AIML - CS 335

Lab 2: Linear Algebra and basic Torch

11^{th} August. 2022

Lecturer: Abir De

Note: Please read the instructions mentioned in the questions carefully. We have provided boilerplate code for each question. Please ensure that you make changes in the areas marked with TODO.

1 LU Decomposition

LU decomposition refers to the factorization of square matrix A into two factors, a unit in lower triangular matrix L and an upper triangular matrix U, A = LU.

Linear equations We can represent linear equation with matrix form $Ax = b$ where we want to solve the equation for x given A and B. Firstly we need to decompose matrix A to $A = LU$ with LU decomposition. After that our linear equations turns to $LUx = B$. We can insert substitution $y = Ux$ and get equation $Ly = B$. Now we can easily solve this equation with forward substitution and find the y. After finding y we solve $Ux = y$ with backward substitution and in the end we find the missing x. You may assume that the system of linear equations admit an unique solution.
Your task in this question is to obtain the matrices L and U for a given square matrix $A \in \mathbb{R}^{n \times n}$.

Complete the function `LU_decomposition` in the script.

Function Signature: `def LU_decomposition(A: np.array):`
This returns two matrices L, U which are of same shape as A.

2 Simple Torch

Using the PyTorch library, write Python scripts to achieve the tasks listed below. To install the PyTorch library, you can use the following command:

```
pip3 install torch --extra-index-url https://download.pytorch.org
/whl/cpu
```

Tasks:

1. Create and return a torch matrix A of shape $50 \times 40 \times 5$ containing random numbers in the range [0,1).
2. You are given a matrix B of datatype `float32`. Return this matrix after converting it's datatype to `int32`

3. Generate a random matrix C of shape 3×100. Permute the rows of the matrix to generate a new matrix D, such that the 1st row becomes the 2nd row, the 2nd row becomes the 3rd row and the 3rd row becomes the 1st. Return both C and D.

4. Generate a random matrix E of shape 20×10. Compute the sum along each row of this matrix in a new vector F. Return both E and F

5. You are given three matrices, $G1, G2, G3$, each of shape 10×10. Generate a new matrix H of shape $10 \times 10 \times 3$ by combining the given matrices. Return H

3 Vectorization

In this problem, we will implement a function to compute the pairwise ranking loss between two sets of scores. Given a set of positive scores $P \in \mathbb{R}^{n_1 \times 1}$ and a set of negative scores $N \in \mathbb{R}^{n_2 \times 1}$, we want to impose the constraint that the positive scores should be greater than the negative scores. Therefore, we design the following loss function:

$$L(P, N) = \sum_{p \in P, n \in N} max(0, n - p)$$

1. Complete the function `pairwise_ranking_loss_looped` using for loops to obtain $L(P, N)$ for any given vectors P and N.

2. Note that vectorized operations can make the above computations blazingly fast. Complete the function `pairwise_ranking_loss_vec` to obtain the same value as above in a vectorized manner.

We will be checking the execution time of the functions rigorously, so make sure that `pairwise_ranking_loss_vec` has no for loop. Also make sure that the two functions `pairwise_ranking_loss_looped` and `pairwise_ranking_loss_vec`, return the same value for any given input. The main function is provided. You can manipulate the scores P, N to observe the difference between the execution times of the functions.

4 Submission instructions

Complete the functions in `assignment.py`. Keep the file in a folder named `<ROLL_NUMBER>_L1` and compress it to a tar file named `<ROLL_NUMBER>_L1.tar.gz` using the command

```
tar -zcvf <ROLL_NUMBER>_L1.tar.gz <ROLL_NUMBER>_L1
```

Submit the tar file on Moodle. The directory structure should be -

```
<ROLL_NUMBER>_L1
| - - - - assignment.py
```

Replace `ROLL_NUMBER` with your own roll number. If your Roll number has alphabets, they should be in *“small”* letters.