符号函数

矩阵平方根

`scipy.linalg.sqrtn(A)` # 矩阵平方根

任意函数

评估矩阵函数

`scipy.linalg.funm(A, lambda x: x*x)`

2.7 矩阵分解

特征值与特征向量

求解方阵的普通或广义特征值问题

`la, v = scipy.linalg.eig(A)`

解包特征值

`l1, l2 = la`

第一个特征值

`v[:, 0]`

第二个特征值

`v[:, 1]`

解包特征值

`scipy.linalg.eigvals(A)`

奇异值分解 (SVD)

`U, s, Vh = scipy.linalg.svd(B)`

`m, N = B.shape`

在 SVD 中构建 Sigma 矩阵

`sig = scipy.linalg.diagsvd(s, M, N)`

LU 分解

`p, L, U = scipy.linalg.lu(C)`

解构稀疏矩阵

特征值与特征向量

`la, v = sparse.linalg.eigs(F, 1)`

奇异值分解 (SVD)

`sparse.linalg.svds(H, 2)`

调用帮助

help 函数

`help(scipy.linalg.diagsvd)`

`np.info(np.matrix)`



SciPy 是著名的 python 开源科学计算库。SciPy 构建于 NumPy 之上进行科学计算，统计分析。SciPy 提供了许多科学计算的库函数，如线性代数、微分方程、信号处理、图像处理、系数矩阵计算等。



扫码回复“数据科学”

下载最新全套速查表

SciPy 速查表

获取最新版 | <http://www.showmeai.tech/>

作者 | 韩信子 @ShowMeAI

设计 | 南乔 @ShowMeAI

参考 | DataCamp Cheatsheet



数据科学工具库速查表



NumPy 是 Python 数据科学计算的核心库，提供了高性能多维数组对象及处理数组的工具。使用以下语句导入 NumPy 库：

```
import numpy as np
```



SciPy 是基于 NumPy 创建的 Python 科学计算核心库，提供了众多数学算法与函数。



Pandas 是基于 NumPy 创建的 Python 库，为 Python 提供了易于使用的数据结构和数据分析工具。使用以下语句导入：

```
import pandas as pd
```



Matplotlib 是 Python 的二维绘图库，用于生成符合出版质量或跨平台交互环境的各类图形。

```
import matplotlib.pyplot as plt
```



Seaborn 是基于 matplotlib 开发的高阶 Python 数据可视图库，用于绘制优雅、美观的统计图形。使用下列别名导入该库：

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```



Bokeh 是 Python 的交互式可视图库，用于生成在浏览器里显示的大规模数据集高性能可视图。Bokeh 的中间层通用 **bokeh.plotting** 界面主要为两个组件：数据与图示例。

```
from bokeh.plotting import figure
```

```
from bokeh.io import output_file, show
```



PySpark 是 Spark 的 Python API，允许 Python 调用 Spark 编程模型。Spark SQL 是 Apache Spark 处理结构化数据模块。

AI 垂直领域工具库速查表



Scikit-learn 是开源的 Python 库，通过统一的界面实现机器学习、预处理、交叉验证及可视化算法。



Keras 是强大、易用的深度学习库，基于 Theano 和 TensorFlow 提供了高阶神经网络 API，用于开发和评估深度学习模型。



“TensorFlow™ is an open source software library for numerical computation using data flow graphs.” **TensorFlow** 是 Google 公司开发的机器学习架构，兼顾灵活性和扩展性，既适合用于工业生产也适合用于科学研究。



PyTorch 是 Facebook 团队 2017 年初发布的深度学习框架，有利于研究人员、爱好者、小规模项目等快速搞出原型。**PyTorch** 也是 Python 程序员最容易上手的深度学习框架。



Hugging Face 以开源的 NLP 预训练模型库 **Transformers** 而广为人知，目前 GitHub Star 已超过 54000+。**Transformers** 提供 100+ 种语言的 32 种预训练语言模型，简单，强大，高性能，是新手入门的不二选择。



OpenCV 是一个跨平台计算机视觉库，由 C 函数 /C++ 类构成，提供了 Python、MATLAB 等语言的接口。**OpenCV** 实现了图像处理和计算机视觉领域的很多通用算法。

编程语言速查表



SQL 是管理关系数据库的结构化查询语言，包括数据的增删查改等。作为数据分析的必备技能、岗位 JD 的重要关键词，SQL 是技术及相关岗位同学一定要掌握的语言。



Python 编程语言简洁快速、入门简单且功能强大，拥有丰富的第三方库，已经成为大数据和人工智能领域的主流编程语言。

More...

AI 知识技能速查表



Jupyter Notebook 交互式计算环境，支持运行 40+ 种编程语言，可以用来编写漂亮的交互式文档。这个教程把常用的基础功能讲解得很清楚，对新手非常友好。



正则表达式 非常强大，能匹配很多规则的文本，常用于文本提取和爬虫处理。这也是一门令人难以捉摸的语言，字母、数字和符号堆在一起，像极了“火星文”。

More...



ShowMeAI 速查表 (©2021)

获取最新版 | <http://www.showmeai.tech/>

作者 | 韩信子 @ShowMeAI

设计 | 南乔 @ShowMeAI

数据科学工具库速查表

扫码回复“数据科学”

获取最新全套速查表

AI 垂直领域工具库速查表

扫码回复“工具库”

获取最新全套速查表

编程语言速查表

扫码回复“编程语言”

获取最新全套速查表

AI 知识技能速查表

扫码回复“知识技能”

获取最新全套速查表