# 深度学习时用到的那些 numpy 函数

##### [numpy.median](https://docs.scipy.org/doc/numpy/reference/generated/numpy.median.html)

* 求中位数，如果数组有奇数个数，返回中位数，如果数据有偶数个数，返回中间两个数的平均值
* >>> a = np.array([[10, 7, 4], [3, 2, 1]])  
  >>> a  
  array([[10, 7, 4],  
   [ 3, 2, 1]])  
  >>> np.median(a)  
  3.5  
  >>> np.median(a, axis=0)  
  array([6.5, 4.5, 2.5])  
  >>> np.median(a, axis=1)  
  array([7., 2.])

##### [numpy.load](https://docs.scipy.org/doc/numpy/reference/generated/numpy.load.html)

* 从 npy,npz文件中载入数组
* >>> np.save('/tmp/123', np.array([[1, 2, 3], [4, 5, 6]]))  
  >>> np.load('/tmp/123.npy')  
  array([[1, 2, 3],  
   [4, 5, 6]])

##### [numpy.concatenate](https://docs.scipy.org/doc/numpy/reference/generated/numpy.concatenate.html)

连接一系列的数组

>>> a = np.array([[1, 2], [3, 4]])  
>>> b = np.array([[5, 6]])  
>>> np.concatenate((a, b), axis=0)  
array([[1, 2],  
 [3, 4],  
 [5, 6]])  
>>> np.concatenate((a, b.T), axis=1)  
array([[1, 2, 5],  
 [3, 4, 6]])  
>>> np.concatenate((a, b), axis=None)  
array([1, 2, 3, 4, 5, 6])

##### [numpy.zeros](https://docs.scipy.org/doc/numpy/reference/generated/numpy.zeros.html)

* 返回一个特定形状的数组，用0填满
* >>> np.zeros(5)  
  array([ 0., 0., 0., 0., 0.])  
    
  >>> np.zeros((5,), dtype=int)  
  array([0, 0, 0, 0, 0])

##### [numpy.ones](https://docs.scipy.org/doc/numpy/reference/generated/numpy.ones.html)

返回一个特定形状的数组，用1填满

>>> np.ones(5)  
array([1., 1., 1., 1., 1.])  
  
>>> np.ones((5,), dtype=int)  
array([1, 1, 1, 1, 1])

##### [numpy.full](https://docs.scipy.org/doc/numpy/reference/generated/numpy.full.html)

* 返回一个特定形状的数组，用指定值填满。
* >>> np.full((2, 2), np.inf)  
  array([[inf, inf],  
   [inf, inf]])  
  >>> np.full((2, 2), 10)  
  array([[10, 10],  
   [10, 10]])

##### [numpy.reshape](https://docs.scipy.org/doc/numpy/reference/generated/numpy.resharp.html)

把数组变换成指定形状. 参考： [python基础之numpy.reshape详解](https://www.jianshu.com/p/fc2fe026f002)

>>> a = np.array([[1,2,3], [4,5,6]])  
>>> np.reshape(a, 6)  
array([1, 2, 3, 4, 5, 6])  
>>> np.reshape(a, 6, order='F')  
array([1, 4, 2, 5, 3, 6])  
>>>  
>>> np.reshape(a, (3,-1)) # the unspecified value is inferred to be 2  
array([[1, 2],  
 [3, 4],  
 [5, 6]])

##### [numpy.dot](https://docs.scipy.org/doc/numpy/reference/generated/numpy.dot.html)

* 两个数组相乘
* >>> np.dot(3, 4)  
  12  
    
  >>> np.dot([2j, 3j], [2j, 3j])  
  (-13+0j)  
    
  >>> a = [[1, 0], [0, 1]]  
  >>> b = [[4, 1], [2, 2]]  
  >>> np.dot(a, b)  
  array([[4, 1],  
   [2, 2]])

##### [numpy.log](https://docs.scipy.org/doc/numpy/reference/generated/numpy.log.html)

* 求对数
* >>> np.log([1, np.e, np.e\*\*2, 0])  
  array([ 0., 1., 2., -Inf])

##### [numpy.sum](https://docs.scipy.org/doc/numpy/reference/generated/numpy.sum.html)

把数据按照指定的轴求和

>>> np.sum([0.5, 1.5])  
2.0  
>>> np.sum([0.5, 0.7, 0.2, 1.5], dtype=np.int32)  
1  
>>> np.sum([[0, 1], [0, 5]])  
6  
>>> np.sum([[0, 1], [0, 5]], axis=0)  
array([0, 6])  
>>> np.sum([[0, 1], [0, 5]], axis=1)  
array([1, 5])  
>>> np.sum([[0, 1], [np.nan, 5]], where=[False, True], axis=1)  
array([1., 5.])

1. [numpy.mean](https://docs.scipy.org/doc/numpy/reference/generated/numpy.mean.html)

* 按照轴计算算术平均值
* >>> a = np.array([[1, 2], [3, 4]])  
  >>> np.mean(a)  
  2.5  
  >>> np.mean(a, axis=0)  
  array([2., 3.])  
  >>> np.mean(a, axis=1)  
  array([1.5, 3.5])

1. [numpy.absolute](https://docs.scipy.org/doc/numpy/reference/generated/numpy.absolute.html)

为每个元素计算绝对值。如果是向量比如a+ib，绝对值是fig:.

>>> x = np.array([-1.2, 1.2])  
>>> np.absolute(x)  
array([ 1.2, 1.2])  
>>> np.absolute(1.2 + 1j)  
1.5620499351813308

1. [numpy.random.rand](https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.rand.html)

创建一个指定形状的数组，并且填充随机数

>>> np.random.rand(3,2)  
array([[ 0.14022471, 0.96360618], #random  
 [ 0.37601032, 0.25528411], #random  
 [ 0.49313049, 0.94909878]]) #random

14.[numpy.tanh](https://docs.scipy.org/doc/numpy/reference/generated/numpy.tanh.html)

[激活函数-双曲正切函数tanh函数- 知乎](https://zhuanlan.zhihu.com/p/105141338)

>>> np.tanh((0, np.pi\*1j, np.pi\*1j/2))  
array([ 0. +0.00000000e+00j, 0. -1.22460635e-16j, 0. +1.63317787e+16j])  
>>> # Example of providing the optional output parameter illustrating  
>>> # that what is returned is a reference to said parameter  
>>> out2 = np.tanh([0.1], out1)  
>>> out2 is out1  
True

1. [numpy.multiply](https://docs.scipy.org/doc/numpy/reference/generated/numpy.multiply.html)

数组相乘

8.0  
>>> x1 = np.arange(9.0).reshape((3, 3))  
>>> x2 = np.arange(3.0)  
>>> np.multiply(x1, x2)  
array([[ 0., 1., 4.],  
 [ 0., 4., 10.],  
 [ 0., 7., 16.]])