

python进阶-迭代器和生成器

实验背景

迭代器是一种数据遍历的方式，是许多编程语言的一个重要功能；python中也有，而生成器应该是python独有的特性。

我们开始认识一下这两个功能特性吧

实验内容

什么是迭代器

python中的迭代器是访问集合的一种方式，每次访问记住遍历的位置。

- 迭代器通过 `iter()` 将一个可迭代对象变成迭代器，例如 `list tuple dict set str` 等，还有就是后面讲到的生成器 (`generator`) 都是可迭代对象
- 通过 `next()` 或者 `for` 循环访问迭代的数据

我们来看下例子

```
from collections import Iterable, Iterator
# del iter
a = [1, 2, 4, 6, 8, 6, 10]
print(isinstance(a, Iterable))
print(type(a))

it = iter(a)
print(next(b))
# True
# <class 'list'>
# 8
```

使用迭代器

```
for i in it:
    print(i)

while True:
    try:
        print(next(it))
    except StopIteration:
        break
```

自定义一个迭代器：通过实现 `__iter__` 和 `__next__` 来实现迭代器的功能；其实python很多的对象都实现的迭代器的功能，都可以通过for循环的方式来访问，比如文件访问。

```
with open('./test', 'rb') as f:
    for line in f:
        print(line) # 读取文件的每一行较其他语言是相当的方便的
```

```
class MyNumbers:
    def __iter__(self):
        self.a = 1
        return self

    def __next__(self):
        x = self.a
        self.a += 1
        return x

myclass = MyNumbers()
myiter = iter(myclass)
next(myiter)
```

tensorflow中的迭代器

tf.data也是通过迭代器的方式来生成数据的，如下

```
train_dataset = tf.data.Dataset.from_tensor_slices(train)
iterator = tf.data.Iterator.from_structure(train_dataset.output_types,
train_dataset.output_shapes)
img, label = iterator.get_next()
```

什么是生成器

上面说过生成器的结果是产生一个可以迭代的对象，可以通过for循环来访问。

那它和其他的可迭代对象有什么不同呢？

生成器的特性：

- 它是一个延迟的可迭代对象，也就是说它不是一次性生成所有的迭代的内容（如列表中的所有元素），而是需要的时候才生成结果即：一边循环一边生成计算
- 好处是：不用一次性存储所有的内容，节省内存。这个对于访问大量数据的时候是十分有帮助的

我们来看看上面的这些特性

(1)通过yield来构造一个生成器

```
def normal_iter():
    ret = []
```

```
    for i in range(10):
        ret.append(i**2)
    return ret
def generator():
    for i in range(10):
        yield i**2
# normal_iter 存储所有的结果
# generator 通过yield返回一个生成器，调用的时候才返回生成的对象
for i in normal_iter():
    print(i)
for i in generator():
    print(i)

# 另外可以通过()构建一个生成器
L = [x * x for x in range(10)]
g = (x * x for x in range(10))
print(L)
# [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
print(g)
# <generator object <genexpr> at 0x7ff566b8cf10>
```

实验结语

本实验和大家简单描述了python迭代器和生成器的使用，希望对大家有帮助。