Adamson University
College of Engineering
Computer Engineering Department

Experiment No. # 7

MATRICES

Objective

1. Be familiar with matrices and their relation to linear equations.
2. Perform basic matrix operations.
3. Program and translate matrix equations and operations using Python.

Algorithm

1. Type the main title of this activity as "Matrices"
2. On your GitHub, create a repository name Linear Algebra 58013
3. On your Colab, name your activity as Python Exercise 7.ipynb and save a copy to your GitHub repository

Discussion

Matrices

The notation and use of matrices is probably one of the fundamentals of modern computing. Matrices are also handy representations of complex equations or multiple inter-related equations from 2-dimensional equations to even hundreds and thousands of them.

Declaring Matrices

Just like our previous laboratory activity, we'll represent system of linear equations as a matrix. The entities or numbers in matrices are called the elements of a matrix. These elements are arranged and ordered in rows and columns which form the list/array-like structure of matrices. And just like arrays, these elements are indexed according to their position with respect to their rows and columns. This can be represented just like the equation below. Whereas A is a matrix consisting of elements denoted by $a_{i,\ j}$. Denoted by i is the number of rows in the matrix while j stands for the number of columns.
Do note that the size of a matrix is i x j.

$$A = \begin{bmatrix} a_{(0,\ 0)} & a_{(0,\ 1)} & \cdots & a_{(0,\ j-1)} \\ a_{(1,\ 0)} & a_{(1,\ 1)} & \cdots & a_{(1,\ j-1)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{(i-1,\ 0)} & a_{(i-1,\ 1)} & \cdots & a_{(i-1,\ j-1)} \end{bmatrix}$$

Coding Activity 7

We already gone over some of the types of matrices as vectors but we'll further discuss them in this laboratory activity. Since you already know how to describe vectors using shape, dimensions and size attributes, we'll use them to analyze these matrices.

```
## Since we'll keep on describing matrices. Let's make a function.
def describe_mat(matrix):
  print(f'Matrix:\n{matrix}\n\nShape:\t{matrix.shape}\nRank:\t{matrix.ndim}\n')
```

```
## Declaring a 2 x 2 matrix
A = np.array([
  [1, 2],
  [3, 1]
])
describe_mat(A)
```

```
G = np.array([
   [1,1],
   [2,2]
])
describe_mat(G)
```

```
## Declaring a 3 x 2 matrix
B = np.array([
   [8, 2],
   [5, 4],
   [1, 1]
])
describe_mat(B)
```

```
H = np.array([1,2,3,4,5])
describe_mat(H)
```

Answers:

```
## Declaring a 2 x 2 matrix
A = np.array([
    [1, 2],
    [3, 1]
])
describe_mat(A)

G = np.array([
    [1,1],
    [2,2]
])
describe_mat(G)
```

```
Matrix:
[[1 2]
 [3 1]]

Shape:  (2, 2)
Rank:   2

Matrix:
[[1 1]
 [2 2]]

Shape:  (2, 2)
Rank:   2
```

```
## Declaring a 3 x 2 matrix
B = np.array([
    [8, 2],
    [5, 4],
    [1, 1]
])
describe_mat(B)
```

```
Matrix:
[[8 2]
 [5 4]
 [1 1]]

Shape:  (3, 2)
Rank:   2
```

```
H = np.array([1,2,3,4,5])
describe_mat(H)
```

```
Matrix:
[1 2 3 4 5]

Shape:  (5,)
Rank:   1
```

GitHub Permalink:
https://github.com/MNLLEMM/58013-Linear-Algebra/blob/a89d0e51b0c0d2d5f6c03bbe02a71a467ab2d65d/Python_Exercise_7.ipynb