hw01_eac721

September 19, 2019

0.1 Homework 1

Please import the following packages.

[1]:
```
import numpy as np
import scipy.sparse
```

Please download `memory.py` from Resources/Homework/Homework01 on NYU Classes. Save it to the same directory as the Jupyter notebook. Please import the following package.

[2]:
```
import memory
```

0.1.1 Loops

How to go through the entries of an array top to bottom/left to right?

1. Given an array `inputArray`, write a `for` loop that flattens it to `outputArray`. For example, `inputArray = np.array([[1,2], [3,4]])` would yield `np.array([1,2,3,4])` for `outputArray`.

[3]:
```
inputArray = np.array([[1,2], [3,4]])

temp=[]
for i in range(len(inputArray)):
    for j in range(len(inputArray[i])):
        temp.append(inputArray[i][j])
outputArray = np.array(temp)

# Checks:
print(outputArray)
#print(type(outputArray))

#np.array_equal(np.array([1,2,3,4]), outputArray)
```

```
[1 2 3 4]
```

2. Given a jagged array `inputArray`, write a `for` loop that flattens it to `outputArray`. For example, `inputArray = np.array([[1,2,3], [4]])` would yield `np.array([1,2,3,4])` for `outputArray`.

[4]:
```
inputArray = np.array([[1,2,3], [4]])

temp=[]
for i in range(len(inputArray)):
    for j in range(len(inputArray[i])):
        temp.append(inputArray[i][j])
outputArray = np.array(temp)

#Checks:
print(outputArray)
#print(type(outputArray))
#np.array_equal(np.array([1,2,3,4]), outputArray)
```

```
[1 2 3 4]
```

0.1.2 Packages

How to import and use packages?

3. Create an array `A` from the list

   `[[1, 0, 0, 1, 0, 0], [0, 0, 2, 0, 0, 1], [0, 0, 0, 2, 0, 0]]`
   Use `memory.getsizeof` to determine how much space `A` takes up in memory.

[5]:
```
A=np.array([[1, 0, 0, 1, 0, 0], [0, 0, 2, 0, 0, 1], [0, 0, 0, 2, 0, 0]])
#type(A)
memory.getsizeof(A)
```

[5]:
```
144
```

4. Use `scipy.sparse.csr_matrix` to covert `A` into `S`. Use `memory.getsizeof` to determine how much space `S` takes up in memory.

[6]:
```
S=scipy.sparse.csr_matrix(A)
memory.getsizeof(S)
```

[6]:
```
76
```

5. What accounts for the difference? Try calling `print` on `S`.

[7]:
```
#?scipy.sparse.csr_matrix
print(S)
```

```
  (0, 0)        1
  (0, 3)        1
  (1, 2)        2
  (1, 5)        1
  (2, 3)        2
```

S is a sparse row matrix, which only reports the locations of non-zero entries in the array A. For example, the first entry at (0,0) is 1 and the next reported entry is not until the (0,3) entry which is also 1. By only reporting the locations non-zero entries rather than every entry, the array size in memory is smaller.