

CS 251 Inlab 4 : Python, Numpy, Scipy Matplotlib

Please refer to the submission guidelines at the end of this document before submitting.

General Instructions

- **Unless otherwise stated, you CAN'T use any kind of loops or comprehensions (list, dict, etc).**
- For task0, put your code in appropriate locations in task0.py
- For other tasks, put your code at appropriate locations in inlab4_tasks.py
- By matrix, we mean two dimensional numpy arrays.
- If we mention array, we mean an n-dimensional numpy array.
- Make sure you know what you write, you might be asked to explain your code at a later point in time.
- The submission will be graded automatically, so stick to the naming conventions strictly.
- The deadline for this lab is **5pm on 5th September 2019**.

How to use the autograder

- Run: `python3 autograder.py --task <task_number>`
- `<task_number>` can be an integer between 1 and 5 (both inclusive) or 'all'
- No argument implies all test cases will be checked.
- Following are some valid ways to run autograder:
 - `python3 autograder.py`
 - `python3 autograder.py --task 3`
 - `python3 autograder.py --task all`
- Following is an invalid way to run it:
 - `python3 autograder.py --task 1,2`

Task 0 - (12 Marks)

Do as directed in task0.py

ProTip: some of the functions that you might find useful for this task are: reshape, argmax, amax, copy, matmul and astype.

Task 1 - (10 Marks)

Write a function checkerboard which accepts a **positive integer** n and returns a $n \times n$ **integer matrix** M which contains checkerboard pattern. Note that M can have 0's and 1's only. Top left corner of M should be 0.

ProTip: just basic indexing should do the trick! Lookup zeros, astype

Task 2 - (8 Marks)

Write a function `SimpleNonLinearity` which takes a parameter called '`leak`' and an array and computes $g(x)$ for each element x in the array. The function should not change the input array. Input can be either an int or a float array, but, the output must be a float array. Assume that $0 \leq \text{leak} < 1$.

$$g(x) = \begin{cases} x & \text{if } x > 0 \\ \text{leak} \times x & \text{otherwise} \end{cases}$$

Task 3 - (10 marks)

Write a function to normalise a matrix of shape (m, n) along its columns. By normalise, we mean that the mean of each column should be 0 and variance of each column should be 1.

To do this, first compute the mean of input along its columns. Subtract the mean from input matrix. Divide the result by its standard deviation along its columns and return it.

ProTip: Useful functions include `mean` and `std`. Use `reshape` wisely and it will become a cakewalk.

Task 4 - (15 marks)

Write a function to get the positions of `top_k` values of each row of a matrix of shape (m,n).

Concretely, if `A` is the input matrix of shape (m,n), then return an integer matrix `R`, such that:

R_{ij} is the index of an element in i^{th} row of `A` which has rank j when that row is sorted in decreasing order.

For example, if the input array '`arr`' is:

```
[[ 6  4  4  7]
 [ 7  8  1  0]]
```

```
[11 10 5 11]]
```

Then `top_k(arr, 2)` will be:

```
[[3 0]
```

```
[1 0]
```

```
[3 0]]
```

Explanation for first row: when we sort `6,4,4,7` in decreasing order, we get: `7, 6, 4, 4`. Top 2 elements are 7 and 6. And, their positions in `arr` are: 3 and 0 respectively. So, the first row of the result is `3,0`.

Note: for breaking ties, higher indices should come first.(See the last row of result).

ProTip: If you use `argsort` wisely, then you should be able to do it in 1 line.

Task 5 - (15 marks)

Magic square is a $n \times n$ array where sum of each row, each column and each diagonal is same.

Write a function `is_magic_square` which takes a matrix of shape (n, n) and returns `True` if the input is a non-trivial magic square and `False` if it isn't. Assume **n is odd**. In a non-trivial magic square, you have to make sure that all the elements are distinct.

Submission Instructions

After creating your directory, package it into a tarball

`inlab4-<rollno>.tar.gz`

The directory structure should be as follows (nothing more nothing less)

```
inlab4-<rollno>/
├── task0.py
└── inlab4_tasks.py
```