Homework 1 - Python Tutorial : Gauss Elimination

Theory and Practice in Deep Learning

Due Date : 17:00, 25 Sep 2020

- One of the fundamental problems in many scientific applications is to solve the following linear system for unknown $\boldsymbol{x}$.

$$\boldsymbol{Ax} = \boldsymbol{b}$$

This homework considers the basic algorithm, **which is known as Gauss Elimination**, for solving above linear system.
For simple explanation, consider the following system with four unknowns:

$$\begin{cases} 6x_1 - 2x_2 + 2x_3 + 4x_4 &= 16 \\ 12x_1 - 8x_2 + 6x_3 + 10x_4 &= 26 \\ 3x_1 - 13x_2 + 9x_3 + 3x_4 &= -19 \\ -6x_1 + 4x_2 + x_3 - 18x_4 &= -34 \end{cases} \tag{1}$$

With usual convention as we always do, we can delete every coefficient of x_1 except the first equation:

$$\begin{cases} 6x_1 & -2x_2 & +2x_3 & +4x_4 & = 16 \\ & -4x_2 & +2x_3 & +2x_4 & = -6 \\ & -12x_2 & +8x_3 & +x_4 & = -27 \\ & 2x_2 & +3x_3 & -14x_4 & = -18 \end{cases}$$

Repeating on x_2,x_3, we can have the following upper triangular system:

$$\begin{cases} 6x_1 & -2x_2 & +2x_3 & +4x_4 & = 16 \\ & -4x_2 & +2x_3 & +2x_4 & = -6 \\ & & +2x_3 & -5x_4 & = -9 \\ & & & -3x_4 & = -3 \end{cases}$$

- Above linear system can be written as matrix-vector multiplication as follow:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}$$

With this matrix form, well-known pseudocode for Gauss Elimination is given below.

Homework(20pt)

1. With the given pseudocode, make a **python program** that computes the solution of (1) using Gauss Elimination. In this problem, DO NOT use any function from numpy! Only use python-basic structure(such as list, for..) (10pt).
2. Repeat problem 1 with numpy:) (10pt)

The following must be considered before you submit this homework.

- Work with Jupyter Notebook or other python interpreter and export results as .py files.
- The file name MUST be name_studentnumber_HW1_1.py(for problem 1) and name_studentnumber_HW1_2.py(for problem 2). For example, BJ_200313673_HW1_2.py will be mine.
- Each problem MUST contain the function whose name is Naive_Gauss.
- Suggested : Add comments for explanation.
- Important : You can make some googling for this HW, but DO NOT copy them. Always be honest!

procedure *Naive_Gauss*$(n, (a_{ij}), (b_i), (x_i))$
integer i, j, k, n; **real** $sum, xmult$
real array $(a_{ij})_{1:n\times 1:n}, (b_i)_{1:n}, (x_i)_{1:n}$
for $k = 1$ **to** $n-1$
  for $i = k+1$ **to** n
    $xmult \leftarrow a_{ik}/a_{kk}$
    $a_{ik} \leftarrow xmult$
    for $j = k+1$ **to** n
      $a_{ij} \leftarrow a_{ij} - (xmult)a_{kj}$
    end for
    $b_i \leftarrow b_i - (xmult)b_k$
  end for
end for
$x_n \leftarrow b_n/a_{nn}$
for $i = n-1$ **to** 1
  $sum \leftarrow b_i$
  for $j = i+1$ **to** n
    $sum \leftarrow sum - a_{ij}x_j$
  end for
  $x_i \leftarrow sum/a_{ii}$
end for
end procedure *Naive_Gauss*

Figure 1: Gauss Elimination Pseudocode

Have fun with your first HW:)