

第九次作业

王舵 19377163

生成器和迭代器有两种常见的使用场景。

- 一. 后项需要前项导出，且无法通过列表推导式生成。例如，时间序列中的“随机游走”便是一种满足上述条件的序列数据。其公式为 $X_t = \mu + X_{t-1} + w_t$ ，其中 μ 为漂移量， w_t 是满足某种条件的独立同分布的随机变量，这里假设其服从正态分布 $N(0, \sigma^2)$ 。本题要求写出实现该功能的迭代器函数。具体要求如下：

1. 实现 random_walk 生成器，输入参数 μ ， X_0 ， σ^2 ， N ，函数将迭代返回 N 个随机游走生成的变量。

```
def random_walk(mu,X_0,sigma,N):
    count = 0
    X_b = X_0
    while(count<N):
        Norm=np.random.normal(loc=0,scale=sigma,size=1)
        X_a=mu+X_b+Norm
        yield X_a
        X_b = X_a
        count += 1
    return None
```

```
if __name__ == "__main__":
    r=random_walk(3,5,3,5)
    while True:
        try:
            print(next(r))
        except StopIteration as si:
            print(si.value)
            break
```

2. 利用 zip，实现拼合多个 random_walk 的生成器，以生成一组时间上对齐的多维随机游走序列

```
r1=random_walk(3,5,3,10)
r2=random_walk(3,6,4,10)
while True:
    try:
        print(next(zip(r1,r2)))
    except StopIteration as si:
        print(si.value)
        break
```

测试结果：

```
[4.5240711]
[7.10547985]
[10.17550709]
[16.91248559]
[23.04069133]
None
(array([6.17419324]), array([9.86935645]))
(array([9.94021049]), array([19.61895067]))
(array([14.61407127]), array([26.83257343]))
(array([14.74633752]), array([31.44363195]))
(array([18.88161653]), array([35.22972986]))
(array([19.86013199]), array([47.41511155]))
(array([19.59412312]), array([57.79069673]))
(array([24.42097558]), array([58.00307161]))
(array([25.81445765]), array([59.88706713]))
(array([30.43852888]), array([60.43883375]))
None
```

二. 需要迭代的内容数据量过大，无法一次性加载。例如，在图像相关的深度学习任务中，由于数据总量过大，一次性加载全部数据耗时过长、内存占用过大，因此一般会采用批量加载数据的方法。（注：实际应用中由于需要进行采样等操作，通常数据加载类的实现原理更接近字典，例如 pytorch 中的 Dataset 类。）现提供文件 FacelImages.zip(<http://vis-www.cs.umass.edu/fddb/originalPics.tar.gz>，其中包含 5000 余张人脸图片。要求设计 FaceDataset 类，实现图片数据的加载。具体要求：

1. 类接收图片路径列表

根据路径遍历所有目录，并将所有图片路径添加到列表中

```
class FaceDataset:
    def __init__(self,filepath) -> None:
        self.filepath = filepath
        self.photo_list = []
        self._n= 0

    def ptoto_list_generate(self,filepath):
        for a,b,c in os.walk(filepath):
            if b==[]:
                for i in c:
                    self.photo_list.append(os.path.join(a,i))
```

列表如下：共有 28000 张左右图片

```
PS C:\Users\86186\Desktop\现代程序设计> c:; cd 'c:\Users\86186\Desktop\现代程序设计'; & 'C:\Users\86186\AppData\Local\Programs\Python\Python37\python.exe' 'c:\Users\86186\.vscode\extensions\ms-python.python-2021.11.1422169775\pythonFiles\lib\python\debugpy\launcher' '13975' '-.' 'c:\Users\86186\Desktop\现代程序设计\9\import math.py'

['C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_130.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_135.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_140.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_141.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_158.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_160.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_163.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_18.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_197.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_198.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_203.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_209.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_230.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_247.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_255.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_269.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_278.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_300.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_352.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_367.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_372.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_381.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_389.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_392.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_393.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_397.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_407.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_408.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_414.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_416.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_422.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_423.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_426.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_427.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_429.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_432.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_445.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_463.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_488.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_491.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_499.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_501.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_517.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_521.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_523.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_534.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_552.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_561.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_576.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_581.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_582.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_607.jpg', 'C:\Users\86186\Desktop\现代程序设计\originalPics\2002\07\19\big\img_63
```

2. 类支持将一张图片数据以 ndarray 的形式返回（可以利用 PIL 库实现）。

```
def transform_to_ndarray(self, img):
    return np.array(img)
```

运行结果：

```
...
[[193 217 245]
 [196 220 248]
 [195 219 247]
 ...
 [213 206 178]
 [210 203 175]
 [212 205 177]]

[[196 217 246]
 [194 215 244]
 [196 217 246]
 ...
 [243 230 196]
 [241 228 196]
 [241 228 196]]

[[197 216 246]
 [195 214 244]
 [197 216 246]
 ...
 [244 230 193]
 [246 232 197]
 [247 233 198]]

[[48 41 35]
 [53 46 38]
 [61 52 45]
 ...
 [21 20 16]
 [21 20 16]
 [21 20 16]]

[[50 43 37]
 [54 47 39]
 [61 52 43]
 ...
 [20 19 15]
 [20 19 15]
 [20 19 15]]
```

3. 实现__iter__方法。

返回定义了__next__的对象

```
def __iter__(self):  
    return self
```

4. 实现__next__方法，根据类内的图片路径列表，迭代地加载并以 ndarray 形式返回图片数据。

```
def __next__(self):  
    if self._n < len(self.photo_list):  
        img=Image.open(self.photo_list[self._n])  
        self._n += 1  
        print(self.transform_to_ndarray(img))  
    else:  
        raise StopIteration('大于max:{}'.format(len(self.photo_list)))
```

测试结果（只输出前 100 张）：

```
import numpy as np  
import os  
from PIL import Image  
  
filepath = r"C:\Users\86186\Desktop\现代程序设计\originalPics"
```

```
fd=FaceDataset(filepath)  
fd.ptoto_list_generate(filepath)  
fditer=iter(fd)  
for i in range(100):  
    next(fditer)
```

问题 输出 终端 调试控制台

```
[[193 192 190]
 [192 191 189]
 [190 189 187]
 ...
 [184 183 181]
 [187 186 184]
 [191 190 188]]

[[190 189 187]
 [191 190 188]
 [188 187 185]
 ...
 [186 185 183]
 [187 186 184]
 [189 188 186]]

[[189 188 186]
 [190 189 187]
 [188 187 185]
 ...
 [182 181 179]
 [183 182 180]
 [185 184 182]]]
[[[121 166 189]
 [133 178 201]
 [131 176 199]
 ...
 [138 184 208]
 [134 180 204]
 [130 176 200]]

[[128 173 196]
 [140 185 208]
 [138 183 206]
 ...
 [139 185 209]
 [136 182 206]
 [131 177 201]]

[[127 172 195]
 [139 184 207]
 [137 182 205]
 ...
 [140 186 210]
 [136 182 206]
```