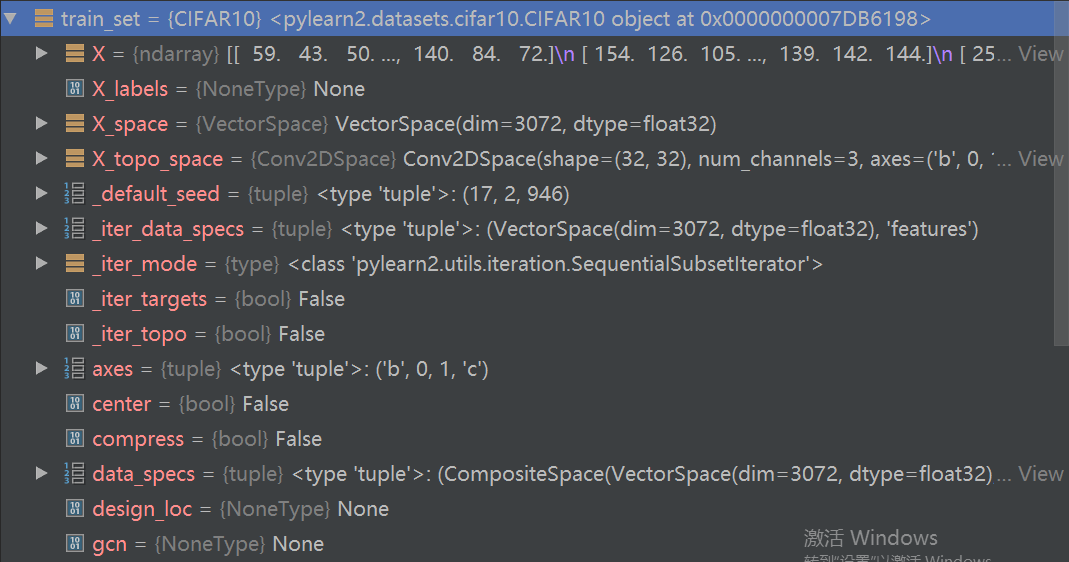
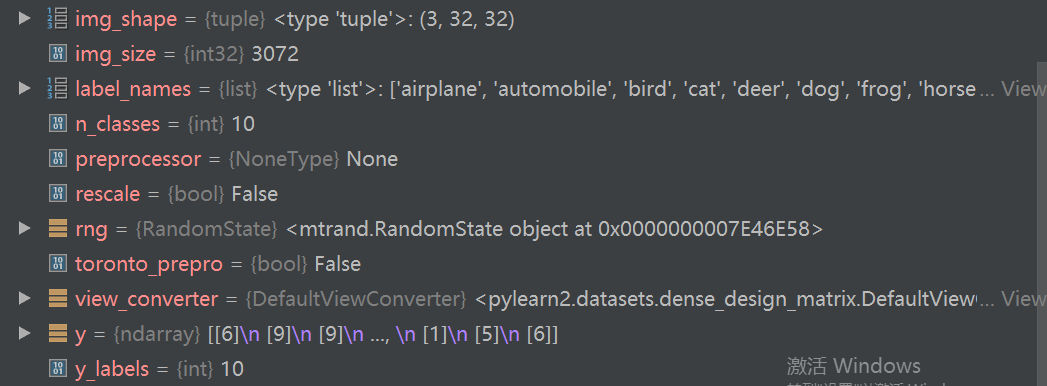
Cifar-10 数据集的处理

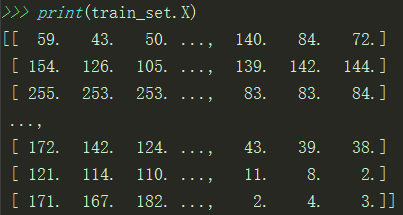
train\_set = CIFAR10(which\_set="train",start=0,stop = train\_set\_size)  
valid\_set = CIFAR10(which\_set="train",start=train\_set\_size,stop = 50000)  
test\_set = CIFAR10(which\_set="test")

*def* \_\_init\_\_(self, *which\_set*, *center*=False, *rescale*=False, *gcn*=None,  
 *start*=None, *stop*=None, *axes*=('b', 0, 1, 'c'),  
 *toronto\_prepro* = False, *preprocessor* = None):

dtype = 'uint8'  
ntrain = 50000  
nvalid = 0 # artefact（人工制品，人工产物; 人为现象; [医] 伪假象; [生物学] 矫作物）, we won't use it  
ntest = 10000









45000是训练集图片数目，3072是32\*32\*3，一个图像的RGB三通道所有灰度值。

*if start is not* None:  
 # This needs to come after the prepro so that it doesn't  
 # change the pixel means computed above for toronto\_prepro  
 *assert start* >= 0  
 *assert stop* > *start  
 assert stop* <= X.shape[0]  
 X = X[*start*:*stop*, :]  
 y = y[*start*:*stop*, :]  
 *assert* X.shape[0] == y.shape[0]  
  
*if which\_set* == 'test':  
 *assert* X.shape[0] == 10000

start和stop用来取X的start到stop的样本。

*assert*

1、assert语句用来声明某个条件是真的。

2、如果你非常确信某个你使用的列表中至少有一个元素，而你想要检验这一点，并且在它非真的时候引发一个错误，那么assert语句是应用在这种情形下的理想语句。

3、当assert语句失败的时候，会引发一AssertionError。

测试程序：

>>> mylist = ['item']

>>> assert len(mylist) >= 1

>>> mylist.pop()

'item'

>>> assert len(mylist) >= 1

Traceback (most recent call last):

  File "<stdin>", line 1, in <module>

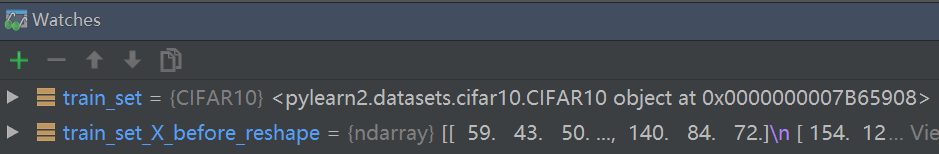
AssertionError

>>>

cifar10数据集有

reshape究竟干了啥？

train\_set\_X\_before\_reshape：

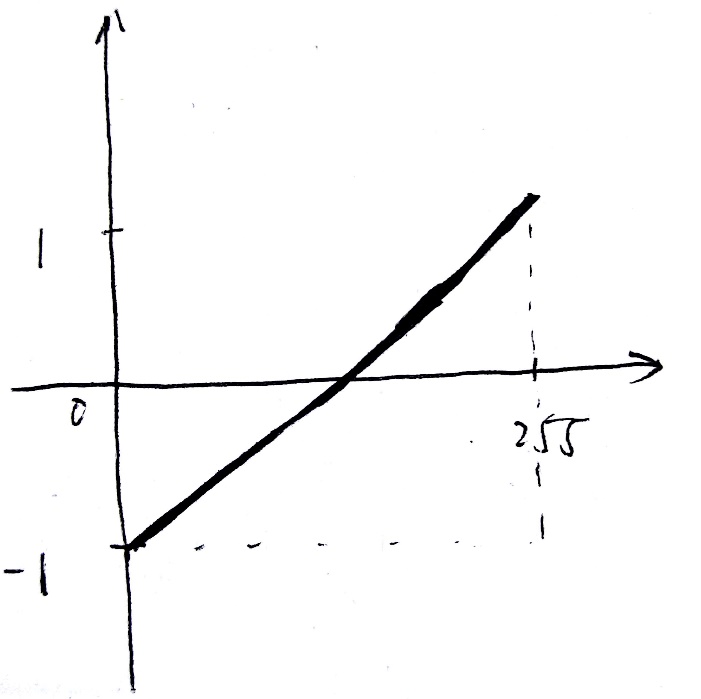


>>> train\_set\_X\_before\_reshape.shape

Out[1]: (45000L, 3072L)

np.subtract(np.multiply(2./255.,train\_set.X),1.)

做线性变换，一个45000×3072的矩阵全体乘以2除以255，减1，变换到[-1，1]区间。

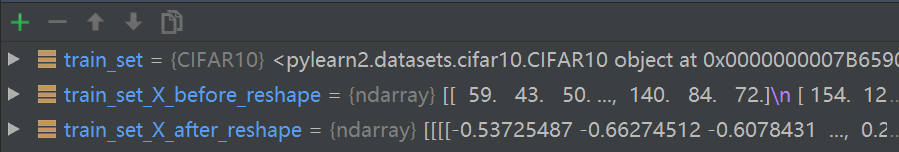


然后再reshape到

>>> train\_set.X.shape

Out[4]: (45000L, 3L, 32L, 32L)

Before和after



train\_set.y = np.hstack(train\_set.y)

>>> train\_set.y.shape

Out[5]: (45000L, 1L)

>>> train\_set.y.shape

Out[6]: (45000L,)

>>> train\_set.y

Out[1]:

array([[6],

[9],

[9],

...,

[1],

[5],

[6]], dtype=uint8)

>>> train\_set.y

Out[2]: array([6, 9, 9, ..., 1, 5, 6], dtype=uint8)

**numpy.hstack**

numpy.hstack(*tup*)[[source]](http://github.com/numpy/numpy/blob/v1.10.1/numpy/core/shape_base.py#L232-L280)

Stack arrays in sequence horizontally (column wise).

Take a sequence of arrays and stack them horizontally to make a single array. Rebuild arrays divided by [hsplit](http://docs.scipy.org/doc/numpy/reference/generated/numpy.hsplit.html#numpy.hsplit).

|  |  |
| --- | --- |
| **Parameters:** | **tup** : sequence of ndarrays  All arrays must have the same shape along all but the second axis. |
| **Returns:** | **stacked** : ndarray  The array formed by stacking the given arrays. |

See also

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

Join a sequence of arrays along a new axis.

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

Stack arrays in sequence vertically (row wise).

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

Stack arrays in sequence depth wise (along third axis).

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

Join a sequence of arrays along an existing axis.

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

Split array along second axis.

Notes

Equivalent to np.concatenate(tup, axis=1)

Examples

>>>

>>> a = np.array((1,2,3))

>>> b = np.array((2,3,4))

>>> np.hstack((a,b))

array([1, 2, 3, 2, 3, 4])

>>> a = np.array([[1],[2],[3]])

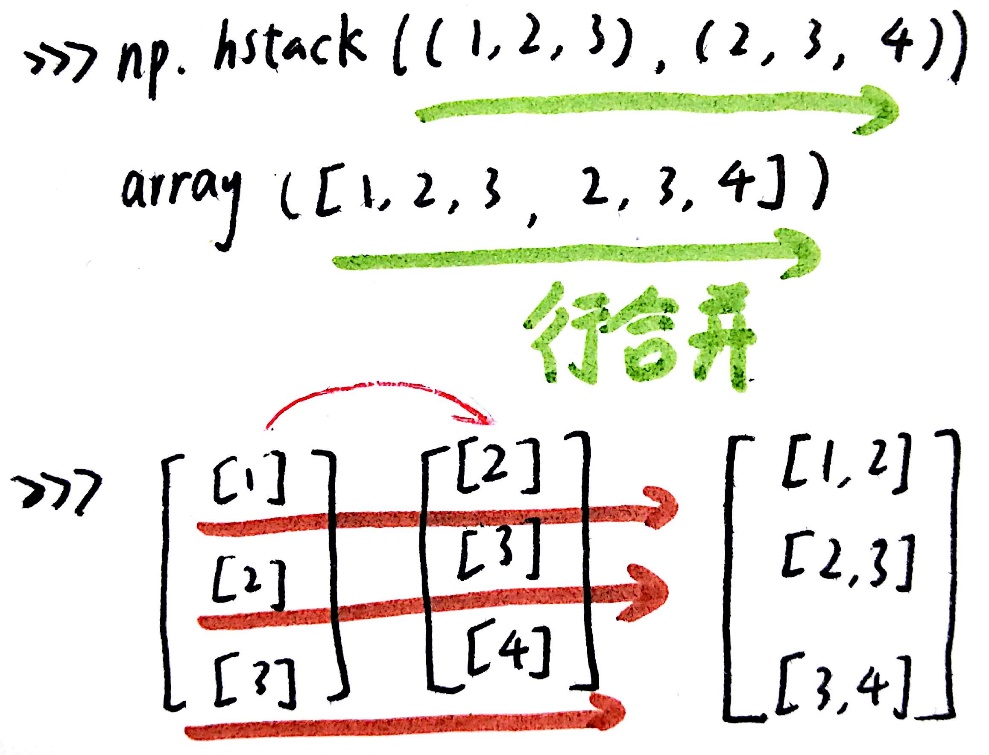
>>> b = np.array([[2],[3],[4]])

>>> np.hstack((a,b))

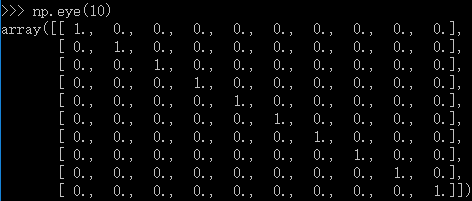
array([[1, 2],

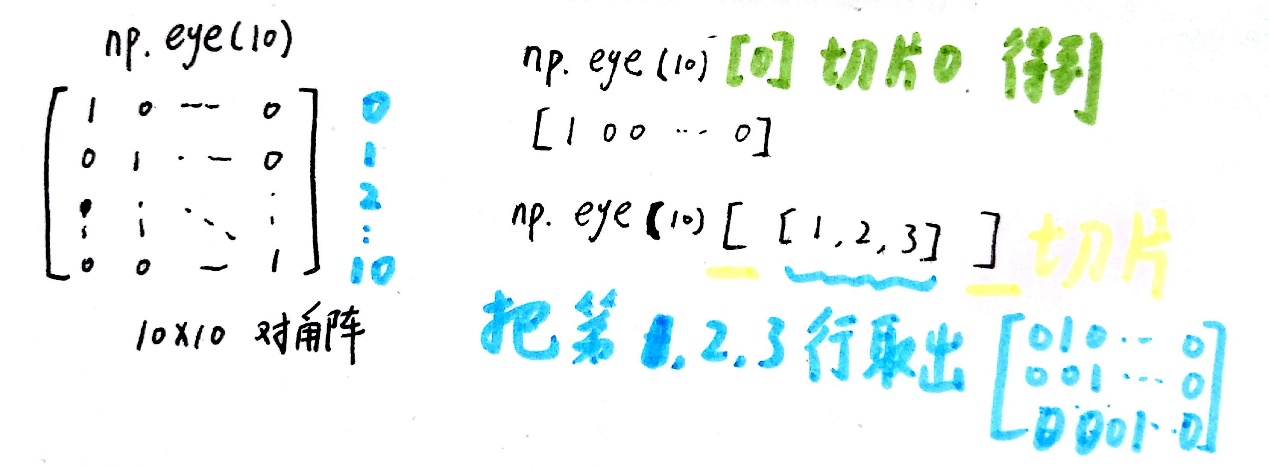
[2, 3],

[3, 4]])



train\_set.y = np.float32(np.eye(10)[train\_set.y])





train\_set.y = np.float32(np.eye(10)[train\_set.y])

把0，1变成-1，1

